BLOCK DESIGNS AND GRAPH THEORY

by

LINDSAY PATERSON

Thesis submitted for the degree of Doctor of Philosophy in
the University of Edinburgh 1981

ABSTRACT OF THESIS

In this thesis we use graph theory to investigate the structure of incomplete block designs. Specifically, we use graph theory methods and ideas to help develop simple and reliable means of producing efficient cyclic designs and α designs; we also point out, however, that there is a great deal of scope for graph theory to be used in other ways in connection with block designs.

The necessary background from design of experiments and graph theory is sketched in Chapter 1, where also we define the principal concepts which link the two subjects: the variety concurrence graph, the block concurrence graph, and the design graph. We describe examples to clarify all these ideas. We observe, furthermore, that other researchers have made only inchoate and sporadic use of graph theory as a methodology for the investigation of block designs.

Having outlined this background, we then turn, in Chapter 2, to one area in particular: we show that a fairly accurate assessment of the efficiency of a block design can be had by counting the numbers of circuits of various lengths in its variety concurrence graph. On the strength of this we define the graphical criteria of efficiency: $\underline{\underline{C}}_h$ (for $h \geq 2$) is defined to be the number of circuits of length h in this graph. We examine the strengths - both intuitive and mathematical - of these criteria, and also their close theoretical relationship with the widely used harmonic mean efficiency factor. We indicate that their main advantage over the latter criterion is their ease of calculation - and that this stems precisely from

their origins in the graph.

In Chapters 3 and 4 we demonstrate in further detail just how straightforward the graphical criteria are when they are applied to, respectively, cyclic designs and α designs. We devise formulas to enumerate circuits of lengths two, three, and four in such designs by exploiting the considerable simplifications that are made possible by our graph theory approach. And we illustrate the power of the graphical criteria - and their facility of computation - by describing their application to several examples.

Subsequently, in Chapter 5, we present algorithms which use some of the formulas obtained in Chapter 4 to generate efficient α designs. The rapidity of these algorithms is due, again, to the simplicity that arises from graph theory. This rapidity makes practicable a wider searching for efficient designs than has been possible hitherto. (In Appendix 2 we exemplify the improvements which can be obtained in consequence of this.)

Finally, in Chapter 6, we discuss the link between the precision with which a design estimates variety differences and certain structural properties of the variety concurrence graph. Out of this discussion emerges a set of rules for the efficient allocation of controls in an experiment.

DECLARATION

This thesis has been composed by me, and the work presented in it is my own.

TABLE OF CONTENTS

INTRODUCTION

There often comes a time in the development of a subject when it is profitable to step outside it and look back in. That is what we do in this thesis for the subject of general block designs: we seek to demonstrate that many of the important problems associated with them can be illuminated by the methods of graph theory. Of course, our work here can be no more than a first step in that direction. We do, certainly, at the outset, give a comprehensive survey (in Chapter 1) of the ways in which graph theory arises in connection with block designs. But thereafter we narrow and deepen our attention. Principally, we use graph theory to devise a series of efficiency criteria, which we will call, accordingly, the graphical efficiency criteria: their theory we present in Chapter 2; examples of their applications (to cyclic designs and to α designs) in Chapters 3 and 4; and their incorporation into simple algorithms for generating efficient α designs in Chapter 5. Throughout this, graph theory will recur as the main aspect of the mathematical foundation of the criteria, facilitating theory and applications alike. We also, in Chapter 6, investigate briefly one further, related, application of graph theory to block designs: we show how it is relevant to the estimation of differences between the effects of pairs of varieties.

Nevertheless, if the thesis is only a first step in the application of graph theory to block designs, we do indicate at several points - especially in Chapter 1 - where there might be possibilities for future progress along these lines.

As a preliminary, we summarise the most commonly occurring notation; all other notation will be explained fully as it arises.

We always denote matrices by upper case letters, and vectors by lower case underlined. We denote the transpose of the matrix A by A^T, and its trace by Tr(A). The n×1 vector all of whose entries are 1 we denote by $\underline{1}_n$, the n×n matrix all of whose entries are 1 by J_n, and the n×n identity matrix by I_n. (Sometimes we omit the suffix n from these, when the context makes clear what is intended.) And we have borrowed and adapted from Williams (1975) a shorthand notation for modular arithmetic: a dot above the operator ($\dot{+}$ or $\dot{-}$) means modulo v in Chapter 3, and modulo s in Chapters 4, 5, and 6.

CHAPTER 1

BACKGROUND

1.1 Introduction

The main purpose of this first Chapter is to present the background required for the rest of the thesis. That background has two facets. On the one hand, and most obviously important, is the context in which block designs are used; this we outline in Section 1.2. On the other hand, and equally essential to the approach we will adopt, are the basic concepts of graph theory; these we describe in Section 1.3. Then, in Section 1.4, we bring the two together, defining the structures which will allow us, later, to use graph theory in the search for efficient block designs.

But this Chapter has, furthermore, a subsidiary purpose: to give a general, brief, survey - also in Section 1.4 - of all the main links between graph theory and design. The intention of this survey is not only to explicate the ways in which these links have been exploited in the past, but moreover to indicate as well some possibilities which they open up for future research.

1.2 Background from Design of Experiments

We will posit throughout the thesis the following experiment, which will be the context in which all block designs[†] we consider would be used: v varieties are to be tested in b incomplete blocks,

[†] For an account of the relevant definitions, see John (1971), Chapter 11.

each with k plots, in such a way that each variety is replicated r times. There may be many ways of arranging the v varieties according to this general pattern; deciding which arrangement - that is, which design - is "best" is precisely where the problem lies. In order to be able to make this decision, it is, of course, necessary first of all to have some appropriate rules of choice. Suitable criteria have usually been derived from the normal equations for the intrablock estimation of the variety effects, and we will follow this approach here: the graphical criteria of efficiency which will form the main subject of the later Chapters will, like the customary existing criteria, be based on these equations.

In this Section, therefore, we present the equations and summarise the existing criteria. Since the theory behind these has been developed elsewhere, we will not go into it in any detail: we will simply state the results which will be of use to us later on.[†]

1.2.1 Notation and definitions

We will need the following notation.

We let the effect of the i^{th} variety be τ_i, and the vector of variety effects be $\underline{\tau} = (\tau_1, \ldots\ldots, \tau_v)^T$. Also, we let the j^{th} block effect be β_j, and the vector of block effects be $\underline{\beta} = (\beta_1, \ldots\ldots, \beta_b)^T$.

The statistical model which we will assume for the data is the

[†] The summary of existing criteria is based on Shah (1960), where also can be found much of the theory associated with them.

standard one. If y_{ij} is the yield obtained from a plot in block j which contains variety i, then this model postulates that:

$$y_{ij} = \mu + \tau_i + \beta_j + e_{ij}, \tag{1.1}$$

where μ is the overall mean, and the e_{ij} are independent normal variables with expected value 0 and variance σ^2.

We will consider for the most part - except, that is, when we make it explicitly clear otherwise - only binary designs: that is, those in which each variety occurs no more than once in each block. (It can be shown that, if $k < v$, as will usually be the case in, for example, variety trials, then an optimal design on any of the existing criteria will be binary if such exists.)

Next, we let N be the v×b incidence matrix of the design. That is, $N = (n_{ij})$, where n_{ij} is 1 if variety i occurs in block j, and is 0 if it does not.

The matrix NN^T is called the variety concurrence matrix of the design (because its entry $(NN^T)_{ij}$ is the number of times varieties i and j concur together in a block).

Then, under the side condition $\underline{1}_v^T\underline{\tau} = 0$, the normal equations for estimating $\underline{\tau}$ are:

$$C\underline{\tau} = \underline{q}, \tag{1.2}$$

where $C = rI_v - \frac{1}{k}NN^T$ and $\underline{q} = \underline{v} - \frac{1}{k}N\underline{b}$; $\underline{v}$ is the v×1 vector of variety totals; that is $\underline{v} = (V_1,......,V_v)^T$, V_i being the total yield of all plots which contain variety i; $\underline{b}$ is the b×1 vector of block totals; that is, $\underline{b} = (B_1,......,B_b)^T$, B_j being the total yield of all plots which lie in block j.

A contrast in the variety effects is an expression:

$$\underline{c}^T\underline{\tau} = \sum_{i=1}^{v} c_i\tau_i,$$

where $\underline{c}$ is such that $\underline{1}^T\underline{c} = 0$.

Finally, if the design is connected[†] (as will be all those which will interest us), then 0 is an eigenvalue of C with multiplicity one. We let the non-zero eigenvalues of $\frac{1}{r}C$ be:

$$e_1 \geq e_2 \geq e_3 \geq \ldots\ldots \geq e_{v-1}.$$

We will sometimes refer to e_{v-1} (which is the smallest non-zero eigenvalue) as e_{min}. These eigenvalues are usually called the canonical efficiency factors of the design; they are all less than or equal to 1, and greater than 0.

1.2.2 Summary of the main existing criteria of efficiency

The second part of the background from design of experiments concerns the customary criteria which have been proposed hitherto. These are usually given the names:

(a) A-criterion;

(b) D-criterion;

(c) E-criterion;

(d) S-criterion.

(a) A-criterion

Under this criterion, the aim is to maximise what is called

[†] The concept of connectedness is explained more fully later, in 1.4.3(a), and its relevance is explained in Chapter 6.

the harmonic mean efficiency factor, namely:

$$\underline{\underline{A}} = \frac{v-1}{\sum_{i=1}^{v-1} e_i^{-1}} .$$

(This can be shown to be equivalent to minimising the average variance of the difference between pairs of variety effects: see Kempthorne (1956) and Kshirsagar (1958).)

(b) <u>$\underline{\underline{D}}$-criterion</u>

This involves maximising the geometric mean efficiency factor:

$$\underline{\underline{D}} = \left(\prod_{i=1}^{v-1} e_i \right)^{\frac{1}{v-1}} .$$

(This can be shown to be equivalent to minimising the generalised variance of the estimates of $\underline{\tau}$: see Wald (1943) and Kiefer (1958).)

(c) <u>$\underline{\underline{E}}$-criterion</u>

This requires us to maximise the smallest canonical efficiency factor, e_{min} (equivalent to minimising the maximum variance of a contrast: see Ehrenfeld (1955) and Wald (1943)).

(d) <u>$\underline{\underline{S}}$-criterion</u>

This fourth criterion is widely used in practice (though its many manifestations are not immediately recognisable as being the same). However (as we will explain shortly) it has a rather more tenuous theoretical base than the $\underline{\underline{A}}$- , $\underline{\underline{D}}$- , and $\underline{\underline{E}}$- criteria.

It was first proposed by Shah (1960), and, in his formulation, involved minimising the variance of the canonical efficiency factors e_i; that is, minimising

$$\frac{1}{v-1} \sum_{i=1}^{v-1} (e_i - \bar{e})^2 ,$$

where $\bar{e}$ is the average of the e_i. But $\bar{e}$ is simply $\frac{v(k-1)}{k(v-1)}$, and so this amounts to minimising

$$\sum_{i=1}^{v-1} e_i^2 .$$

Now, $$\mathrm{Tr}(C^2) = \sum_{i,j=1}^{v} c_{ij}^2$$

(C being the matrix (c_{ij})), and since this quantity is related (by means only of scalar factors and constants) to the quantity

$$\sum_{i,j=1}^{v} (NN^T)_{ij}^2,$$

the $\underline{\underline{S}}$-criterion can also be expressed as minimising

$$\underline{\underline{S}} = \sum_{i,j=1}^{v} (NN^T)_{ij}^2 .$$

Finally, since the quantities

$$\sum_{i,j=1}^{v} (NN^T)_{ij} \text{ and } \sum_{i=1}^{v} (NN^T)_{ii}$$

are both fixed, being, respectively, rkv and rv, the criterion can assume the form in which it is most often expressed: minimise the spread of the off-diagonal entries of the concurrence matrix NN^T.

These criteria are used to compare designs for given numbers of varieties, blocks and replicates. A design which has an optimal (that is, maximal) value of $\underline{\underline{A}}$ will often be called $\underline{\underline{A}}$-optimal; similarly for each of the other criteria.

Now, the advantage of the first three of these criteria is that they have a firm theoretical foundation, being related (as we have indicated briefly above) to important statistical features of the experiment. Their disadvantage is that they are relatively complicated to evaluate, having, moreover, no immediately obvious link with the purely combinatorial properties of the layout of the varieties. On the other hand, the S-criterion is easy to calculate, and does have a transparently direct relationship with certain combinatorial aspects of the design (since, as we saw, it is equivalent to minimising the spread of concurrences between varieties). However, it lacks any rigorous theoretical justification: indeed its adoption would seem, historically, to have been based on heuristic arguments (derived, primarily, from the observation that a balanced incomplete block design is optimal under all four criteria, but derived also, through time, from widely attested experience of it as a useful guide in the selection of efficient designs).

The criteria which we will propose later in the thesis are, at once, easier to calculate than criteria (a) - (c), yet also more firmly based theoretically than (d). They will be seen to be, in a sense to be explained, a generalisation of the S-criterion, and collectively (that is, if the sequence could be continued indefinitely) a stronger condition than the A-criterion.

All these points will emerge as natural consequences of the graph-theoretic approach which we will adopt throughout; it is to introducing the necessary background for this that we now turn.

1.3 Background from Graph Theory

The recurrent mathematical theme of the thesis will be graph theory. Not only will this facilitate and clarify many of the definitions, results, and applications; it will also set the whole problem of efficiency criteria in a framework which, we will suggest, is intuitively appealing, mathematically coherent, and, in consequence, aesthetically elegant.

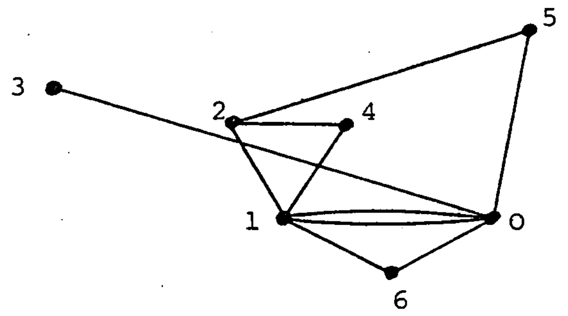
The purpose of this Section is to introduce the necessary concepts and definitions from graph theory. We will not digress into technical details: these have been treated rigorously elsewhere (for example, in Harary (1969)or Berge (1973)). All we will do here is select those parts of the subject which we will require later.

It should be emphasised at the outset that a "graph" in this context does not mean what working statisticians usually take it to mean. It has nothing to do with plotting curves against axes; it is defined, rather, as a network of points, along with lines joining pairs of these points. It is unfortunate that this semantic ambiguity has arisen, and the word "network" might, perhaps, convey more of the flavour of what is meant.[†] But the word "graph" has become so firmly established in the literature that there is no prospect of dislodging it. So we will adhere to this conventional terminology, at the risk of offending the instincts of the statistically-minded reader.

[†] At least one eminent graph theorist agrees with this: Nash-Williams.

A graph, then, is simply a collection of points (or vertices), along with a set of lines joining together certain specified pairs of points. For example, the graph in Figure 1 has 7 points (numbered 0,1,2,3,4,5,6) and 10 lines: two lines joining 0 and 1; and single lines joining each of the pairs 0 and 3, 0 and 5, 0 and 6, 1 and 2, 1 and 4, 1 and 6, 2 and 4, 2 and 5.

Figure 1



There arises at once a problem of presentation: because these sheets of paper are two-dimensional, the lines of graphs will tend to intersect each other in a misleading fashion. (For example, in the graph in Figure 1, the line 0-3 appears to intersect the lines 1-4 and 1-2.) We will therefore adopt the convention that points will be denoted by unmistakably black dots; line intersections other than at such dots will have no meaning.

First, some convenient terminology. Two points which are joined by a line are said to be adjacent, as are two lines which meet at a point. (For example, in Figure 1, the points 0 and 5 are adjacent, and the lines 0-5 and 6-0 are adjacent.) If there is more than one line joining the same two points, then we will sometimes refer to these lines collectively as multiple lines. (We would use this to refer to, for example, the two lines joining the points 0 and 1 in Figure 1.) The valency of a given point is the number of lines on which it lies. (Thus, for example, the valency of 0 in Figure 1 is 5.)

Such terminology is purely to avoid cumbersome circumlocutions. On a less mundane level are a further two notions, both of which will figure prominently later: namely, path and circuit.

A path[†] from i to j is exactly what it might be supposed to be intuitively: it is a route from i to j traversing lines of the graph. For example, in the graph in Figure 1, a path joining 5 to 4 is as follows:

5 - 2 - 1 - 4 .

A path might loop back on itself, and might traverse certain lines more than once. For example, another 5-4 path in Figure 1 is:

5 - 0 - 1 - 6 - 0 - 1 - 4.

A circuit[†] is a path which starts and finishes at the same point. For example, the following is a circuit joining 2 to itself in Figure 1:

2 - 5 - 0 - 1 - 2.

A proper circuit is one whose lines are all different (such as, in Figure 1:

2 - 4 - 1 - 2).

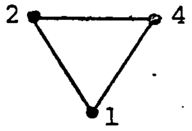
[†] The definitions of path and circuit given here are different from those given in Harary (1969), but the same as in Berge (1973). We have chosen this definition to suit our purposes here.

The length of a path or circuit is the number of lines contained in it (not, it should be noted, the number of distinct lines: each line is counted once for each of its appearances). For example, the length of the first path above is 3 (lines 5-2, 2-1, and 1-4); that of the second path is 6 (lines 5-0, 0-1, 1-6, 6-0, 0-1, and 1-4); that of the first circuit is 4 (lines 2-5, 5-0, 0-1, and 1-2); and that of the second circuit is 3 (lines 2-4, 4-1, and 1-2).

That all this does have relevance to the problems of design will be seen, later, to arise from the fact that the properties of a graph can be encapsulated in the form of a matrix: the adjacency matrix. If the graph G has n points, then its adjacency matrix is the n×n matrix whose $(i,j)^{th}$ entry is the number of lines in G which join the points i and j. (If i is the same as j, this is taken to be zero.) In other words, if we denote the adjacency matrix by (a_{ij}) or A or A(G), then a_{ij} is the number of lines i-j. Thus, for example, for the graph in Figure 1, A is as follows:

$$A = \begin{bmatrix} 0 & 2 & 0 & 1 & 0 & 1 & 1 \\ 2 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} .$$

Of considerable importance will be the fact that this matrix can be used to enumerate numbers of paths and circuits in the graph: to be precise, the number of i-j paths of length h is the $(i,j)^{th}$ entry of the matrix A^h - that is, $(A^h)_{ij}$; and the number of circuits of length h joining i to itself is $(A^h)_{ii}$. (This result can be proved by simple mathematical induction on h.) Thus, for example, in the graph in Figure 1, there are three 1-5 paths of length 2 (two paths of the form 1-0-5, and one 1-2-5), and $(A^2)_{15} = 3$. There is one slight complication to bear in mind when enumerating circuits by $(A^h)_{ii}$. This is that orientation is implicitly assumed to be significant. For example, in Figure 1, $(A^3)_{22} = 2$, reflecting the fact that the configuration:

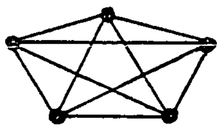


yields two 2-2 circuits of length three: namely, 2-4-1-2, and 2-1-4-2.

Finally in this Section on the background from graph theory, we mention three particular types of graph which we will use several times again.

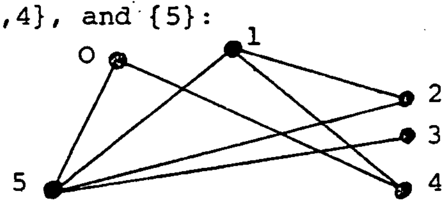
First, a connected graph is one in which every two points are joined by some path; that is, the points are all connected together by paths.

Next, the complete graph with n points is the one in which every two points are joined by a line: for example, the following is the complete graph with 5 points:



And, finally, the multipartite graph, which is slightly more complicated. In a graph of this kind, the points can be grouped into sets in such a way that no two points in the same set are joined by a line. These sets are called the sides of the graph.

For example, the following graph has three sides: namely, the sets {0,1}, {2,3,4}, and {5}:



(The grouping into sides is usually not unique. Here, another, way of grouping is {0,1}, {2,3}, and {4,5}.) This graph is called tripartite - because it has three sides. A graph which has two sides will be called bipartite.

These are now all the main definitions and ideas from graph theory which we will require repeatedly. Occasionally we will need further concepts, but it will be more convenient - and will involve far less complication - if we introduce these as they arise.

Having, therefore, sketched the relevant backgrounds from both design of experiments and graph theory, we turn next to drawing them together.

1.4 The link between Graph Theory and Design

The link between graph theory and general block designs has received only sporadic and inchoate attention in the literature. That is not to overlook the considerable volume of work that has employed combinatorial graph theory to construct balanced and partially balanced incomplete block designs. But such designs are available for only certain severely constrained values of the parameters; our interest here, in contrast, is at once more general and more unified, subsuming into a framework of graph theory the structure of block designs of any type. It is this, more general, use of graph theory which has not been investigated very much hitherto. The principal - and just about the only - paper is by Patterson and Williams (1976a), but important as it is in breaking new ground in this area, it does not progress far beyond definitions. Apart from that, there is some interesting, if not as yet fully developed, use of graph theory in recent work by Mitchell and John on optimal incomplete block designs (Mitchell and John (1976) and John and Mitchell (1977)); and there are papers on the connectedness of designs by, amongst others, Jacroux, which, although not explicitly phrased in terms of graph theory, do point in that direction.

We will describe and comment on the work of these authors as we present, in this Section, the several ways in which graph theory arises in the context of design of experiments. The main link - and the one which will concern us for most of the thesis - involves the variety concurrence graph, and this we introduce in 1.4.1. There are, also, the block concurrence graph (which we will deal with in 1.4.2), and the design graph (1.4.3). What little work that has been done in the past on the possibilities of graph theory as a tool in the design of experiments has been concerned with these three graphs (or with slightly modified forms of them), There have, furthermore, been several other, more specialised, uses of graph theory which are essentially irrelevant to the search for efficient block designs. We do, however, include these briefly as well (1.4.4) since a part of our intention in this Chapter is (as we mentioned on page 3) to give a reasonably comprehensive survey of the known - or, at least, the explicitly formulated - links between graph theory and design.

1.4.1 The variety concurrence graph

The graph which we have chosen to use most often in the thesis is the variety concurrence graph. This choice is, in a sense, arbitrary: in principle, it would have been possible to have presented all our material in terms of either the block concurrence graph or the design graph. However, there are a number of good reasons - mainly to do with ease of comprehension - which have led us to prefer the variety concurrence graph. (Occasionally, all the same, we will turn to the other two - whenever that allows for a more cogent exposition.)

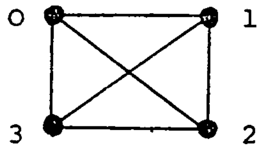
First of all, we define what the variety concurrence graph of a block design is[†]: its points correspond to the v varieties of the design, and its lines to concurrences of pairs of varieties: that is, there is a line joining points i and j for each concurrence of these varieties together in a block.

A simple example of this is the following design for 4 varieties in 6 blocks of size 3. The varieties are denoted by the numbers 0, 1, 2, and 3, and the blocks are the rows of the array:

[†] This definition is an adaption of the one given by Patterson and Williams (1976a).

0 1
1 2
2 3
3 0
0 2
1 3 .

Then the variety concurrence graph of this design is:



(Each variety concurs once with every other variety, and so each pair of varieties are joined by exactly one line.)

One immediate consequence of this definition is that the valency of each point in the variety concurrence graph is $r(k-1)$[†]: that is, each point lies on exactly $r(k-1)$ lines. To show this is straightforward: each variety lies in exactly r different blocks, and concurs in each block with $k-1$ other varieties; so each of the r blocks gives rise to $k-1$ lines which the given variety lies on, and so that variety lies on, in total, $r(k-1)$ lines. (Notice that this argument holds good regardless of whether the variety concurs more than once with some other variety: that is, regardless of whether the r sets of $k-1$ varieties determined by the blocks which contain this variety are mutually disjoint in pairs. Of course, if two varieties concur together twice, say, then there are two lines joining them in the variety concurrence graph.)

The importance of the variety concurrence graph lies in its connection with the statistical analysis of the design. On this

[†] This result is stated in Patterson and Williams (1976a).

connection is based the theory which forms the central part of the thesis, including, in particular, the development of our proposed graphical criteria of efficiency. We will elucidate this connection when we turn to these criteria in Chapter 2; all that is necessary here as background for the rest of this Chapter is to show how the adjacency matrix of this graph is related to the concurrence matrix NN^T.

This derives from the following observation, simple in itself, yet crucially important to the entire structure that links graph theory and design. For distinct varieties i and j, the entry in position (i,j) of the concurrence matrix is the number of concurrences of i and j together in blocks. But this is also, of course, the number of lines joining these varieties in the graph, since that is precisely how the graph is defined. And this number is the entry in position (i,j) of the adjacency matrix A. In other words, for i not equal to j, $(NN^T)_{ij} = a_{ij}$. Now, the diagonal entries of NN^T are each r, and the diagonal entries of A are each 0. So we have the equation:

$$A = NN^T - rI_v \,. \qquad (1.3)$$

We return to this equation in Chapter 2, where we use it to show how the adjacency matrix enters into the solution of the normal equations. This will allow us, subsequently, to demonstrate how our graphical criteria are both based on these solutions, and also related to properties of the variety concurrence graph; it will be these two features of the criteria which will underlie their important strengths.

Meanwhile, however, we continue with our general exposition of the links between graph theory and design. We divide our discussion here of the variety concurrence graph into two main sections. First, in 1.4.1(a), we investigate its structure in certain special cases, thus providing some insight into the ways in which it arises from the form of the design: we consider balanced and partially balanced designs in some detail, and cyclic and α designs rather more briefly (since we will be returning to these two later on). Then, in 1.4.1(b), we describe the most important ways in which the variety concurrence graph has been used by researchers in the past. We indicate, where appropriate, some of the limitations of previous approaches, and we refer forward, in passing, to our attempts to widen and deepen this graph's uses.

1.4.1(a) The structure of the variety concurrence graph

I Balanced incomplete block designs

The variety concurrence graph of a balanced design takes a particularly simple form - exemplified by the design we described on page 18. It will be noticed that in the graph of that design, every two points are joined by a line - reflecting the fact that each pair of varieties concur together once in some block of the design. This - or, rather, a slight generalisation of this - happens for any balanced design: every two points are joined by the same number of lines - the generalisation being that this number can be greater than one. The number is the parameter λ of the design: the parameter which has, in the literature, come to stand for the number of times each pair of varieties concur together in blocks. Following John (1971), page 220 and Patterson and Williams (1976a), λ equals $\frac{r(k-1)}{v-1}$ (which must, of course, be an integer). Thus, in our example on page 18, λ is 1, since r is 3, k is 2, and v is 4. An example in which λ is more than one is the cyclic design for 11 varieties in 11 blocks of size 5 each which has the initial block $\{0,3,5,6,7\}$. In this, λ is 2: that is, each pair of varieties concur together twice in blocks. So the variety concurrence graph has 11 points, each pair of which are joined by 2 lines.

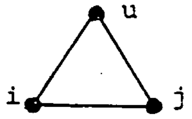
In this sense, the variety concurrence graph of a balanced design for v varieties can be described as a multiple of the complete graph on v vertices.

II Partially balanced incomplete block designs

The variety concurrence graphs of balanced designs are, therefore, particularly simple. As soon, however, as the property of balance is removed, the structure of the variety concurrence graph can become much more complex. Indeed, the whole of this thesis from Chapter 2 onwards could be regarded as a first step in the investigation of this complexity in the cases of cyclic designs and α designs. Here, we give a preliminary idea of our later methods of reasoning by looking at partially balanced designs in which every pair of varieties either concur together once, or do not concur together at all.

We will adopt the fairly standard notation used by, for example, John (1971), pages 251ff: varieties which concur together once are called first associates, and those which do not concur are called second associates. So, in the variety concurrence graph, first associates are joined by one line, and second associates are not joined by any lines. So each variety has r(k-1) first associates: its valency in the graph is, as always, r(k-1), and since it is not joined more than once to any other variety, this implies that it is joined to r(k-1) other varieties - that is, that it has r(k-1) first associates. The number of second associates of each variety is, in consequence, v-1-r(k-1). That is, in John's notation, n_1 = r(k-1) and n_2 = v-1-r(k-1).

The distinguishing feature which makes the design partially balanced is the following (John (1971), page 251): if two varieties are first associates, then the number of varieties that are both u^{th} associates of one and w^{th} associates of the other is the fixed quantity p^1_{uw} (independent of the pair of first associates chosen); likewise with second associates: the fixed quantity is denoted, this time, by p^2_{uw}. This amounts to saying that the graph is what is known as "strongly regular": that is, the number of points adjacent to both of two points that are themselves adjacent is fixed (being p^1_{11}); and the number of points adjacent to both of two points that are not adjacent is also fixed (being p^2_{11}). Indeed, p^1_{11} and p^2_{11} are the numbers of paths of length two between, respectively, two points that are adjacent and two that are not. For example, if the points i and j are adjacent, then p^1_{11} is the number of triangles to which the line i-j belongs:



One useful consequence of this observation is that it enables us to enumerate circuits of length three in the graph. (Why this is of interest for us will become clear in Chapter 2, where our graphical criteria of efficiency will be seen to be based on just such counting.) There are, altogether, $\frac{1}{2}vr(k-1)$ lines, and each of them is part of p^1_{11} triangles. So the quantity $\frac{1}{2}vr(k-1)p^1_{11}$ counts each triangle three times. So the total number of circuits of length three is $vr(k-1)p^1_{11}$. (Each diagram gives rise to six distinct circuits of length three once the different starting points and orientations are taken into account.)

III Cyclic designs and α designs

Since we will be discussing these designs later, we do no more than allude here to a couple of salient features of their variety concurrence graphs.

On the one hand, it is interesting to observe that the variety concurrence graph of a cyclic design is what Biggs (1974), page 16, has defined as a circulant graph: that is, the adjacency matrix is a circulant matrix. The proof and expansion of this is contained in Chapter 3 below (page 91).

On the other hand, the variety concurrence graph of an α design which has block size k and has s blocks in each replicate is multipartite, with k sides and with s points in each side. This is because varieties in an α design can be grouped into k columns of s varieties each, such that no pair of varieties in any column concur with each other; that is, the s varieties in a column are not joined together by any lines in the variety concurrence graph, and so they form a side of the graph.

1.4.1(b) Uses made by other researchers of the variety concurrence graph and related ideas

The variety concurrence graph has not been extensively used by other researchers. Raghavarao (1971), page 187, does refer to it briefly: he demonstrates, for example, that the variety concurrence graph of any partially balanced design with two associate classes is a strongly regular graph, and, conversely, that every strongly regular graph arises as the variety concurrence graph of some such design. (Our discussion above, pages 21-23, established a particular case of this.) He then generalises the result by generalising the concept of a strongly regular graph: he defines what he calls a strongly regular graph of order m in such a way as to characterise the variety concurrence graphs of partially balanced designs with m associate classes. The remainder of his material on the variety concurrence graph is, mostly, in this vein: concerned, ultimately, with using graph theory to help in the construction of designs that have certain specified combinatorial properties.

Indeed, it is concerns such as these which have underlain most previous uses of the variety concurrence graph in the design of experiments: establishing the existence or non-existence of balanced or partially balanced designs. However, there are two important, recent, exceptions. The one is the work of Patterson and Williams

(1976a) on efficiency factors of block designs; however, since they make less use of the variety concurrence graph than they do of the block concurrence graph and the design graph, we will postpone discussion of their work until 1.4.2 and 1.4.3 below. The other exception, which we turn to now, is the work on Regular Graph Designs, a concept which has been used by a number of authors in the search for efficient block designs.

The concept was introduced by Mitchell and John (1976). They define a Regular Graph Design to be one in which every pair of varieties concur together either λ or $\lambda+1$ times (for some fixed integer λ which is the same for each pair). Thus, the variety concurrence graph of a Regular Graph Design has every pair of points joined by either λ or $\lambda+1$ lines. Strictly speaking, the graph which Mitchell and John consider is not the variety concurrence graph, but, rather, a slightly modified form of it: since all they are interested in is the fact that the concurrences differ by one, they as it were ignore λ of the lines between each pair, thus defining a graph (which we will call G_1) in which each pair of points are either not joined, or else joined by one line. Put more formally: instead of considering the variety concurrence graph - which has adjacency matrix $A = NN^T - rI$ - they consider the graph G_1 whose adjacency matrix A_1 is obtained by subtracting λ from each of the off-diagonal entries of A; that is,

$$A_1 = NN^T - rI - \lambda(J - I) \qquad (1.4)$$

Before we go into the uses to which this has been put, there are some important - critical - remarks to be made on Mitchell and

John's nomenclature, which, we will argue, is unfortunate in that it is the source of some potential confusion. The reason which they give for choosing the name "Regular Graph Design" is that in the graph G_1 each of the points is adjacent to the same number - that is, $r(k-1)-\lambda(v-1)$ - of other points. This choice is, however, both misguided and misleading.

It is misguided in that it misinterprets the standard definition of a regular graph from graph theory: that defines a graph to be regular if each of its points lies on the same number of lines - regardless of whether or not each point is adjacent to the same number of other points. Of course, if there are no multiple lines - as in the graph G_1 of a Regular Graph Design - then a regular graph in the standard graph theory sense would be regular also in the idiosyncratic sense adopted by Mitchell and John. (No doubt, in fact, this is what underlay the confusion.) But when there are multiple lines, the two senses are not the same.

Now, this is no mere quibble: the unfortunate aspect of the confusion is that the more commonly occurring definition of a regular graph tends either to render the nomenclature of Mitchell and John inexplicable, or else to cause the category of Regular Graph Designs to become redundant. For, it is not only Regular Graph Designs that have associated with them a graph G_1: any design for v varieties in b blocks of size k gives rise to such a graph by a simple extension of the ideas of Mitchell and John: if λ is the smallest value of the off-diagonal entries of the variety adjacency matrix A, then G_1 would have adjacency matrix $A-\lambda(J-I)$. (This definition comes down to the same as that of Mitchell and John when the off-diagonal entries of A differ by only 1.) Then, since the variety concurrence graph of any such design is regular (in the standard sense), this graph G_1 is regular too (also in the standard sense). So, on the one hand, it seems rather perverse to use the term Regular Graph Design to refer only to those designs in which the graph G_1 happens to have no multiple lines; yet, on the other hand, the only natural way of avoiding this perversity would be to extend the category of Regular Graph Designs so that it would become coterminous with the class of all designs for v varieties in b blocks of size k - in which case the category would become redundant.

In this sense, therefore, the usage is misguided, resting, as it does, on a misinterpretation of a definition in graph theory. And, moreover, it is misleading: not only because it might be supposed from it that a Regular Graph Design was one whose graph G_1 was regular in the standard sense we have just described; but also because it lays emphasis on a property of the graph G_1 which is irrelevant to the purpose. The fact that each point is adjacent to the same number of other points (or, for that matter, lies on the same number of lines) has got nothing to do with the property that is supposed to be the defining feature of designs in this category: namely, that any two points concur with each other either λ or $\lambda+1$ times.

All that said, however, we will retain this usage - not out of preference, but because it has become firmly established in the design literature. (Though, of course, the perpetuation of this usage does violate an even more firmly established convention in graph theory: the standard definition of "regular".) The reason why we have analysed at some length what we consider to be an unfortunate solecism is due to our feeling that since names in statistical theory are presumably supposed to make difficult concepts easy to remember, the features they encapsulate ought to be well-defined and essential, rather than vague and incidental.

Mitchell and John, themselves, do not use the concept of Regular Graph Designs in any detailed way. They merely use it as a convenient method of expressing a conjecture on which they base a search for optimal designs: they conjecture that if there does exist a Regular Graph Design in a given class, then every A-optimal design in that class is a Regular Graph Design. (They also make analogous conjectures for D-optimal and E-optimal designs.) They use this conjecture to facilitate the search for efficient designs by searching, first of all, for regular graphs (according to their meaning of the term: but since the graphs they look at have no multiple lines, this amounts, as we have said, to the same as the standard notion). For each regular graph, they calculate what would have to be the concurrence matrix of any design that gave rise to that graph (by reversing the equation (1.4): it can be shown that λ in that equation must be the integer part of the quantity $\frac{r(k-1)}{v-1}$). From this, they work out what would be the corresponding value of A. Of course, there is no guarantee that there does exist a design with this particular concurrence graph and value of A. So the final stage in their search is to take that concurrence matrix which has yielded the best value of A, and look for a design that would have given rise to it; if no such design exists, then they do the same for the matrix which yielded the second best value of $\underline{\underline{A}}$; and so on. If their conjecture is correct, then this process will always lead to an $\underline{\underline{A}}$-optimal design.

(We will return to this conjecture later, when we will show that it is equivalent to a conjecture of our own which we arrive at in a rather different way by means of graph theory: a way, in fact, which allows it to be generalised naturally. See, in particular, 2.5. We will also show that Regular Graph Designs are optimal according to the $\underline{\underline{S}}$-criterion.)

This technique of Mitchell and John - fruitful as it has turned out to be in their hands - does not go far beyond a merely terminological use of graph theory, enabling them to draw on catalogues of regular graphs as part of their search for efficient designs. Their terminology does, however, allow a very much more concise description of the algorithm which they used for this search: see their report (1976).

At least, however, they do draw on some graph theory terminology (even if waywardly) and results. Work by other authors on Regular Graph Designs has, in contrast, made no use of graph theory at all. Indeed, Mitchell and John were the first to notice the link with graph theory. Thus, for example, although the results of Conniffe and Stone (1974,1975) and of Shah et al (1976) establish the $\underline{\underline{A}}$-optimality of certain Regular Graph Designs, these authors did not phrase their theory in that way. (They talk, equivalently but more cumbersomely, of designs in which the off-diagonal entries of the variety concurrence matrix NN^T differ by at most 1.) Likewise with the work of Takeuchi (1961), on the E-optimality of certain group divisible designs that happen also to be Regular Graph Designs. Jacroux (1980), on the other hand - writing after the work of Mitchell and John - does employ the term "Regular Graph Design", but he makes no explicit use of graph theory.

Even papers by Williams, Patterson and John (1976,1977) - despite reference to the pioneering introduction of a graph theory approach by the first two of these authors in 1976 (which we describe in the course of 1.4.2 and 1.4.3), and despite using the results which (as we have outlined) the third author obtained along with Mitchell - do not, regrettably, develop the use of graph theory any further.

So the variety concurrence graph has been used only occasionally in the literature, and then only as a source of a convenient (if at times - in the case of Regular Graph Designs - potentially confusing) terminology.

1.4.2 The block concurrence graph

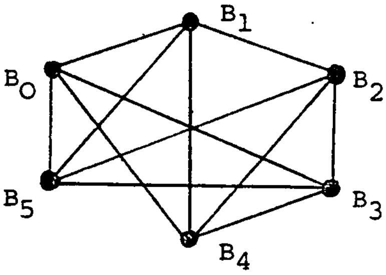
The second graph which we define is the block concurrence graph, which we will sometimes denote by G_B. This is, perhaps, the least important of the three, for reasons which we will explain, but it has been used productively by Patterson and Williams (1976a).

The points this time are the b blocks of the design, and the lines correspond to varieties which pairs of blocks have in common. That is to say, if B_0 and B_1 are any two blocks, then they are joined by a line for each variety that lies in both of them.

To illustrate this, we return to the design which we used to exemplify the variety concurrence graph (page 18). The blocks in that design are as follows:

B_0: 0 1
B_1: 1 2
B_2: 2 3
B_3: 3 0
B_4: 0 2
B_5: 1 3 .

Then, for example, blocks B_0 and B_1 are joined by exactly one line in the block concurrence graph, since they share exactly one variety (namely, the variety 1). The graph is:



A number of interesting properties follow immediately from this definition. The valency of each point is k(r-1)[†]: there are k varieties in any given block, and each of them lies also in r-1 other blocks, thus producing r-1 lines which the given block lies on. If the design has no multiple concurrences, then no two blocks are joined by more than one line: if they were, then the two varieties which produced these two lines would concur in both these blocks. And if the design is resolvable, then the block concurrence graph is multipartite: no variety occurs more than once in any replicate,

[†] This result is stated in Patterson and Williams (1976a).

and so the blocks of a replicate form a side of the graph.

But these - like most of the properties of the block concurrence graph - follow also from a far more general result which, in theory, renders this graph superfluous: the block concurrence graph is, simply, the variety concurrence graph of the dual design. The most concise way of establishing this is through the corresponding adjacency matrices. Recall that the adjacency matrix of the variety concurrence graph is $A = NN^T - rI$, where N is the $v\times b$ incidence matrix of the design. Then, since the incidence matrix of the dual design is N^T (see, for example, Raghavarao (1971), page 199), it follows that the variety concurrence graph of the dual design is $N^TN - kI$. (In the dual, the replication parameter is k.) But this is, also, the adjacency matrix of the block concurrence graph of the original design†: an off-diagonal entry $(N^TN)_{ij}$ can easily be shown to record the number of varieties which the blocks numbered i and j have in common, and the diagonal entries of N^TN are all k. This establishes the result.

This equivalence does mean that the concept of the block concurrence graph is, perhaps, unnecessary: any results concerning it could be re-expressed in terms of the variety concurrence graph of the dual. Thus, for example, the three results we presented immediately above could be deduced in this fashion. That the valency of any point is $k(r-1)$ follows from the result which we established on page 18 concerning the valencies of the points in the variety concurrence graph, along with the fact that the parameters r and k

†This result is stated in Patterson and Williams (1976a).

are interchanged in the dual. That there are no multiple lines in the block concurrence graph whenever there are no multiple concurrences in the design can be deduced by way of the dual design by, simply, appropriately re-phrasing the proof we gave above. And that the block concurrence graph of a resolvable design is multipartite follows by applying to the dual design the fact that the variety concurrence graph is multipartite whenever the varieties can be resolved into k columns such that no two varieties in the same column concur. (This is the case in, for example, α designs, as we pointed out on page 23 above.)

Nevertheless, although, in a sense, therefore, the concept of the block concurrence graph is redundant - and although, consequently, it is the least important of the three graphs which we describe in this Chapter - it is, all the same, useful. It is useful, especially, insofar as it allows some more convenient notation: usually, for example, the number of blocks will be considerably less than the number of varieties, and so the block concurrence graph with its b points will be smaller, and therefore more manageable, than the variety concurrence graph with its v points.

The block concurrence graph is useful, also, sometimes, for deriving results which it would be considerably more difficult to prove using the variety concurrence graph. One illustration of this is contained in 2.6.1. Another is in the work of Patterson and Williams (1976a). Using the block concurrence graph, they produce two important results, the first leading to the second.

First, they establish a relationship between the harmonic mean efficiency factors of a design and its dual: denoting these efficiency factors by, respectively, $\underline{\underline{A}}$ and $\underline{\underline{A}}_d$, they show that:

$$\underline{\underline{A}} = \frac{v-1}{v-b+(b-1)\underline{\underline{A}}_d^{-1}} \quad .$$

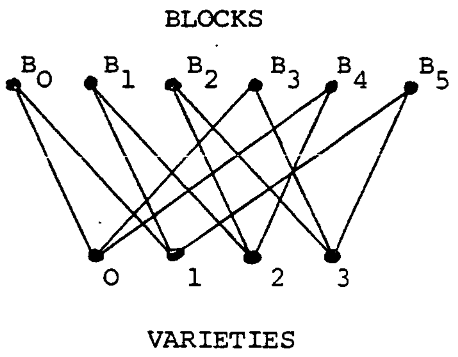
One interesting (but less important) consequence of this first result is that the design itself is $\underline{\underline{A}}$-optimal if the dual design is $\underline{\underline{A}}$-optimal. In particular, this means that the design is optimal if the dual is balanced - or, equivalently, if the block concurrence graph is a multiple of the complete graph.

But the use which Patterson and Williams make of this first result has potentially wider implications: they derive from it upper bounds for the harmonic mean efficiency factor of any block design. Their aim is to facilitate the search for efficient designs: a design which attains the upper bound will be known to be optimal, and one which is near it, nearly optimal. The authors expand these results elsewhere (Williams and Patterson (1977)): they incorporate the upper bounds they obtained by graph theory methods as the basis of a series of increasingly tight (and, therefore, more useful) upper bounds for the harmonic mean efficiency factor. We will not go into details here, as we do not intend to develop this topic in the thesis. It should, however, already be clear - even from what we have described - that this work by Patterson and Williams provides a cogent illustration of how graph theory can illuminate the problems of design.

1.4.3 The design graph

The third graph we describe in this Chapter is the design graph, which we will sometimes denote by G_D. This graph has a rather more complicated structure than either of those we have presented so far. It is bipartite, the points of one side being the varieties of the design, and those of the other being the blocks. Lines correspond to plots - that is, to incidence of varieties with blocks: a variety is joined to a block if the variety lies in a plot of the block. Thus, in a binary design, this graph has no multiple lines: no variety occurs more than once in any block.

To illustrate this definition, we will, again, take the example which we used for the variety concurrence graph and the block concurrence graph. This design is listed on page 31. The two sides of the design graph here consist of the set of four varieties and the set of six blocks. A line joins a variety to a block if the variety lies in the block: thus, for example, a line joins variety O to block B_O since O lies in B_O. The whole graph is:



Some initial properties of the design graph follow immediately from the definition. The valency of each variety is r (since each variety occurs in r plots), and the valency of each block is k (since each block has k plots). If the design is resolvable, then the side consisting of the blocks can be divided into r subsets (corresponding to the replicates) in such a way that each variety is joined to exactly one block in each subset. These properties have been established by Patterson and Williams (1976a), who also derive the form of the adjacency matrix of the design graph: it is

$$A(G_D) = \begin{bmatrix} O & N \\ N^T & O \end{bmatrix} .$$

(This is easy: no varieties are joined to each other - hence the top left-hand segment of zeroes - and neither are any two blocks - hence the bottom right-hand zeroes; and variety i is adjacent to block j if and only if it lies in that block, which happens precisely when $n_{ij} = 1$.)

This graph could enable some of the concepts which we presented earlier to be re-stated in simpler forms. Thus, for example, as Patterson and Williams point out, the dual design is obtained by interchanging the meaning of the two sides.

There are many more such ramifications of the design graph, offering potentially fruitful areas of research. We will concentrate on describing, briefly, two of its principal uses in the literature to date. On the one hand (in (a)), we will describe how it is relevant to the concept of $(\underline{\underline{M}},\underline{\underline{S}})$-optimality (as developed by Shah, and by Eccleston and Hedayat); and, on the other hand (in (b)), we will outline how Patterson and Williams use this graph to help in the construction of resolvable two-replicate designs.

1.4.3(a) The design graph and $(\underline{M},\underline{S})$-optimality

The link between the design graph and $(\underline{M},\underline{S})$-optinality has been established by Eccleston and Hedayat (1974) (and by Jacroux (1980)), although they do not phrase their results explicitly in terms of graph theory.[†] They define three forms of "connectedness" for a block design - local, global, and pseudo-global - and then establish that in certain very general circumstances any $(\underline{M},\underline{S})$-optimal design possesses one or other of these properties. We will show here no more than that these three concepts have a natural interpretation in terms of the design graph: there is probably scope in this area for a great deal of further research invoking graph theory.

A locally connected design in the terminology of Eccleston and Hedayat is defined to be the same as a connected design in the terminology of Bose (1947). And Bose's definition of connectedness has a very simple interpretation in graph theory terms: a design is connected in his sense if, and only if, its design graph is connected in the graph theory sense. (This is obvious: in his definition, a design is connected if for each two varieties there is what he calls

[†] $(\underline{M},\underline{S})$-optimality is a generalisation - invented by Eccleston and Hedayat - of $\underline{S}$-optinality: it involves selecting first of all those designs which maximise $Tr(C)$, and then choosing among them those which minimise $Tr(C^2)$. These two types of optimality are equivalent for all classes of designs which we consider in the thesis: when each variety is replicated the same number of times and each block has the same number of plots, the quantity $Tr(C)$ is fixed.

a "chain" joining them; and this concept of a chain is precisely the same as that of a path in the design graph: it consists of alternating blocks and varieties such that consecutive elements in it are associated by design incidence - that is, are adjacent in the design graph.)

Global connectedness is a stronger concept than local connectedness. A design is said by Eccleston and Hedayat to be globally connected if each pair of its varieties is globally connected; and two varieties i and j are globally connected if (in their words, page 1241),

> "each replicate of i is connected by a chain, as defined by Bose(1947), to each replicate of j."

This means - since lines in the design graph correspond to plots in the design - that given any two lines on which lie i and j respectively, there is an i-j path including them; that is, more formally, if i lies on the line x in the design graph, and if j lies on the line y, then there is a path joining i and j in the design graph which has x as its first line and y as its last.

Finally, pseudo-global connectedness is weaker than global connectedness, but still stronger than local connectedness. Analogously to global connectedness, a design is said to be pseudo-globally connected if each pair of its varieties is pseudo-globally connected; and two of its varieties i and j are pseudo-globally connected if (page 1242),

> "each replicate of i is connected by a chain, as defined by Bose, to at least one replicate of j and vice versa."

This means - proceeding as for global connectedness - that given any line in the design graph which i lies on, there is an i-j path which includes it, and likewise for any line which j lies on.

So the three concepts of connectedness as defined by Eccleston and Hedayat correspond to properties of the design graph. That this is so does suggest that graph theory techniques might enable a fruitful development of their ideas; it would, at any rate, be an interesting line of research.

1.4.3(b) The design graph and resolvable two-replicate designs

The second aspect of the design graph which we describe concerns the use made of it by Patterson and Williams (1976a) to help in the construction of resolvable two-replicate designs. Their work in this area constitutes a revealing example of how graph theory can simplify concepts that already exist in design theory - and can, besides, point to ways in which they can be generalised. These authors show that certain notions previously advanced by Bose and Nair (1962) can be expressed in a particularly simple form by means of the design graph; and they show, furthermore, that this re-interpretation indicates how these notions can be naturally extended.

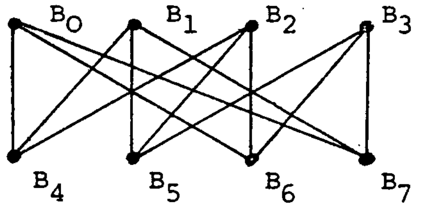
It is easier, in fact, to describe the generalisation first. It rests on their definition of what they call the "contraction" of a resolvable two-replicate design, which they present in terms of graphs as follows. The block concurrence graph of a resolvable two-replicate design is bipartite (as we observed on page 32): if there are s blocks in each replicate, then there are s points in each side, each having valency k (which is what k(r-1) becomes when r is 2). Hence that same graph can be taken as the design graph of a design for s varieties in s blocks of size k: the points of the first side correspond to varieties, and those of the second to blocks. It is this design which is the contraction.

For example, consider the following resolvable two-replicate design for 12 varieties in 8 blocks of size 3:

Replicate 0
Block B_0: 0 4 8
Block B_1: 1 5 9
Block B_2: 2 6 10
Block B_3: 3 7 11

Replicate 1
Block B_4: 0 5 10
Block B_5: 1 6 11
Block B_6: 2 7 8
Block B_7: 3 4 9 .

The block concurrence graph of this design is:



So this is also the design graph of the contraction, which has 4 varieties in 4 blocks of size 3 each. The varieties of the contraction correspond to the points labelled B_0, B_1, B_2, and B_3 in this diagram, and the blocks of the contraction to the points labelled B_4, B_5, B_6, and B_7. So, re-labelling these varieties as 0, 1, 2, and 3, the contraction is:

0 1 2
1 2 3
2 3 0
3 0 1 .

The designs of Bose and Nair can be best described, now, by employing the above terminology: their method of construction amounts to, in effect, (as Patterson and Williams show) selecting as the contraction an appropriately sized design that is balanced. Thus the use of the contraction has simplified and generalised Bose and Nair's ideas.

This not only affords an interesting illustration of the way in which graph theory can illuminate the structure of designs; it also, more practically, helps in the search for efficient resolvable two-replicate designs. For there exists a simple relationship (derived by Patterson and Williams) between the harmonic mean efficiency factors of such a design and its contraction. This is established through the two adjacency matrices. The design incidence matrix of the resolvable two-replicate design can be written $N = (N_O, N_1)$, where N_O is the v×s incidence matrix corresponding to the replicate O, and N_1 is that corresponding to the replicate 1. Then the adjacency matrix of the block concurrence graph is:

$$A(G_B) = \begin{bmatrix} O & N_O^T N_1 \\ N_1^T N_O & O \end{bmatrix}$$

(since $N_O^T N_O = kI_s = N_1^T N_1$).

Patterson and Williams use this - along with the further result that the adjacency matrix of the variety concurrence graph of the contraction is $N_O^T N_1 N_1^T N_O - kI_s$ - to show that if the harmonic mean efficiency factors of the design and its contraction are, respectively, $\underline{\underline{A}}$ and $\underline{\underline{A}}_c$, then they are related by the expression:

$$\underline{\underline{A}} = \frac{v-1}{v-2s+1+4(s-1)\underline{\underline{A}}_c^{-1}} \quad .$$

In later papers, these authors, along with John (Williams, Patterson and John (1976, 1977)), use this formula to help in the search for efficient resolvable two-replicate designs: it enables this search to be reduced to the potentially far simpler one of looking for efficient designs for s varieties in s blocks of size k. We return to these ideas later (2.6.1 and 4.4.3), when we relate them to our graphical criteria of efficiency.

The three graphs which we have now presented - the variety concurrence graph, the block concurrence graph, and the design graph - offer many promising lines of research. A number of these, of course, we follow up in some detail in the main part of the thesis. But two such possibilities which we do not return to are worth mentioning very briefly here.

The first concerns upper bounds for the smallest canonical efficiency factor (that is, for the $\underline{\underline{E}}$ criterion). The efficiency factors e_i are related in a straightforward manner to the eigenvalues of the variety adjacency matrix A (see later, 2.3). Now, it turns out that certain approximations to these eigenvalues can be had in terms of the eigenvalues of the adjacency matrices of subgraphs of the variety concurrence graph.[†] (See Biggs (1974).) Initial results which we have obtained - admittedly in a

[†] A subgraph of a graph is exactly what it sounds like: a collection of some of its points, together with all the lines which join pairs of points in this collection.

rather sporadic, unsystematic fashion - suggest that these approximations can provide fairly useful upper bounds for the smallest canonical efficiency factor: for example, in several individual cases we have looked at, they can be used to show that a design with double concurrences cannot have as high a value of its smallest canonical efficiency factor as the best known design with no double concurrences.

Similar use can be made of the block concurrence graph and the design graph.

The second possibility of further research concerns the **D**-criterion. This criterion turns to be directly linked to the number of spanning trees in the variety concurrence graph.[†] (The principal graph theory result of use here is known as the Matrix-Tree Theorem: see Biggs (1974), Theorem 6.3.) The main implication of this link is that the more spanning trees there are, the higher is the value of the **D**-criterion; this allows, for example, the derivation of results on the **D**-criterion that are analogous to those on the **S**-criterion obtained by Eccleston and Hedayat (1974).

A similar link can be established between the **D**-criterion and the numbers of spanning trees in the block concurrence graph and the design graph.

1.4.4 Further, miscellaneous, uses of graph theory in design

The three graphs which we described in 1.4.1, 1.4.2, and 1.4.3

[†] A tree is a connected subgraph (see footnote on page 42) with no proper circuits; a spanning tree is a tree which contains all the points.

will be the only ones which we will use later on. However, there have been one or two other applications of graph theory to design which have little to do with any of these three graphs. Although, as we have already mentioned on page 16, these applications are irrelevant to what will mainly concern us in the thesis (that is, the search for efficient block designs), we will, nevertheless, for the sake of completeness, briefly describe them here.

At the most elementary level, a number of authors writing on cyclic designs have used graph digrams as an aid to comprehension. This is found in, for example, the catalogue of cyclic designs (John, Wolock and David (1972)), and in the original paper by David and Wolock (1965): they represent the varieties by points spaced round a circle to help explain the cyclic method of construction.[†]

However, this use of graph diagrams does not go beyond mere illustration. Of more importance are three further areas in which graph theory has been put to some use: the construction of balanced and partially balanced designs; an application by Williams (1952) to designs for serially correlated observations; and the use of graph theory by Patterson to extend methods of constructing change-over designs that were first advanced by Quenouille. In the remainder of this Chapter, we will ouline each of these three.

1.4.4(a) The construction of balanced and partially balanced designs

We have already mentioned that the variety concurrence graph can help in the construction of balanced and partially balanced

[†] We return to the graphs of cyclic designs later, in Chapter 3.

designs (as in, for example, Raghavarao (1971)). There have also been a number of other ways in which graph theory has been used for this purpose. But this work has limited relevance to the kind of designs which concern us here - since its bias is mathematical, not statistical.

Typical of this has been the use of finite geometries to construct balanced designs: the points of the geometry are the varieties and the lines are the blocks. (See, for example, Raghavarao (1971) or John (1971); also Ray Chauduri and Wilson (1970) and Ray Chaudhuri (1970).) Indeed, in this connection there is a paper by di Paola (1966) with the same title as this thesis.

But that congruence of titles points to one of the principal limitations of the mathematicians' approach. A "block design" in their terminology is merely a balanced block design in ours; and the graph theory methods which mathematicians such as di Paola have developed for constructing designs that are balanced are not readily adaptable for constructing designs that are not: since these methods are not directly related to the statistical purposes of the designs - estimating effects or analysis of variance or fitting regression lines, for instance - they do not indicate how graph theory could be employed to construct general block designs of high efficiency.

1.4.4(b) The construction of designs for serially correlated observations

The second of the miscellaneous uses of graph theory which we refer to was developed by Williams (1952). We describe this in slightly more detail in order to suggest ways in which graph theory could be used to extend further his line of research. He was interested in constructing designs for testing serially correlated observations: that is, in experiments where the observations form a one-dimensionsal sequence. In such a situation, it is intuitively desirable - and, indeed, Williams shows it also to be sensible statistically - to ensure that every two varieties are beside each other equally often: that is, occur with consecutive observations equally often. The method of construction which Williams uses to achieve this is based on graph theory. He constructs a graph with v points, which are to correspond to the v varieties; a line joins two points for each occasion on which they are to occur beside each other in the sequence: so, if each two are to occur beside each other c times, then they are joined by c lines in the graph. Such a graph can, of course, be constructed for any number v of varieties and for any value of c; the problem of constructing the design then becomes - as Williams shows - that of finding what in graph theory terminology is called an Eulerian circuit in this graph: that is, a circuit which traverses each line exactly once. Such a circuit will always exist if the valencies of the points of the graph are all even (that is, if c(v-1) is even), but will not exist if these valencies are odd (by a theormem of Euler - see Harary (1969), page 64).

This application of graph theory is rather closer to our interests than the one we described in 1.4.4(a). Indeed, the graph which Williams considers can be interpreted as the variety concurrence graph of a design in which each consecutive pair in the sequence forms a block of size two. Constructing designs of the type Williams is interested in is then equivalent to constructing balanced incomplete block designs for v varieties in $\frac{1}{2}cv(v-1)$ blocks of size 2. However, this interpretation is only possible when the Markov process which forms the basis of the statistical model for the design is assumed to be of the first order - that is, when there is only assumed to be correlation between consecutive observations. Complications arise as soon as the model is extended.

There is, nevertheless, probably scope for further use of graph theory in this area. As an illustration, we discuss a problem which Williams only raises and does not investigate. Ideally, what is needed is a sequence in which the varieties are arranged in complete blocks: that is, which can be divided up into portions of v consecutive observations, such that each portion contains each variety exactly once. The problem is that the method of constructing the design by means of an Eulerian circuit does not guarantee that the sequence will take this form. What would guarantee it is some method of splitting up the set of lines of the graph into $\frac{1}{2}c(v-1)$ disjoint spanning circuits: that is, into circuits each of which passed through each point exactly once, and which did not have any lines in common. The circuits could then be put together to form an Eulerian circuit, and hence a design of the desired type.

Such circuits are called Hamiltonion in graph theory. So, in this graph theory terminology, the problem is to find $\frac{1}{2}c(v-1)$ disjoint Hamiltonian circuits. And this problem can be solved. In those cases in which the number v of points is odd, the solution is a straightforward generalisation of a result given by Deo (1974), Theorem 2.8, page 33: he shows how to decompose the complete graph with v points into ½(v-1) disjoint Hamiltonian circuits; the graph we have here is composed of c copies of the complete graph, and Deo's technique can be applied to each of these in turn. In those cases where v is even, on the other hand, the solution is not quite so straightforward, but nevertheless can be constructed by a rather more subtle use of Deo's method. (A necessary condition for the existence of a solution in this case is the same as the necessary condition for the existence of an Eulerian circuit: that c(v-1) be even, which, since v-1 is odd, is the same as requiring that c be even.)

1.4.4(c) The construction of change-over designs

The last of the three miscellaneous applications of graph theory which we mention is, in a sense, similar to Williams's method: Patterson (1973) uses graphs to extend a method of constructing change-over designs which was originally devised (without using graph theory) by Quenouille. In these designs, v varieties are to be tested over 2v periods on v^2 subjects in such a way that each subject receives each variety twice in the course of the 2v periods, and that each combination of varieties in consecutive periods occurs with exactly one subject. Patterson shows that the enumeration of these designs is equivalent to the enumeration of Eulerian circuits in certain directed graphs.[†]

[†] A directed graph is one in which an arrow (indicating direction) is associated with each line. An Eulerian circuit in such a graph is, analogously to undirected graphs, a circuit which traverses each directed line once in the direction of the arrow.

CHAPTER 2

GRAPHICAL CRITERIA OF EFFICIENCY

2,1 Introduction

Having surveyed, in Chapter 1, the links between graph theory and design of experiments, we now concentrate our attention on one area in particular: on developing a set of criteria for assessing the efficiencies of block designs. These criteria are based on properties of the variety concurrence graph. From that basis - as we will demonstrate in this and later Chapters - stems their intuitive appeal, their mathematical justification, and, pre-eminently, the facility with which they can be applied in practice.

Before we present the full mathematical details of these graphical criteria, we will introduce them more informally. We will explain (in 2.2) how they arise quite naturally, if heuristically, in the context of one of the principal purposes of the design: namely, the estimation of the differences between variety effects. Only then (in 2.3) will we show that they can be given a firmer mathematical foundation, by examining certain links between the variety concurrence graph, the solution of the normal equations of the design, and the $\underline{\underline{A}}$-criterion. Subsequently (in 2.4), we will discuss the strengths - both mathematical and intuitive - of the graphical criteria, especially as they compare with the customary criteria. Next (in 2.5), we will put forward a conjecture concerning the way in which the graphical criteria can be best used to help construct efficient designs. Finally (in 2.6) we will narrow the discussion, establishing two further results which further justify the graphical criteria being applied to resolvable two-replicate designs.

2.2 The intuitive background of the graphical criteria: paths, circuits, and the estimation of variety differences

That the graphical criteria will make sense from a statistical point of view will derive, broadly speaking, from the connection between, on the one hand, the estimation of differences between variety effects, and, on the other, paths and circuits in the variety concurrence graph. For, paths in this graph correspond naturally to unbiased estimators of these differences. To be precise, if i and j are any two varieties, then each path joining them gives rise as follows to an unbiased estimator of the difference $\tau_i - \tau_j$.[†]

Let π be any such path: that is, π is of the form:

$$i = i_0 - i_1 - i_2 - \dots\dots - i_n = j .$$

In other words, each of the lines $i_u - i_{u+1}$ in this path corresponds to a concurrence of the varieties i_u and i_{u+1} in some block, which we will call, for convenience, block h_u. Then the standard model (see equation (1.1)) gives us that:

$$y_{i_u h_u} = \mu + \tau_{i_u} + \beta_{h_u} + e_{i_u h_u} ,$$

and that: $$y_{i_{u+1} h_u} = \mu + \tau_{i_{u+1}} + \beta_{h_u} + e_{i_{u+1} h_u} .$$

[†] The observations which follow are an adaptation into the language of graphs of certain remarks made originally by Bose (1947).

We use these equations to show that the sum:

$$\sum_{u=0}^{n-1} (y_{i_u h_u} - y_{i_{u+1} h_u}) ,$$

which we will call T , is an unbiased estimator of the difference we are interested in: namely, the difference $\tau_i - \tau_j$.

Subtracting the equation for $y_{i_{u+1} h_u}$ from that for $y_{i_u h_u}$, we get that for each u = 0,1,...,n-1,

$$y_{i_u h_u} - y_{i_{u+1} h_u} = \tau_{i_u} - \tau_{i_{u+1}} + e_{i_u h_u} - e_{i_{u+1} h_u} ,$$

and so that the expected value of $y_{i_u h_u} - y_{i_{u+1} h_u}$ is $\tau_{i_u} - \tau_{i_{u+1}}$. From this the result follows, for we now get that the expected value of T is $\tau_i - \tau_j$, since it equals

$$\sum_{u=0}^{n-1} (\tau_{i_u} - \tau_{i_{u+1}}) ,$$

which equals $\tau_{i_0} - \tau_{i_n}$, which is $\tau_i - \tau_j$.

That is, the difference between the effects of varieties i and j can be estimated by the quantity T , which is a function of the yields of the individual plots. Thus, each i-j path in the variety concurrence graph gives rise to an estimator T of the difference $\tau_i - \tau_j$.

This observation suggests the following, initially intuitive, speculation, which will lead us heuristically to the graphical criteria: the more i-j paths there are in the graph, the more information the design gives about the difference $\tau_i - \tau_j$. That is, the more i-j paths there are, the greater the precision (or the lower the variance) with which the design estimates this difference.

It transpires that this intuitive speculation can actually be fairly firmly grounded in theory.[†] What concerns us here is not so much this speculation itself as what it suggests about the significance of circuits in the graph in assessing the overall quality of the design. Circuits are, in a sense, redundant paths: because they merely join a variety back to itself, they contribute nothing to the capacity of the design to estimate differences between variety effects. So it might be surmised that a "good" design will tend to have "few" of these circuits.

Put differently, and more mathematically, the speculation suggests that if we want to maximise the information about all variety differences, then we should try to maximise the number of paths which join pairs of varieties in the variety concurrence graph. But, for fixed h, the total number of paths of length h is simply half the sum of the off-diagonal entries of the matrix A^h. (This is a direct consequence of the result on page 13 and the fact that A is a symmetric matrix.) Furthermore, the total number of circuits of length h (bearing in mind, again, the results on page 13) is the sum of the diagonal entries of A^h - that is, $Tr(A^h)$. So, since the sum of all entries of A^h is fixed (being $vr^h(k-1)^h$ - a consequence of equation (1.3)), the process of maximising the number of paths of length h joining pairs of points is exactly the same process as minimising the number of circuits of length h.

[†] See Chapter 6. The development of the graphical criteria here is entirely independent of that theory.

In short, this speculative reasoning suggests that if we want to maximise the information about differences between variety effects, then we should try to minimise the number of circuits (if possible of all lengths h) contained in the variety concurrence graph.

It is this aim, in essence, which underlies the graphical criteria: they will involve enumerating circuits in the graph. That the justification of these criteria is rather more solid than the simple heuristic reasoning we have given here we establish next, in 2.3. But their origins in the links between paths and the estimation of variety differences remains one of their most compelling, most intuitively appealing, attractions.

2.3 The mathematical background of the graphical criteria: the variety concurrence graph, the normal equations, and the A-criterion

Not, indeed, that the mathematical details do not also provide some further intuitive cogency. For they, too, are based in certain links between the variety concurrence graph and the design: this time, the relevance of the adjacency matrix to solving the normal equations. We present that in 2.3.1. Then, in 2.3.2, we show how this enables us to establish a connection between the A-criterion and the graph, from which, in turn, we will derive the graphical criteria more formally.

2.3.1 The variety concurrence graph and the normal equations

The relevance of the adjacency matrix to the normal equations rests on the equation (1.3), which linked that matrix to the concurrence matrix NN^T :

$$A = NN^T - rI .$$

Recall also that the normal equations for estimating the variety effects $\underline{\tau}$ (subject to the side condition $\underline{1}^T\underline{\tau} = 0$) are:

$$C\underline{\tau} = \underline{q} ,$$

where $$C = rI - \frac{1}{k}NN^T .$$

We will show, first, that a solution to these equations is:

$$\underline{\hat{\tau}} = (C + r\mathcal{J})^{-1}\underline{q} , \qquad (2.1)$$

where $\mathcal{J} = \frac{1}{v}J$. Then we will show that this solution can be expressed in terms of the adjacency matrix A.[†]

Showing that (2.1) is a solution involves showing:

(a) that the matrix $C + r\mathcal{J}$ is non-singular;

(b) that the vector $\underline{\hat{\tau}}$ defined by (2.1) satisfies the normal equations.

(a) In order to show that the matrix is non-singular, we show that all its eigenvalues are strictly positive. In fact, we show that its eigenvalues are r and re_i (for i = 1,2,....,v-1), where the e_i are the canonical efficiency factors of the design as defined on page 6.

Now, the eigenvalues of C are 0 and re_i $(1 \leq i \leq v-1)$. An

[†] The reason for taking this particular solution out of the many possible solutions of the normal equations is, principally, to simplify the algebra. That this solution also has some other desirable properties is established in Appendix 1. It is, furthermore, the one used by Pearce (1963) and Tocher (1952).

eigenvector associated with the eigenvalue 0 is $\underline{1}$. So there is a set of orthogonal eigenvectors:

$$\{\ \underline{1}\ ,\ \underline{x}_i\ :\ 1 \le i \le v\text{-}1\ \}\ ,$$

where re_i has eigenvector $\underline{x}_i$. In particular, therefore, $J\underline{x}_i = \underline{0}$ for each i. So for each i,

$$\begin{aligned}(C + rJ)\underline{x}_i &= C\underline{x}_i \\ &= re_i\underline{x}_i\ ;\end{aligned}$$

and also, $(C + rJ)\underline{1} = r\underline{1}$.

Hence the eigenvalues of $(C + rJ)$ are, indeed, r and re_i for $1 \le i \le v\text{-}1$, and so $(C + rJ)$ is non-singular.

(b) To show that the vector $\underline{\hat{t}}$ defined by equation (2.1) satisfies the normal equations, we must show that:

$$C\underline{\hat{t}} = \underline{q}\ ;$$

that is, that: $C(C + rJ)^{-1}\underline{q} = \underline{q}$.

Now, of course, $(C + rJ)(C + rJ)^{-1} = I.$

Hence, $C(C + rJ)^{-1} = I - rJ(C + rJ)^{-1}$.

Furthermore, $J(C + rJ) = rJ$

(since $JC = 0$ and $J^2 = J$).

Therefore, $J = rJ(C + rJ)^{-1}$.

It follows from these observations that:

$$C(C + rJ)^{-1} = I - J,$$

and so that $C(C + rJ)^{-1}\underline{q} = \underline{q} - J\underline{q}$.

But $J\underline{q} = \underline{0}$. (This stems from the definition of $\underline{q}$: see page 5.) The result follows.

For convenience, we will often denote $(C + rJ)^{-1}$ by C^{-}, it being a generalised inverse of C. (C^{-} is in Tocher's notation Ω .)

So the vector $\hat{\underline{\tau}}$ defined by equation (2.1) is a solution of the normal equations. We now demonstrate the role of the adjacency matrix A - first by showing how this solution can be expressed in terms of it, and then (in 2.3.2) by describing how, in consequence, it is relevant to the $\underline{\underline{A}}$-criterion.

The first of these points is straightforward. We have that:

$$C = rI - \frac{1}{k}NN^T ,$$

and we can use equation (1.2) to express NN^T in terms of A:

$$NN^T = A + rI .$$

This now enables us to express C in terms of A:

$$C = \frac{r(k-1)}{k} I - \frac{1}{k}A ,$$

and, as a result, to express the solution $\hat{\underline{\tau}}$ in terms of A:

$$\begin{aligned} \hat{\underline{\tau}} &= (C + rJ)^{-1}\underline{q} \\ &= (rI - \frac{1}{k}NN^T + rJ)^{-1}\underline{q} \\ &= \frac{1}{r}(I - \frac{1}{rk}NN^T + J)^{-1}\underline{q} \\ &= \frac{1}{r}(I - \frac{1}{rk}(A + rI) + J)^{-1}\underline{q} . \qquad (2.2) \end{aligned}$$

2.3.2 The variety concurrence graph and the $\underline{\underline{A}}$-criterion

The second point - the connection of the adjacency matrix A with the $\underline{\underline{A}}$-criterion - comes from this. We have that:

$$\underline{\underline{A}} = \frac{v-1}{\sum_{i=1}^{v-1} e_i^{-1}} .$$

Now, the eigenvalues of $(C + rJ)^{-1}$ are $\frac{1}{r}$ and $\frac{1}{r}e_i^{-1}$ for $1 \leq i \leq v-1$. (This follows from (a) on pages 56-57.) So:

$$\frac{1}{r} + \frac{1}{r}\sum_{i=1}^{v-1} e_i^{-1} = \mathrm{Tr}((C + rJ)^{-1}) ,$$

and so, $$\sum_{i=1}^{v-1} e_i^{-1} = r\mathrm{Tr}((C + rJ)^{-1}) - 1 ;$$

that is, $$\underline{\underline{A}} = \frac{v-1}{r\mathrm{Tr}(C^-) - 1} . \qquad (2.3)$$

So maximising $\underline{\underline{A}}$ is equivalent to minimising the quantity $r\mathrm{Tr}(C^-)$, and this equals $\mathrm{Tr}((I - \frac{1}{rk}(A + rI) + J)^{-1})$. From this we will derive the graphical criteria, which - as we have already indicated tentatively - are concerned with enumerating circuits. These criteria will emerge once we have expressed the quantity $\mathrm{Tr}((I - \frac{1}{rk}(A + rI) + J)^{-1})$ as an infinite sum of terms involving $\mathrm{Tr}(A^h)$ for $h \geq 1$. ($\mathrm{Tr}(A^h)$ is, of course, the number of circuits of length h in the graph.)

Expressing that quantity in this way involves two steps: first, we show that the matrix $(I - \frac{1}{rk}(A + rI) + J)^{-1}$ can be expressed in the form:

$$I + \sum_{n=1}^{\infty} (\frac{1}{rk}(A + rI) - J)^n ;$$

and, then, we expand the terms of this sum so that powers of the matrix A appear explicitly.

For the first of these steps, we must verify that the infinite sum converges, and this can be done by showing that the eigenvalues of the matrix $\frac{1}{rk}(A + rI) - J$ all lie in the interval $[0,1)$.. Now, we have already shown that the eigenvalues of $C + rJ$ are r and re_i for $1 \leq i \leq v-1$. Also,

$$\begin{aligned} \frac{1}{rk}(A + rI) - J &= I - \frac{1}{r}C - J \\ &= I - \frac{1}{r}(C + rJ) , \end{aligned}$$

and so the eigenvalues of $\frac{1}{rk}(A + rI) - J$ are 0 and $1 - e_i$ for

$1 \leq i \leq v-1$, all of which do indeed lie in $[0,1)$. This establishes that:

$$(I - \frac{1}{rk}(A + rI) + \mathfrak{J})^{-1} = I + \sum_{n=1}^{\infty} (\frac{1}{rk}(A + rI) - \mathfrak{J})^{n} \quad . \qquad (2.4)$$

So it is the trace of this infinite sum which is to be minimised if $\underline{A}$ is to be maximised. That this is connected with minimising the numbers of circuits of various lengths can be established by altering the form of the terms in the sum so that powers of A appear explicitly. That involves expanding the matrix $(\frac{1}{rk}(A + rI) - \mathfrak{J})^{n}$.

We show, first, that for $n \geq 1$ this equals:

$$\frac{1}{r^{n}k^{n}}(A + rI)^{n} - \mathfrak{J} \ ;$$

then we will expand $(A + rI)^{n}$. For the first of these, we use mathematical induction on n. The result is obviously true for n equal to 1. As the inductive step, suppose that it holds for some $n \geq 1$. Then the quantity $(\frac{1}{rk}(A + rI) - \mathfrak{J})^{n+1}$ must equal

$$(\frac{1}{r^{n}k^{n}}(A + rI)^{n} - \mathfrak{J})(\frac{1}{rk}(A + rI) - \mathfrak{J}) \ ,$$

which equals:

$$\frac{1}{r^{n+1}k^{n+1}}(A + rI)^{n+1} - \frac{1}{r^{n}k^{n}}(A + rI)^{n}\mathfrak{J} - \frac{1}{rk}\mathfrak{J}(A + rI) + \mathfrak{J}^{2} \ .$$

Now, $(A + rI)\mathfrak{J} = rk\mathfrak{J} = \mathfrak{J}(A + rI)$, and so $(A + rI)^{n}\mathfrak{J} = r^{n}k^{n}\mathfrak{J}$. So, putting all these together, we have that:

$$(\frac{1}{rk}(A + rI) - \mathfrak{J})^{n+1} = \frac{1}{r^{n+1}k^{n+1}}(A + rI)^{n+1} - \mathfrak{J} \ .$$

This completes the inductive step in the proof. So, indeed,

$$(\frac{1}{rk}(A + rI) - \mathfrak{J})^{n} = \frac{1}{r^{n}k^{n}}(A + rI)^{n} - \mathfrak{J} \ .$$

Finally, we expand $(A + rI)^n$. This is easy: it equals:

$$\sum_{h=0}^{n} \binom{n}{h} r^{n-h} A^h \quad .$$

Hence we have expanded the matrix $(\frac{1}{rk}(A + rI) - J)^n$ as we wanted - that is, so that powers of the adjacency matrix A appear explicitly:

$$(\frac{1}{rk}(A + rI) - J)^n = \frac{1}{r^n k^n} \sum_{h=0}^{n} \binom{n}{h} r^{n-h} A^h - J \ .$$

And this in turn equals :

$$\frac{1}{k^n} \sum_{h=0}^{n} (\binom{n}{h}\frac{1}{r^h} A^h - \frac{k^n}{n+1} J) \ .$$

So, substituting back in (2.4), we get:

$$(I - \frac{1}{rk}(A + rI) + J)^{-1} = I + \sum_{n=1}^{\infty} \sum_{h=0}^{n} \frac{1}{k^n}(\binom{n}{h}\frac{1}{r^h}A^h - \frac{k^n}{n+1}J) \ .$$

Therefore:

$$\mathrm{Tr}(rC^-) = v + \sum_{n=1}^{\infty} \sum_{h=0}^{n} \frac{1}{k^n}(\binom{n}{h}\frac{1}{r^h}\mathrm{Tr}(A^h) - \frac{k^n}{n+1}) \ . \qquad (2.5)$$

Hence arise the graphical criteria. Minimising $\mathrm{Tr}(rC^-)$ is the same as minimising the expression on the right-hand side of (2.5), The only variable components of this expression are the quantities $\mathrm{Tr}(A^h)$. Now, $\mathrm{Tr}(A^h)$ is the number of circuits of length h in the variety concurrence graph. So, in a general sense, minimising $\mathrm{Tr}(rC^-)$ is connected with minimising the number of circuits (and so, as we have shown, with maximising the number of paths between distinct varieties) in this graph. It is on this, so far somewhat vague, notion that we base the graphical criteria. Briefly, these are what we have already heuristically outlined when we discussed paths and variety differences earlier: that is, criterion $\underline{\underline{C}}_h$ is defined to be the number of circuits of length h in the graph; that is

$$\underline{\underline{C}}_h = \mathrm{Tr}(A^h) \ .$$

And the aim is to minimise the number of these circuits; that is, to minimise $\underline{\underline{C}}_h$. Further, these criteria will be used sequentially: that is, in comparing two designs, criterion $\underline{\underline{C}}_h$ will be invoked only if the designs are indistinguishable on the basis of the criteria $\underline{\underline{C}}_1$, $\underline{\underline{C}}_2$,....., $\underline{\underline{C}}_{h-1}$.

2.4 Further mathematical justification of the graphical criteria, and comparisons with existing criteria

Thus far, therefore, we have defined the graphical criteria, and indicated how we intend to use them. In doing so, we have made clear why we consider them to be intuitively appealing by expressing them in terms of maximising the numbers of paths in the variety concurrence graphs. But there remain gaps in the theoretical justification of their application. Clearly, equation (2.5) does provide some indication on its own of their relevance to the $\underline{\underline{A}}$-criterion. It will not, in general, be possible to say that they are equivalent to the $\underline{\underline{A}}$-criterion, any more than is the $\underline{\underline{S}}$-criterion - or, for that matter, the $\underline{\underline{D}}$-criterion or the $\underline{\underline{E}}$-criterion. However, it is possible to establish a number of results which suggest strongly that these graphical criteria are, mathematically, preferable to the $\underline{\underline{S}}$-criterion and almost as good as the $\underline{\underline{A}}$-criterion, and that they are, intuitively, more attractive than either. These results can be summarised as follows.

2.4.1 We will provide a justification of the sequential approach by establishing two results: first (in (a)) we will derive a measure of the relative significance of the successive graphical criteria, and then (in (b)) we will use this to show that what variations that there can be in values of **A** must be due to circuits of shorter lengths.

2.4.2 We will show that if two designs are indistinguishable on all the graphical criteria, then they are indistinguishable on the **A**-criterion.

2.4.3 We will show that the graphical criteria can be interpreted as a generalisation of the **S**-criterion.

2.4.4 As a corollary of 2.4.3, we will suggest that the graphical criteria are more satisfactory intuitively than the **S**-criterion as it was formulated by Shah (1960) and as it has been interpreted ever since.

Moreover, three further points in favour of the graphical criteria will arise later in the thesis: we will show in the next two Chapters that (in contrast to the **A**-, **D**-, and **E**-criteria) they can be reduced to relatively straightforward algorithms in the cases of certain standard series of designs; we will follow up the result in 2.4.1(b) by providing evidence that, in practice, the first few graphical criteria (for h = 2, 3, and 4) perform almost as well as the $\underline{\underline{A}}$-criterion; and we will show in Chapter 6 that the graphical criteria are linked to the precision with which the design estimates variety differences in a more detailed way than are the criteria currently in use.

2.4.1 Justification of the sequential use of the graphical criteria

(a) First, then, how sound mathematically is our decision to use the graphical criteria sequentially? This practice implicitly assumes that the successive criteria are of decreasing significance. Is this justified, or could circuits of length h turn out to be less important than circuits of length n greater than h? In fact, we can answer these questions precisely: we show here in (a) that the criteria are, indeed, of decreasing significance, in a way that can be quantified exactly.

The significance of $\underline{\underline{C}}_h$ which equals $\mathrm{Tr}(A^h)$ depends on its contribution to the sum in the right-hand side of equation (2.5): that is, its contribution to the quantity:

$$\sum_{n=1}^{\infty} \sum_{h=0}^{n} \frac{1}{k^n} \left(\binom{n}{h} \frac{1}{r^h} \underline{\underline{C}}_h - \frac{k^n}{n+1} \right) .$$

It seems natural to measure that contribution by the coefficient that multiplies it in this sum, for that coefficient is the weight which is given in the sum to each circuit of length h. We will call the coefficient ω_h.

Fix some positive integer h. Then for $n \geq h$, $\underline{\underline{C}}_h$ appears once in:

$$\sum_{u=0}^{n} \frac{1}{k^n} \left(\binom{n}{u} \frac{1}{r^u} \underline{\underline{C}}_u - \frac{k^n}{n+1} \right) ,$$

and it is multiplied there by $\frac{1}{k^n} \frac{1}{r^h} \binom{n}{h}$. So, summing over all $n \geq h$, the weight ω_h given to $\underset{=}{C}_h$ is, in total, $\frac{1}{r^h} \sum_{n=h}^{\infty} \frac{1}{k^n}\binom{n}{h}$.

Now, so far, this is a purely formal sum: we do not yet know if it converges, far less what its value might be. In fact, however, we can show: (i) that this series converges;

and (ii) that its sum equals $\frac{k}{r^h(k-1)^{h+1}}$.

(i) To show that the series converges we use the limit form of the Ratio Test. (See, for example, White (1968).)

If $a_n = \frac{1}{k^n}\binom{n}{h}$, then

$$\begin{aligned}\frac{a_{n+1}}{a_n} &= \frac{\frac{1}{k^{n+1}}\binom{n+1}{h}}{\frac{1}{k^n}\binom{n}{h}} \\ &= \frac{1}{k}\,\frac{(n+1)!}{(n+1-h)!}\;\frac{(n-h)!}{n!} \\ &= \frac{1}{k}\,\frac{n+1}{n+1-h} \\ &= \frac{1}{k}\,\frac{1}{1-\frac{h}{n+1}} \\ &\to \frac{1}{k} \quad \text{as} \quad n \to \infty \\ &< 1 \text{ since } k \geq 2.\end{aligned}$$

So the series does converge.

(ii) Next, we find an explicit expression for the sum of the series. Define, for each non-negative integer h,

$$S_h = \sum_{n=h}^{\infty} \frac{1}{k^n}\binom{n}{h} .$$

We show by mathematical induction on h that $S_h = \frac{k}{(k-1)^{h+1}}$.

Now, $S_0 = \sum_{n=0}^{\infty} \frac{1}{k^n}\binom{n}{0} = \sum_{n=0}^{\infty} \frac{1}{k^n} = (1 - \frac{1}{k})^{-1} = \frac{k}{k-1}$ and so the result holds for h = 0. Suppose that it holds for some non-negative integer h. Then

$$S_{h+1} = \sum_{n=h+1}^{\infty} \frac{1}{k^n}\binom{n}{h+1}$$

$$= \frac{1}{(h+1)!} \sum_{n=h+1}^{\infty} \frac{n(n-1)\ldots.(n-h)}{k^n} \quad . \qquad (2.6)$$

Hence

$$kS_{h+1} = \frac{1}{(h+1)!} \sum_{n=h+1}^{\infty} \frac{n(n-1)\ldots..(n-h)}{k^{n-1}}$$

$$= \frac{1}{(h+1)!} \sum_{n=h}^{\infty} \frac{(n+1)n(n-1)\ldots..(n+1-h)}{k^n} \quad . \qquad (2.7)$$

Notice that the sum (2.6) can be started at h, since when n = h, n - h is 0. So, by (2.6) and (2.7), and since convergent series of positive terms can be re-arranged in any way we like,

$$(k-1)S_{h+1} = \frac{1}{(h+1)!} \sum_{n=h}^{\infty} \frac{n(n-1)\ldots..(n+1-h)(n+1-n+h)}{k^n}$$

$$= \sum_{n=h}^{\infty} \frac{n(n-1)\ldots..(n+1-h)}{k^n} \; \frac{h+1}{(h+1)!}$$

$$= \sum_{n=h}^{\infty} \frac{1}{k^n}\binom{n}{h}$$

$$= S_h \; .$$

But, by the induction hypothesis, $S_h = \dfrac{k}{(k-1)^{h+1}}$.

So $S_{h+1} = \dfrac{k}{(k-1)^{h+2}}$, and hence the result holds for all $h \geq 0$.

This shows, therefore, that the weight given to $\underline{\underline{C}}_h$ in the sum (2.5) is:

$$\omega_h = \frac{k}{r^h(k-1)^{h+1}} \quad , \qquad (2.8)$$

and this in turn establishes what we wanted: namely that the successive graphical criteria $\underline{\underline{C}}_h$ are of steadily decreasing significance. To be precise, $\underline{\underline{C}}_h$ is $r(k-1)$ times more significant than $\underline{\underline{C}}_{h+1}$. Or, put differently, one circuit of length h counts for the same as $r(k-1)$ circuits of length h+1 (since $\omega_h = r(k-1)\omega_{h+1}$). (It is interesting to note in passing that $r(k-1)$ is the valency of each point in the variety concurrence graph.)

This does not mean, of course, that the infinite expansion (2.5) of $\mathrm{Tr}(rC^-)$ can necessarily be re-arranged in such a way that its principal (or outermost) index is h: usually, indeed, this will not be possible (since the terms of the sum are not all positive). What it does provide, rather, is a measure of the relative significance of circuits of different lengths in this expansion - and that is all that matters for the practice of invoking the graphical criteria sequentially. It is worth noticing, also, in particular, that it would be slightly misleading - if superficially plausible - to measure the contribution to $\mathrm{Tr}(rC^-)$ of circuits of length h by the the ratio $\dfrac{\omega_h \underline{\underline{C}}_h}{\mathrm{Tr}(rC^-)}$. To do so would be to ignore the contribution of the constant terms in the expansion (2.5) - that is, the terms $\dfrac{k^n}{n+1}$ - and so would tend to give an inaccurately low measure of the significance of the circuits.

So this result should be seen only as providing support in a general way for our decision to use the graphical criteria sequentially. Likewise with the following corollary, which shows that for any fixed non-negative integer h, the combined weights given to all circuits with length strictly greater than h can never exceed the weight given to circuits of length h.

These combined weights are:

$$\sum_{m=h+1}^{\infty} \omega_m = \sum_{m=h+1}^{\infty} \frac{k}{r^m(k-1)^{m+1}}$$

$$= \frac{\dfrac{k}{r^{h+1}(k-1)^{h+2}}}{1 - \dfrac{1}{r(k-1)}}$$

$$= \frac{k}{r^h(k-1)^{h+1}} \left(\frac{1}{r(k-1) - 1}\right)$$

$$= \frac{\omega_h}{r(k-1) - 1}$$

So, for each non-negative integer h, the significance of the criterion $\underline{\underline{C}}_h$ is (r(k-1)-1) times greater than the combined significance of all the criteria $\underline{\underline{C}}_{h+1}$, $\underline{\underline{C}}_{h+2}$,.......

(b) But of far greater importance than the significance of circuits in one design on its own is their significance in accounting for differences between designs: the whole point of having criteria of efficiency is to compare different designs. What we show now is that differences between designs are due mainly to circuits of shorter lengths. To be precise, we show (with a fairly minor qualification when k = 2) that $\omega_h \underline{\underline{C}}_h$ - which is the contribution of circuits of length h to the sum (2.5) - converges to a constant quantity as h increases. Hence, in comparing two designs, differences in numbers of circuits of length h become negligible as h gets large; or, in other words, most of the difference between harmonic mean efficiency factors will be accounted for by differences in numbers of circuits of shorter lengths.

We show, first, that for $k \geq 3$,

$$\omega_h \underline{\underline{C}}_h \rightarrow \frac{k}{k-1} \text{ as } h \rightarrow \infty \quad .$$

We return to the slightly more complicated case of $k = 2$ afterwards.

The quantity $\underline{\underline{C}}_h$, which equals $\mathrm{Tr}(A^h)$, equals the sum of the eigenvalues of the matrix A^h. Arguing exactly as in 2.3.1(a), we can show that the eigenvalues of A are $r(k-1)$ and $r(k-1) - rke_i$ for $1 \leq i \leq v-1$. So the eigenvalues of A^h are $r^h(k-1)^h$ and $(r(k-1) - rke_i)^h$ $(1 \leq i \leq v-1)$. Hence:

$$\underline{\underline{C}}_h = r^h(k-1)^h + \sum_{i=1}^{v-1} (r(k-1) - rke_i)^h \quad .$$

It follows that:

$$\omega_h \underline{\underline{C}}_h = \frac{k}{k-1}(1 + \sum_{i=1}^{v-1} (1 - \frac{k}{k-1} e_i)^h) \, ,$$

and this does indeed converge to $\frac{k}{k-1}$ as $h \rightarrow \infty$, since the canonical efficiency factors all lie in the interval $(0,1]$.

Now, this proof depends on the stipulation that k is not equal to 2: specifically, that condition ensures that even for a canonical efficiency factor e_i which equals 1, the quantity $(1 - \frac{k}{k-1} e_i)^h$ - which then is $(\frac{-1}{k-1})^h$ - converges to zero. When k is equal to 2, matters become rather more awkward, though not so much so as to vitiate the general principle that it is circuits of shorter lengths which are important in determining differences in harmonic mean efficiency factors.

Clearly, on the one hand, in this case, if there are no canonical efficiency factors which equal 1, then $\omega_h \underline{\underline{C}}_h$ will still converge to $\frac{k}{k-1}$ (here equal to 2). Moreover, on the other hand, there cannot, if $k = 2$, be more than one canonical efficiency factor which equals 1: if there were, then there would be more than one term in $\sum_{i=1}^{v-1} (r(k-1) - rke_i)^h$ which equalled $(-r)^h$, and so for large odd values of h, $\omega_h \underline{\underline{C}}_h$ would be negative: an obvious contradiction since $\underline{\underline{C}}_h$ is the number of circuits of length h in the graph.

But that does leave the possibility that there is precisely one canonical efficiency factor which equals 1, and in such cases, for large values of h, $\omega_h \underline{\underline{C}}_h$ is close to, alternately, $\frac{2k}{k-1} = 4$ (h even) and O (h odd). A simple example is this design for 4 varieties in 4 blocks of size 2:

$$\begin{array}{cc} 0 & 1 \\ 2 & 3 \\ 0 & 2 \\ 1 & 3 \end{array} .$$

This does not, of course, cause any difficulty if we are comparing two designs which each have no canonical efficiency factor equal to 1, or two which each have one canonical efficiency factor equal to 1; where problems do arise is in comparing a design of the first of these kinds with a design of the second. However, even here, things are not really as serious as they might seem. For we can, instead, shift attention to $\omega_{2g} \underline{\underline{C}}_{2g} + \omega_{2g+1} \underline{\underline{C}}_{2g+1}$: it is clear from what we have already said that this quantity will always converge to the constant quantity $\frac{2k}{k-1} = 4$, and so that, in this slightly modified sense, it is still true to say that circuits of larger lengths are negligible in accounting for differences between designs.

(A further point concerning the sequential approach is contained in 2.5.)

2.4.2 The graphical criteria are collectively a stronger condition than the A-criterion

Secondly, we show that the sequence of graphical criteria is, if it could be continued indefinitely, stronger than the $\underline{\underline{A}}$-criterion. That is, if two designs D_0 and D_1 are indistinguishable on all the graphical criteria, then they are indistinguishable on the $\underline{\underline{A}}$-criterion.

As a matter of fact, this result is almost obvious: if D_0 and D_1 have respective variety adjacency matrices A_0 and A_1, then their equivalence on all the graphical criteria means that $Tr(A_0^h) = Tr(A_1^h)$ for all $h \geq 0$, and so, by equation (2.5), $Tr(rC_0^-) = Tr(rC_1^-)$ (where C_0 and C_1 are the respective C matrices associated with the normal equations).

So by equation (2.4) D_0 and D_1 have equal values of $\underline{\underline{A}}$; in other words, they are indistinguishable on the $\underline{\underline{A}}$-criterion.

2.4.3 The graphical criteria are a generalisation of the S-criterion

The third point in favour of the graphical criteria is that they are a generalisation of the $\underline{\underline{S}}$-criterion. We show this by showing that $\underline{\underline{C}}_2$ is equivalent to the $\underline{\underline{S}}$-criterion, in the sense that two designs D_0 and D_1 are indistinguishable on the $\underline{\underline{C}}_2$ criterion if, and only if, they are indistinguishable on the $\underline{\underline{S}}$-criterion.

We will use suffix 0 to denote matrices associated with design D_0 and the suffix 1 to denote matrices associated with the design D_1.

Now, by equation (1.3), $A_u = N_u N_u^T - rI$ for each u, and so

$$\begin{aligned}
\mathrm{Tr}(A_u^2) &= \sum_{i=1}^{v}\sum_{j=1}^{v} ((A_u)_{ij})^2 \qquad \text{(since } A_u \text{ is symmetric)} \\
&= \sum_{\substack{i,j=1 \\ i\neq j}}^{v} ((A_u)_{ij})^2 + \sum_{i=1}^{v} ((A_u)_{ii})^2 \\
&= \sum_{\substack{i,j=1 \\ i\neq j}}^{v} ((N_uN_u^T)_{ij})^2 \\
&= \sum_{i,j=1}^{v} ((N_uN_u^T)_{ij})^2 - \sum_{i=1}^{v} ((N_uN_u^T)_{ii})^2 \\
&= \underline{\underline{S}}_u - r^2v
\end{aligned}$$

(where $\underline{\underline{S}}_u$ is the $\underline{\underline{S}}$-measure for the design D_u).

So $\mathrm{Tr}(A_0^2) = \mathrm{Tr}(A_1^2)$ if, and only if, $\underline{\underline{S}}_0 = \underline{\underline{S}}_1$.

That is, D_0 and D_1 are equivalent on the $\underline{\underline{C}}_2$-criterion if, and only if, they are equivalent on the $\underline{\underline{S}}$-criterion.

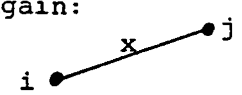
The import of this equivalence can be illustrated as follows, by a comment from Patterson and Williams (1976b). After discussing the use of the $\underline{\underline{S}}$-criterion in selecting efficient α designs, these authors come to the following conclusion (page 87):

> "Thus, given the choice, we prefer an α(0,1) design to an α(0,1,2) design. If we have to use an α(0,1,2) design, we prefer one with as few pairs of varieties as possible concurring twice."

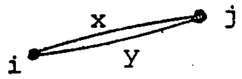
(An α(0,1) design is one in which the off-diagonal entries of the variety concurrence matrix NN^T are all 0 or 1; an α(0,1,2) design is one where these entries are all 0, 1, or 2.)

We explain now how the $\underline{\underline{C}}_2$-criterion would lead us to exactly the same conclusions.

As we saw earlier, the $\underline{\underline{C}}_2$-criterion amounts to minimising the number of circuits of length 2 in the variety concurrence graph. Circuits of length 2 are of two types. On the one hand there are those which traverse the same line twice: that is, if point i is adjacent to point j by the line x, then a circuit of length two can be formed by starting at i, traversing x to j, and then re-traversing x to i again:



On the other hand there are proper circuits of length two: circuits where the lines traversed are different. That is, if i and j are joined by the distinct lines x and y, then a circuit of length two can be formed by traversing x from i to j, and then traversing y from j to i:



Circuits of the first type cannot be avoided, and, moreover, the number of them is precisely twice the number of lines in the graph. (Recall from page 14 that a circuit is counted once for each of its points, here two.) Since there are $\frac{1}{2}vr(k-1)$ lines in the variety concurrence graph, the number of circuits of this type is $vr(k-1)$.

However, the number of circuits of the second type is not simply a function of v, k, and r, and it is this number, therefore, which the $\underline{\underline{C}}_2$-criterion is concerned with reducing. For any two points i and j, the number of lines joining them is a_{ij}, and so, if a_{ij} is greater than one, then the number of circuits of length two made up of pairs of these lines is $2\times2\times\frac{1}{2}a_{ij}(a_{ij} - 1)$. (The two factors of 2 come in because each choice of two lines from the set of a_{ij} lines is counted four times: once for each of the points, and, for each of them, once for each of the two orientations.) So we would want to reduce as far as is possible the number of multiple lines (that is, pairs of points i and j which have a_{ij} greater than one). In particular, this means that the graphical criteria $\underline{\underline{C}}_2$ leads to the same conclusions as those contained in the quotation from Patterson and Williams on page 71.

(The application of the graphical criteria to α designs is explored in more detail in Chapters 4 and 5.)

Before we leave the $\underline{\underline{C}}_2$-criterion here, we establish a result which is of some general interest, and which will be useful later: namely, that Regular Graph Designs have a minimal number of circuits of length two.

Consider any design in the class for v varieties in blocks of size k, and with each variety replicated r times. Then the number of circuits of length two is $\mathrm{Tr}(A^2)$, which equals $\sum_{i,j=1}^{v} a_{ij}^2$, which is $\sum_{i\neq j} a_{ij}^2$ since the diagonal entries of A are all zero. The defining feature of a Regular Graph Design is that the off-diagonal entries of A differ by at most one. We show that this implies that a Regular Graph Design has a minimal value of $\sum_{i\neq j} a_{ij}^2$.

To this end, we invoke a general result from elementary number theory: the partition of any integer z into u integer parts which differ by at most one has smaller sum of squares than any other partition of z into u integer parts. (For a proof, see Williams, Patterson and John (1976), page 298.) Now, for any design, the sum $\sum_{i\neq j} a_{ij}$ is $vr(k-1)$, since each of the v points in the variety concurrence graph has valency $r(k-1)$. That is, the integers a_{ij} $(1\leq i,j\leq v,\ i\neq j)$ form a partition of the integer vr(k-1). Translating the general number theory result into our terminology, we have that the integer partition of vr(k-1) into v(v-1) parts which has the smallest sum of squares is the one whose parts differ by at most one. Since in a Regular Graph Design the off-diagonal entries of the adjacency matrix A form a partition of the integer vr(k-1) into v(v-1) integer parts differing by at most one, it follows that a Regular Graph Design has, indeed, a minimal value of the sum $\sum_{i \neq j} a_{ij}^2$. That is, a Regular Graph Design has a minimal number of circuits of length two, and so is optimal on the $\underline{\underline{C}}_2$-criterion.

It should be emphasised that this does not imply that the range of the off-diagonal entries of the variety adjacency matrix is a reliable guide to the number of circuits of length two. We mention this since such a practice might seem at first sight to be attractive - precisely because of what we have just shown: a Regular Graph Design has both a minimal number of circuits of length two and also a minimal value of the range (namely, 1). Regular Graph Designs are rather special in this respect: in general, the possession of a small number of circuits of length two - or of a high value of the harmonic mean efficiency factor - need not be associated with the possession of a minimal value of the range. In illustration of this, we cite two designs for 90 varieties in 60 blocks of size 9 (found by Gilchrist (1975), page 29). The details are immaterial here: what is important is that the one with the smaller number of circuits of length two (5024 as opposed to 5120) has the larger value of the range (3 as opposed to 2).

2.4.4 The graphical criteria are intuitively more satisfactory than the **S**-criterion

The fourth point - and the final one in this Chapter - which we make in favour of the graphical criteria concerns their resting on a more natural, more intuitively appealing, foundation than that which underlies the **S**-criterion.

As developed by Eccleston and Hedayat (1974) from Shah (1960), the **S**-criterion is a special case of the (**M**,**S**)-criterion, which, as we noted on page 37, entails first selecting all designs which maximise Tr(C), and then choosing amongst them the ones which minimise $\mathrm{Tr}(C^2)$. We might, therefore, be inclined to speculate how these two criteria - **M** and **S** - could be generalised in their own terms: that is, by considering $\mathrm{Tr}(C^h)$ for $h \geq 3$. Immediately, however, an obvious question arises: are we to minimise or maximise $\mathrm{Tr}(C^h)$?

We answer this question by examining in more detail how the **M** and **S** criteria have their roots in the **A**-criterion. We have:

$$\mathbf{A} = \frac{v-1}{\sum_{i=1}^{v-1} e_i^{-1}} .$$

So, since $0 \leq 1-e_i < 1$, $\mathbf{A}^{-1} = \sum_{n=0}^{\infty} M_n$,

where:

$$M_n = \frac{1}{v-1} \sum_{i=1}^{v-1} (1 - e_i)^n .$$

Then:

$$M_0 = \frac{1}{v-1} \sum_{i=1}^{v-1} 1 = 1;$$

$$\bar{M}_1 = \frac{1}{v-1}\sum_{i=1}^{v-1}(1-e_i) = 1-\bar{e} \;;$$

$$M_2 = \frac{1}{v-1}\sum_{i=1}^{v-1}(1-e_i)^2 = 1-2\bar{e} + \frac{1}{v-1}\sum_{i=1}^{v-1} e_i^2$$

$$= 2M_1 - 1 + \frac{1}{v-1}\sum_{i=1}^{v-1} e_i^2 \,.$$

Arguing in an analogous fashion to the one we adopted in devising the graphical criteria, the maximising of $\underline{\underline{A}}$ is seen to be connected with the minimising of all the M_n. Minimising M_1 is equivalent to maximising $\bar{e}$, and that is the $\underline{\underline{M}}$-criterion. For a fixed value of M_1, minimising M_2 is equivalent to minimising $\mathrm{Tr}(C^2) = r^2 \sum_{i=1}^{v-1} e_i^2$, and that is the $\underline{\underline{S}}$-criterion. Continuing, we have:

$$M_3 = 1 - 3\bar{e} + \frac{3}{v-1}\sum_{i=1}^{v-1} e_i^2 - \frac{1}{v-1}\sum_{i=1}^{v-1} e_i^3 \,,$$

which equals: $\quad 1 + 3M_2 - 3M_1 - \frac{1}{v-1}\sum_{i=1}^{v-1} e_i^3 \,,$

and so, for fixed values of M_1 and M_2, minimising M_3 is equivalent to maximising $\mathrm{Tr}(C^3) = r^3 \sum_{i=1}^{v-1} e_i^3$.

So the $\underline{\underline{M}}$ and $\underline{\underline{S}}$ criteria, when generalised in their own terms, could lead to a sequence of criteria that involve alternately minimising and maximising expressions in the matrix C. We would suggest that this has less intuitive appeal than do our graphical criteria, which involve consistently a process of minimising - and minimising, moreover (as we described in 2.2), quantities which have some direct relevance to the statistical purpose of the experiment:

to, that is, the estimation of differences between variety effects.

2.5 Fundamental conjecture concerning the application of the graphical criteria

Of course, the whole point of the graphical criteria is to use graph theory to help in the search for efficient designs. As a preliminary step to that end, we introduce here a conjecture which we will invoke frequently later in practical applications.

This conjecture arises as a natural generalisation of the already widely accepted conjecture concerning the $\underline{\underline{S}}$-criterion which we alluded to in Chapter 1 (page 9): namely that any design which is best on the $\underline{\underline{A}}$-criterion will also be best on the $\underline{\underline{S}}$-criterion. Since the $\underline{\underline{S}}$-criterion is equivalent to the $\underline{\underline{C}}_2$-criterion, this conjecture is equivalent to that which would suggest that an $\underline{\underline{A}}$-optimal design is always $\underline{\underline{C}}_2$-optimal. And from that a generalisation is obvious. Suppose that the design D is $\underline{\underline{A}}$-optimal, and that D' is any other design that is not $\underline{\underline{A}}$-optimal. We conjecture that if h_0 is the first value of h for which the numbers of circuits of length h in the variety concurrence graphs of D and D' differ, then $\underline{\underline{C}}_{h_0}(D) < \underline{\underline{C}}_{h_0}(D')$ (denoting the number of circuits of length h in D by $\underline{\underline{C}}_h(D)$). (That such an h_0 exists follows from 2.4.2.)

If this conjecture is true, then it would be a partial converse to the result in 2.4.2. We have not been able to prove it; but neither have we found a counterexample. Indeed, considering the theory we have presented (especially in 2.4.1(b)), we would suggest that the acceptance of this conjecture is as reasonable as is the widespread acceptance of the conjecture concerning the $\mathbf{S}$-criterion which we mentioned in the last paragraph.

The use to which we propose to put our conjecture is analogous to the use which is made of that existing conjecture. That is, we will use our conjecture as a means of facilitating the search for $\mathbf{A}$-optimal designs by restricting the search to designs which are optimal on certain of the graphical criteria; and we will be able to find these designs easily by exploiting simplifications made possible by the structure of the variety concurrence graphs. Thus, just as Mitchell and John (1977) searched for designs that are $\mathbf{A}$-optimal among Regular Graph Designs (that is, among $\mathbf{C}_2$-optimal designs), so, likewise, will we search for designs that are $\mathbf{A}$-optimal among those which are best on, for example, the $\mathbf{C}_3$-criterion as well as the $\mathbf{C}_2$-criterion; and we will calculate $\mathbf{C}_3$ - that is, the number of circuits of length three - in a straightforward fashion from the graph. If our conjecture is correct, then such searching will find the designs which are $\mathbf{A}$-optimal.

Frequently, also, we will narrow the conjecture to refer only to specific series of designs: thus, for example, we will conjecture in Chapters 4 and 5 that the best α design on the $\mathbf{A}$-criterion is best also on the $\mathbf{C}_2$-, $\mathbf{C}_3$-, and $\mathbf{C}_4$-criteria.

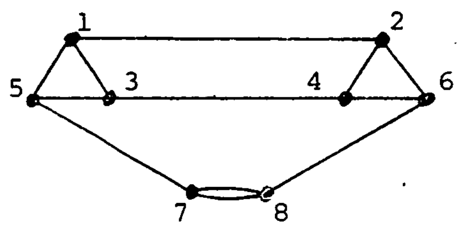
Before we leave this, there are two further comments worth making here. First, it does seem that establishing the truth (or falsity) of this conjecture would probably be very difficult. It would be at least as difficult as establishing the truth of the conjecture concerning the $\underline{\underline{S}}$-criterion - and that is a problem which has remained unresolved for some time. In a sense, of course, this does not really matter; what does matter to the statistician is that such conjectures should work - should, that is, help to construct designs that do what they are supposed to do reasonably well. It seems likely - on the strength of the theory, and on the strength of the examples we have looked at - that this would indeed be the case with our conjecture; but full confirmation of that must await wider testing than we have had time to carry out for this thesis.

The second comment relevant to our conjecture concerns the possibility of generalising it. The crudest generalisation would be to remove the condition that the design D be $\underline{\underline{A}}$-optimal; that is, to alter the conjecture so that it asserted that (in an obvious notation) $\underline{\underline{A}}(D) > \underline{\underline{A}}(D')$ if, and only if, $\underline{\underline{C}}_{h_0}(D) < \underline{\underline{C}}_{h_0}(D')$. However, this general claim turns out to be false. A counterexample can be constructed for 8 varieties in 12 blocks of size 2, the designs being as follows:

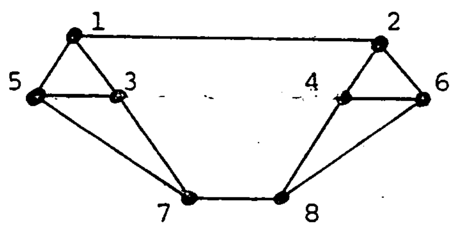
Design D	Design D'
1 2	1 2
1 3	1 3
1 5	1 5
2 4	2 4
2 6	2 6
3 4	3 5
3 5	3 7
4 6	4 6
5 7	4 8
6 8	5 7
7 8	6 8
7 8	7 8

These have the variety concurrence graphs:

$\underline{G_v(D)}$



$\underline{G_v(D')}$



Then $\underline{\underline{A}}(D) = .4011$ and $\underline{\underline{A}}(D') = .3889$. So $\underline{\underline{A}}(D) > \underline{\underline{A}}(D')$, while it is obvious from the graph that $\underline{\underline{C}}_2(D) > \underline{\underline{C}}_2(D')$.

But a more sensible generalisation would be to allow the design D to be in some sense "nearly" $\underline{\underline{A}}$-optimal. The claim would then be that if D were "nearly" $\underline{\underline{A}}$-optimal, and if D' were any other design, then $\underline{\underline{A}}(D) > \underline{\underline{A}}(D')$ if, and only if, $\underline{\underline{C}}_{h_0}(D) < \underline{\underline{C}}_{h_0}(D')$. It seems probable that this restricted generalisation - with some suitably precise definition of "nearly" - is likely to be true. (In the above case, there exist designs for 8 varieties in 12 blocks of size 2 which have harmonic mean efficiency factors of around .5, and so the design D there is far enough away from optimality not to contradict this restricted generalisation.)

2.6 The graphical criteria and resolvable two-replicate designs

Finally in this Chapter, we add two further results, concerning the application of the graphical criteria to resolvable two-replicate designs. First (in 2.6.1) - as an extension of 2.4.1 - we enhance the justification of the sequential approach by showing that, for such designs, the only graphical criteria which are of interest are those for circuits of even length. Then (in 2.6.2) we show that the $\mathbf{C}_4$-criterion for these designs is equivalent to the $\mathbf{C}_2$-criterion for their contractions - thus lending extra credibility to the suggestion in 2.4.3 that the graphical criteria can be seen as a natural extension of the $\mathbf{S}$-criterion.

2.6.1 Redundancy of all the graphical criteria for circuits of odd length

First, then, we show the following. Let D_0 and D_1 both be resolvable two-replicate designs for v varieties in blocks of size k, with respective variety adjacency matrices A_0 and A_1. Then if they are indistinguishable on each of the graphical criteria $\mathbf{C}_2$, $\mathbf{C}_3$,... ..., $\mathbf{C}_{2m}$, they are indistinguishable on the graphical criterion $\mathbf{C}_{2m+1}$. That is, if $\mathrm{Tr}(A_0^i) = \mathrm{Tr}(A_1^i)$ for each i = 2,3,...,2m, then $\mathrm{Tr}(A_0^{2m+1}) = \mathrm{Tr}(A_1^{2m+1})$. In other words, $\mathbf{C}_{2m+1}$ tells us no more about the efficiencies of the designs than do $\mathbf{C}_2$, $\mathbf{C}_3$,...., $\mathbf{C}_{2m}$.

We will establish this by means of a special property of the block concurrence graph of a resolvable two-replicate design - namely, that it has no circuits at all of odd length - and also by means of certain links which we will show to exist between the adjacency matrices of the block concurrence graph and the variety concurrence graph.

Recall from Chapter 1 that if a design is resolvable, then its block concurrence graph is multipartite, with sides corresponding to replicates. In particular, therefore, if the design has two replicates as well as being resolvable, then this graph is bipartite. It therefore has no circuits of odd length, since the successive points in any circuit must come alternately from each side.

In consequence, for such designs, if we denote the adjacency matrix of the block concurrence graph by B, then

$$\mathrm{Tr}(B^{2m+1}) = 0 \text{ for each } m \geq 0 \qquad (2.9)$$

(since $\mathrm{Tr}(B^{2m+1})$ is the number of circuits of length 2m+1 in the graph).

The next step - preliminary to arriving at what we want - is to express the number of circuits of length n in the variety concurrence graph in terms of the number of circuits of length n in the block concurrence graph and the number of circuits of length less than n in the variety concurrence graph. This we do by means of the following formula:

$$\mathrm{Tr}(A^n) = \mathrm{Tr}(B^n) - \sum_{h=0}^{n} \sum_{\substack{x=0 \\ x\neq n}}^{h} (-k)^{n-h} r^{h-x} \binom{n}{h} \binom{h}{x} \mathrm{Tr}(A^x) . \qquad (2.10)$$

The proof of this depends essentially on the fact that both A and B can be expressed in terms of the design incidence matrix N. Thus, as we have seen in Chapter 1, $B = N^TN - kI_b$ and $A = NN^T - rI_v$. So, on the one hand,

$$B^n = \sum_{h=0}^{n} (-k)^{n-h} \binom{n}{h} (N^TN)^h ,$$

and therefore $\mathrm{Tr}(B^n) = \sum_{h=0}^{n} (-k)^{n-h} \binom{n}{h} \mathrm{Tr}((N^TN)^h)$.

On the other hand, $NN^T = A + rI_v$, which implies that

$$(NN^T)^h = \sum_{x=0}^{h} r^{h-x} \binom{h}{x} A^x .$$

Moreover, it also follows from the properties of the trace function that $\mathrm{Tr}((NN^T)^h) = \mathrm{Tr}((N^TN)^h)$. Hence:

$$\mathrm{Tr}(B^n) = \sum_{h=0}^{n} \sum_{x=0}^{h} (-k)^{n-h} r^{h-x} \binom{n}{h} \binom{h}{x} \mathrm{Tr}(A^x) .$$

The equation (2.10) now follows, by simply re-expressing this in terms of $\mathrm{Tr}(A^n)$.

We are now in a position to prove the main result of this Section 2.6.1. Suppose that D_0 and D_1 are as defined above, so that $\mathrm{Tr}(A_0^i) = \mathrm{Tr}(A_1^i)$ for $0 \leq i \leq 2m$. We want to show that $\mathrm{Tr}(A_0^{2m+1}) = \mathrm{Tr}(A_1^{2m+1})$. We have, from the formula (2.10), that for each u = 0 and 1,

$$\mathrm{Tr}(A_u^{2m+1}) = \mathrm{Tr}(B_u^{2m+1}) - \sum_{h=0}^{2m+1} \sum_{\substack{x=0 \\ x\neq 2m+1}}^{h} (-k)^{2m+1-h} r^{h-x} \binom{2m+1}{h} \binom{h}{x} \mathrm{Tr}(A_u^x) \tag{2.11}$$

But this quantity is the same for both values of u: both $\mathrm{Tr}(B_0^{2m+1})$ and $\mathrm{Tr}(B_1^{2m+1})$ are zero since the designs each have two replicates and are resolvable; and the remaining portions of the right-hand side of the equation (2.11) are equal by the hypothesis that $\mathrm{Tr}(A_0^i) = \mathrm{Tr}(A_1^i)$ for $i = 2,3,\ldots.,2m$.

Hence, for resolvable two-replicate designs, the value of the graphical criterion $\underset{=}{C}_{2m+1}$ is fully determined by the values of its predecssors $\underset{=}{C}_2$, $\underset{=}{C}_3$,....., $\underset{=}{C}_{2m}$; in this sense, therefore, it is redundant. (This result is illustrated in a later example: page 176.)

2.6.2 Equivalence of the $\underset{=}{C}_4$-criterion for resolvable two-replicate designs and the $\underset{=}{C}_2$-criterion for their contractions

The second result on resolvable two-replicate designs establishes that the $\underline{\underline{C}}_4$-criterion for them is equivalent to the $\underline{\underline{C}}_2$-criterion for their contractions. Not only does this indicate in a different way from 2.4.3 that the graphical criteria can be seen as an extension of the $\underline{\underline{S}}$-criterion; it also lends weight to our recommendation that to use them would be at least as sensible as the widespread practice of using the $\underline{\underline{S}}$-criterion.

To be precise, what we show is the following. Suppose that D_0 and D_1 are both resolvable two-replicate designs, and suppose, moreover, that their variety concurrence graphs have the same number of circuits of length two. Then these graphs have the same number of circuits of length four (that is, the same number of squares) if, and only if, the variety concurrence graphs of their respective contractions have the same number of circuits of length two.

It will facilitate the exposition if we divide it into two stages. First, we show that the variety concurrence graphs of the designs D_0 and D_1 have the same number of squares if, and only,if, their block concurrence graphs have the same number of squares. Then we show that their block concurrence graphs have the same number of squares if, and only if, the variety concurrence graphs of the contractions have the same number of circuits of length two.

Throughout the proof, matrices pertaining to the two designs will be distinguished by means of the suffices 0 and 1.

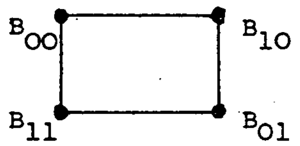
The first part of the proof proceeds by arguments that are similar to those we adopted in 2.6.1 above. We have that the variety adjacency matrices are given by $A_u = N_uN_u^T - 2I_{ks}$, and that the block adjacency matrices are given by $B_u = N_u^TN_u - kI_{2s}$, where

N_O and N_1 are the respective design incidence matrices, and s is the number of blocks in each replicate. Then the number of squares in the variety concurrence graph of the design D_u is the quantity $Tr(A_u^4)$ and the number in the block concurrence graph is $Tr(B_u^4)$. So the first step of the proof entails establishing that $Tr(A_O^4) = Tr(A_1^4)$ if, and only if, $Tr(B_O^4) = Tr(B_1^4)$.

Suppose, on the one hand, that the variety concurrence graphs have the same number of squares: that is, that $Tr(A_O^4) = Tr(A_1^4)$. Now, we have postulated that $Tr(A_O^2) = Tr(A_1^2)$, and, of course, $Tr(A_O) = Tr(A_1)$ (both being O). Also, therefore, since the designs are resolvable and two-replicate, the result of 2.6.1 gives us that $Tr(A_O^3) = Tr(A_1^3)$. Then, by the equation (2.10) in 2.6.1 (with $n = 4$), it follows that $Tr(B_O^4) = Tr(B_1^4)$; that is, that the block concurrence graphs have the same number of squares.

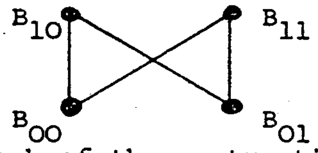
On the other hand, suppose, conversely, that the block concurrence graphs have the same number of squares: that is, that $Tr(B_O^4) = Tr(B_1^4)$. Now, we have that $Tr(B_O^u) = Tr(B_1^u)$ for $u = 1, 2,$ and 3: for $u = 1$ and 3, this follows from equation (2.9) in 2.6.1, and for $u = 2$, it follows from equation (2.10) and the equalities $Tr(A_O^2) = Tr(A_1^2)$ and $Tr(A_O) = Tr(A_1)$. That $Tr(A_O^4)$ and $Tr(A_1^4)$ are equal can then be deduced by deriving formulas for these quantities that, analogously to the equation (2,10), express $Tr(A_u^n)$ in terms of the traces of powers of the matrix B_u. Hence, the variety concurrence graphs have the same number of squares.

This completes the first stage of the proof. Before we go into the formal details of the second stage, we will elucidate how proper squares in the block concurrence graph of a resolvable two-replicate design correspond to proper circuits of length two in the variety concurrence graph of its contraction. The block concurrence graph of such a design is bipartite, with the two sides corresponding to the two replicates. So a proper square in this graph is of the form:



,

where blocks B_{00} and B_{01} belong to one replicate, and blocks B_{10} and B_{11} to the other. The depiction of this square can be altered as follows, to make it more suggestive of a bipartite graph:



,

Then in the design graph of the contraction, the set $\{B_{00}, B_{01}\}$ corresponds to varieties, and the set $\{B_{10}, B_{11}\}$ corresponds to blocks (this being the way in which the contraction is defined). It follows that this pair of varieties makes up a proper circuit of length two: each of the varieties B_{00} and B_{01} lie in each of the blocks B_{10} and B_{11}, and so they are joined by two lines in the variety concurrence graph of the contraction.

Thus, proper squares in the block concurrence graph of a resolvable two-replicate design are associated with proper circuits of length two in its contraction. It is this association which constitutes the essence of the second stage of the proof; it remains only to present the full, formal, details, which we do, as usual, by means of the various adjacency matrices. The block adjacency matrix B_u of the two-replicate design D_u equals:

$$\begin{bmatrix} O & M_u \\ M_u^T & O \end{bmatrix} ,$$

where M_u is the design incidence matrix of the corresponding contraction. Hence:

$$B_u^4 = \begin{bmatrix} (M_u M_u^T)^2 & O \\ O & (M_u^T M_u)^2 \end{bmatrix} .$$

So, $Tr(B_u^4) = Tr((M_u M_u^T)^2) + Tr((M_u^T M_u)^2)$, and, by the properties of trace function, this equals $2Tr((M_u M_u^T)^2)$. Moreover, if A_{cu} is the variety adjacency matrix of the contraction of the design D_u, then, of course, $A_{cu} = M_u M_u^T - kI_s$. (This follows from equation (1.3): in the contraction, there are s varieties, each of which is replicated k times.) Hence $(M_u M_u^T)^2 = A_{cu}^2 + 2kA_{cu} + k^2 I_s$, and so $Tr((M_u M_u^T)^2) = Tr(A_{cu}^2) + k^2 s$. So bringing these points together, we get:

$$Tr(B_u^4) = 2Tr(A_{cu}^2) + 2k^2 s . \qquad (2.12)$$

In other words, the number of squares in the block concurrence graph of the resolvable two-replicate design D_u is twice the number of circuits of length two in the variety concurrence graph of the contraction, plus a constant term $2k^2s$. (This constant term records the circuits of length two that are not proper.) The result of the second stage of the proof follows immediately: from equation (2.12), we deduce that the two block concurrence graphs have the same number of squares (that is, $Tr(B_O^4) = Tr(B_1^4)$) if, and only if, the corresponding contractions have the same number of circuits of length two in their variety concurrence graphs (that is, if, and only if, $\mathrm{Tr}(A_{c0}^2) = \mathrm{Tr}(A_{c1}^2)$).

This completes the proof of the result we stated at the beginning of this Section 2.6.2. So, in the sense defined by this result, the $\underline{\underline{C}}_4$-criterion for resolvable two-replicate designs is equivalent to the $\underline{\underline{C}}_2$-criterion for their contractions. In particular, this establishes the equivalence of special cases of the conjecture which we put forward in 2.5: on the one hand, the conjecture that all $\underline{\underline{A}}$-optimal resolvable two-replicate designs for ks varieties in 2s blocks of size k are optimal on the $\underline{\underline{C}}_4$-criterion (that is, have a minimal number of squares in their variety concurrence graphs); and, on the other hand, the conjecture that all $\underline{\underline{A}}$-optimal designs for s varieties in s blocks of size k are optimal also on the $\underline{\underline{C}}_2$-criterion (that is, have a minimal number of circuits of length two in their variety concurrence graphs). An illustration of the equivalence of these conjectures is contained in the final Section 4.4.3 of the Chapter 4 on α designs, where we investigate at some length its relevance to the search for $\underline{\underline{A}}$-optimal two-replicate α(0,1) designs. (As a matter of fact, the material in that Section is not so much an application of the equivalence to such designs as a re-derivation of it by a method suitable only to them. The advantage of that method will be that it will throw some light on the structure of the variety concurrence graphs of α designs.)

CHAPTER 3

APPLICATION OF THE GRAPHICAL CRITERIA TO CYCLIC DESIGNS

3.1 Introduction

Up till now, we have mostly been describing attributes which differentiate the graphical criteria from the S-criterion. On both mathematical and intuitive grounds, we have argued, the graphical criteria are, collectively, superior to the S-criterion. Now, however, and for most of the rest of the thesis, we turn to an important advantage which the graphical criteria have in common with the S-criterion, and which, in practical applications, renders them preferable to the A-criterion.

This advantage is their ease of calculation. If it were not for this, there would be little reason to question the use of the A-criterion in the first place: for all that the graphical criteria are more rigorously linked to the statistical analysis than is the S-criterion, they remain less so than is the A-criterion. But the problem with the A-criterion (as with the D-criterion and the E-criterion) is that it requires, in general, the calculation of the eigenvalues of the matrix C; and for large designs, this calculation can be laborious, even on a computer. The graphical criteria, like the S-criterion, involve, simply calculating powers of the variety adjacency matrix A. That in itself can provide a considerable saving in computer time; what is more, the graph theory which underlies the C_h-criteria can, in many cases, be used to simplify these calculations even further.

In this Chapter and the next, we give two broad examples of such simplifications. Here, in Chapter 3, we discuss the application of the graphical criteria to cyclic designs; later, in Chapter 4, we deal with the rather more complicated case of application to α designs.

3.2 General form of the variety concurrence graph of a cyclic design

First, in this Section, we make some general observations about the variety concurrence graph of a cyclic design, stating results that will be useful later in the Chapter.

Let D be a cyclic design with incidence matrix N. Then the variety concurrence matrix NN^T is a circulant matrix: that is,

$$NN^T = \sum_{i=0}^{v-1} b_i \Gamma^i \quad , \tag{3.1}$$

for some integers $b_i \geq 0$, where Γ is the v×v basic circulant:

$$\begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 1 & 0 & 0 & \cdots & 0 \end{bmatrix} .$$

Furthermore, the numbers b_i can be derived from the initial blocks of the design, as follows. These initial blocks yield a paired-difference set $\{1^{a_1}, 2^{a_2}, \ldots\ldots, m^{a_m}\}$, where m equals $\frac{1}{2}v$ if v is even and $\frac{1}{2}(v-1)$ if v is odd, and a_i is the number of times the quantity i (modulo v) appears among differences between pairs of elements in the same initial block (with the qualification that if k = 2 and one of the initial blocks is $\{0, \frac{1}{2}v\}$, then the difference

½v arising from this block is counted only once in a_m, not twice; similar remarks apply if k = v-2 and one of the initial blocks is {1, 2,...., ½v-1, ½v+1,...., v-1} . See John, Wolock and David (1972) page 16.) Then the numbers b_i are given by the following expressions:

$$\begin{aligned} &\text{(i)} \quad b_0 = r \\ &\text{(ii)} \quad b_i = a_i = b_{v-i} \quad (1 \leq i \leq m) . \end{aligned} \tag{3.2}$$

(All the above theory has been developed fully by John (1966), David (1963), and John, Wolock and David (1972).)

It follows immediately from (3.1), (3.2), and (1.3) that if A is the adjacency matrix of the variety concurrence graph of D, then

$$A = \sum_{i=1}^{v-1} b_i \Gamma^i \quad . \tag{3.3}$$

So the graph is in fact a circulant graph (defined by Biggs (1974), page 16, as a graph whose adjacency matrix is a circulant matrix, as in (3.3).) However, this observation does not appear to be particularly informative: the thoery of circulant graphs is less fully developed than the theory of cyclic designs.

This notation also has the important advantage of revealing clearly when a cyclic design is balanced. Balance occurs whenever each pair of varieties in the design concur together equally often, and this will happen whenever the numbers a_i are all equal. (The number of times two varieties i and j concur is the number of times the quantity $i \dot{-} j$[†] occurs among the differences between pairs of elements in the same initial block.) In the case of balance, the

[†] In this Chapter, a dot above the operator, as here, means that the operation is to be conducted modulo v.

variety concurrence graph is, of course, a multiple of the complete graph on v points: this is something we have established, in 1.4.1(a), for all balanced designs. It emerges also from equation (3.3), since if the numbers a_i are each equal to a, say, then the numbers b_i (for $i \geq 1$) each equal a, and so $A = a \sum_{i=1}^{v-1} \Gamma^i$, which equals $a(J_v - I_v)$.

Of particular interest to us later will be balanced cyclic designs which have only one initial block. (See Section 4.4.3(b).) In that case, the initial block is a perfect difference set (a definition of which appears later, page 205): that is, if the initial block is $\{x_0, x_1, \ldots\ldots, x_{k-1}\}$, then each of the residues 1, 2,... ..., v-1 occurs equally often among the k(k-1) differences $x_i \dot{-} x_j$ $(i \neq j)$.

We will now consider in more detail two aspects of the application of the graphical criteria to cyclic designs. First (in 3.3) we will use equation (3.3) to show that the criteria can be calculated in a very straightforward manner from the numbers b_i; and, second (in 3.4), we will indicate some of the ways in which these criteria are - on account of their basis in the variety concurrence graph - relevant to the problem of adding further initial blocks to existing cyclic designs (a problem that arises in the work of John (1981)).

3.3 Ease of calculation of the graphical criteria for cyclic designs

First, then, we show that equation (3.3) allows the graphical criteria to be calculated very easily. This follows almost directly from (3.3). A simple mathematical induction argument establishes the following expression for A^h (where h is any positive integer):

$$A^h = \sum_{i_1,..,i_h=1}^{v-1} b_{i_1} b_{i_2} \ldots\ldots b_{i_h} \Gamma^{i_1+\ldots+i_h} \quad . \tag{3.4}$$

Now, the traces of powers of the matrix Γ take particularly simple forms, since the diagonal elements of Γ^n are O unless Γ^n is the identity matrix (which happens whenever n is a multiple of v). That is, $\mathrm{Tr}(\Gamma^n) = 0$ if $n \neq 0$ (modulo v), and $\mathrm{Tr}(\Gamma^n) = v$ if $n = 0$ (modulo v). Using this in combination with equation (3.4), we get:

$$\mathrm{Tr}(A^h) = v \sum_{\substack{i_1,..,i_h=1 \\ i_1 \dotplus .. \dotplus i_h=0}}^{v-1} b_{i_1} b_{i_2} \ldots\ldots b_{i_h} \quad . \tag{3.5}$$

This expression for $\mathrm{Tr}(A^h)$ can be simplified further. First of all, we separate out the summation involving i_h, thus:

$$\mathrm{Tr}(A^h) = v \sum_{i_1,..,i_{h-1}=1}^{v-1} \sum_{\substack{i_h=1 \\ i_h=v \dot{-} i_1 \dot{-} .. \dot{-} i_{h-1}}}^{v-1} b_{i_1} \ldots. b_{i_{h-1}} b_{i_h}$$

The innermost summation can now be seen to pick out, for given values of $i_1,\ldots..,i_{h-1}$, those values of i_h which satisfy:

(a) i_h lies in the set $\{1,2,\ldots..,v-1\}$;

and (b) $i_h = v \dot{-} i_1 \dot{-} i_2 \dot{-} \ldots.. \dot{-} i_{h-1}$.

So we can remove the index i_h altogether by replacing it with $v \dot{-} i_1 \dot{-} i_2 \dot{-} \ldots.. \dot{-} i_{h-1}$ (from condition (b)), and by stipulating that $v \dot{-} i_1 \dot{-} i_2 \dot{-} \ldots.. \dot{-} i_{h-1}$ should not equal O (condition (a)). Thus the entire expression for $\mathrm{Tr}(A^h)$ becomes:

$$\mathrm{Tr}(A^h) = v \sum_{\substack{i_1,..,i_{h-1}=1 \\ v\dot{-}i_1\dot{-}..\dot{-}i_{h-1}\neq 0}}^{v-1} b_{i_1}.....b_{i_{h-1}} b_{v\dot{-}i_1\dot{-}..\dot{-}i_{h-1}} .$$

We can also get round the complication in the summation - that is, the stipulation that $v \dot{-} i_1 \dot{-} ..... \dot{-} i_{h-1}$ should not equal 0 - by re-defining b_0 to be 0 (instead of r). Henceforth, this will always be assumed to be the case, and so we will have:

$$\mathrm{Tr}(A^h) = v \sum_{i_1,..,i_{h-1}=1}^{v-1} b_{i_1}.....b_{i_{h-1}} b_{v\dot{-}i_1\dot{-}..\dot{-}i_{h-1}} .$$

There is one further simplification we can introduce: since $b_{v\dot{-}i} = b_i$ for each i, we can replace the quantity $v \dot{-} i_1 \dot{-} ... \dot{-} i_{h-1}$ by the quantity $i_1 \dotplus ..... \dotplus i_{h-1}$. So we have, finally, the following expression for the number of circuits of length h:

$$\underline{\underline{C}}_h = \mathrm{Tr}(A^h) = v \sum_{i_1,..,i_{h-1}=1}^{v-1} b_{i_1}.....b_{i_{h-1}} b_{i_1\dotplus..\dotplus i_{h-1}} . \qquad (3.6)$$

(This will be considerably easier to calculate than the generalised inverse used in John, Wolock and David (1972), page 4.)

For convenience, we will let d_h denote the sum on the right-hand side of equation (3.6), so that $\underline{\underline{C}}_h = vd_h$.

For example, for the first few values of h, the expression (3.6) gives for the graphical criteria:

$$\underline{\underline{C}}_2 = v \sum_{i=1}^{v-1} b_i^2 ; \qquad (3.7)$$

$$\underline{\underline{C}}_3 = v \sum_{i,j=1}^{v-1} b_i b_j b_{i\dotplus j} ; \qquad (3.8)$$

$$\underline{\underline{C}}_4 = v \sum_{i,j,h=1}^{v-1} b_i b_j b_h b_{i\dotplus j\dotplus h} . \qquad (3.9)$$

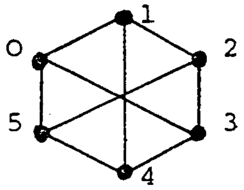
We now give two examples of this.

Example (1)

This example is of cyclic designs for 6 varieties in blocks of size 2, each variety being replicated 3 times. John, Wolock and David (1972) list the best design as having the initial blocks {0, 1} and {0, 3} . Then the design - which we will call D_0 - is:

0 1
1 2
2 3
3 4
4 5
5 0
0 3
1 4
2 5 ,

with harmonic mean efficiency factor $\underline{\underline{A}} = .5556$. The variety concurrence graph of D_0 is:



From these initial blocks, the paired-difference set is $\{1^1, 2^0, 3^1\}$. Hence, $b_1 = b_3 = b_5 = 1$ and $b_2 = b_4 = 0$. So $\sum_{i=1}^{v-1} b_i^2 = 3$, and therefore $\underline{\underline{C}}_2 = 18$. That is, there are 18 circuits of length two. (Since there is no pair of points joined by more than one line, these circuits are simply the ones which arise by traversing the same line twice, of which there are, since each point has valency r(k-1), vr(k-1), which is 18.) The number of circuits of length 3 is, from (3.8),

$$v \sum_{i,j=1}^{v-1} b_i b_j b_{i+j}$$

$$= 6 \times \sum_{i=1}^{5} b_i (b_{i+1} + b_{i+3} + b_{i+5})$$

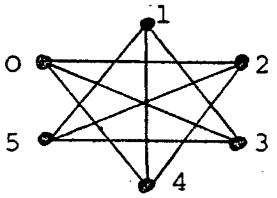
$$= 0 .$$

This is in fact obvious from the graph diagram.

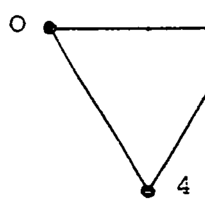
Bearing in mind the isomorphisms of cyclic designs described by John, Wolock, and David (1972), page 3, we find that, essentially, the only other possibility for these values of v, k, and r has the initial blocks {0, 2} and {0, 3}. That is, the design - which we will call D_1 - is:

0 2
1 3
2 4
3 5
4 0
5 1
0 3
1 4
2 5.

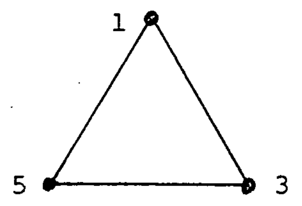
This design has the variety concurrence graph:



For this design, $\underline{\underline{A}}$ = .5319, which is markedly worse than the first design. Here, the paired-difference set is $\{1^0, 2^1, 3^1\}$. So $b_1 = b_5 = 0$ and $b_2 = b_3 = b_4 = 1$. So $\sum_{i=1}^{5} b_i^2 = 3$, the same as in the first case. (Again, there are no points joined by more than one line.) But the designs D_0 and D_1 can be distinguished on the $\underline{\underline{C}}_3$-criterion: it is obvious from the diagram of the variety concurrence graph of D_1 that this second design does contain circuits of length three. By equation (3.8), the number of such circuits is 12. All of these arise from blocks which are generated by the first initial block: the two configurations:



and



each contribute six circuits (orientation and starting point being, as always, significant).

Example (2)

Here, we look at designs for 12 varieties in blocks of size 2, each variety being replicated 4 times. The Table 1 (page 98) lists the distinct designs (after, that is, applying the isomorphisms of John, Wolock and David (1972)). They are listed according to decreasing value of $\underline{\underline{A}}$ (in column 2). All the designs have the same number - namely 48 - of circuits of length 2. The numbers of circuits of length 3 are listed in column 3. Notice that the pattern of the quantities $\underline{\underline{C}}_3$ bears out the conjecture in 2.5: that is, the higher the value of the harmonic mean efficiency factor, the smaller the number of circuits of length three.

The $\underline{\underline{C}}_3$-criterion is, however, in some cases, insufficiently sensitive to distinguish among designs which have different values of the harmonic mean efficiency factor; for these, we have to look at the next graphical criterion, $\underline{\underline{C}}_4$ or number of circuits of length four. Thus, for example, this criterion serves to distinguish among the first four designs in Table 1 (all of which have no circuits of length three) in the same way as does the harmonic mean efficiency factor: they have, respectively, 408, 504, 504, and

Table 1

Cyclic designs for 12 varieties in 24 blocks of size 2.

Comparison of harmonic mean efficiency factor with

number of triangles.

	Column 1	Column 2	Column 3
	Initial blocks	$\underline{\underline{A}}$	$\underline{\underline{C}}_3$
(1)	{o, 2}, {o, 3}	.4793	0
(2)	{o, 1}, {o, 3}	.4705	0
(3)	{o, 3}, {o, 5}	.4705	0
(4)	{o, 1}, {o, 5}	.4648	0
(5)	{o, 1}, {o, 4}	.4595	24
(6)	{o, 4}, {o, 5}	.4595	24
(7)	{o, 3}, {o, 4}	.4580	24
(8)	{o, 1}, {o, 2}	.4024	72
(9)	{o, 2}, {o, 5}	.4024	72

528 circuits of length four; again, therefore, the conjecture in 2.5 is borne out in practice.

3.4 Application of the graphical criteria to adding and extending initial blocks

The second aspect of the application of the graphical criteria to cyclic designs which we will investigate concerns a problem raised by John (1981) in the context of simplifying the search for efficient cyclic designs. An important part of the simplification which he suggests is the following. If we are looking for a design with x initial blocks, then there is, he says, little point in searching through all possible combinations of x initial blocks: we do not risk much loss of efficiency by searching sequentially. That is, we search only those designs which contain among their x initial blocks the set of x-1 which have already been chosen - by a similar method - for the design with x-1 initial blocks. Thus, for example, John proposes a design for 9 varieties, each replicated 6 times, with three initial blocks of size 2: {0, 1}, {0, 3}, and {0, 2}. He arrives at this by adding a third initial block to the first two of these (which form the most efficient cyclic design on the $\underline{\underline{A}}$-criterion for 9 varieties, each replicated 4 times in blocks of size 2: see the catalogue, John, Wolock and David (1972)). The resulting design has harmonic mean efficiency factor $\underline{\underline{A}}$ = .5418. This compares quite well with the efficiency factor for the best cyclic design listed in the catalogue, which has initial blocks {0, 1}, {0, 2}, {0, 4} , and $\underline{\underline{A}}$ = .5455.

We show here how this sequential searching can be facilitated by invoking the graphical criteria: specifically, by exploiting their basis in the variety concurrence graph. At first (Sections 3.4.1 to 3.4.4), we examine the case in which we want to find an efficient design with two initial blocks of size 2; later (Section 3.4.5), we indicate how our approach could be generalised.

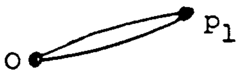
So, first of all, we start with one initial block IB_a, equal to, say, {0, a} (where a $\neq$ 0), and we want to find the best choice of b (not equal to 0) to make up the second initial block IB_b equal to {0, b} . We investigate how to choose b in such a way that the resulting variety concurrence graph has as few circuits as possible of certain specified lengths n.

As a preliminary to this, we find out (in 3.4.1, 3.4.2, and 3,4.3) how many circuits of length n are produced by the different choices of a and b (where n will be, respectively, 2, 3, and 4). Then (in 3.4.4) we apply the results to this search for a second initial block.

In a cyclic design, the number of circuits of a given length n which start at each point is the same. This is because varieties i and j concur if, and only if, varieties $i \dotplus 1$ and $j \dotplus 1$ concur. Or, in the variety concurrence graph, i is adjacent to j if, and only if, $i \dotplus 1$ is adjacent to $j \dotplus 1$. Thus, the set of circuits of length n can be partitioned (according to starting point) into v equal-sized groups. (Notice that this is reflected in the expression (3.5); the number of circuits of length n starting at any given point p_0 is, therefore, d_n.) In enumerating circuits of length n it is, in consequence, only necessary to enumerate those which start at O. (One further thing to note is that the lines which are generated by the initial block {O, x} join varieties i and $i \dotplus x$ for $0 \leq i \leq v-1$.)

3.4.1 Enumeration of circuits of length two

As we explained earlier (page 72), the number of circuits of length two which contain only one line is a simple function of the quantities v, r, and k: it is vr(k-1), which is, here, 4v. (Strictly speaking, we showed this for α designs only, but the reasoning we gave applies equally here.) However, the number of circuits of length two which contain two distinct lines depends also on the values of the numbers a and b. Such circuits look like:



(where the point p_1 is not the same as the point O). Each of these lines corresponds to addition or subtraction of one of a or b from its endpoint O. There are, therefore, four distinct cases, depending on which, if any, of the lines comes from block IB_a.

<u>Case (i)</u>: both lines come from block IB_a.

Since the two lines are distinct, and since they both arise from this one initial block, one of them must correspond to addition of a to the endpoint O, and the other to subtraction of a from O. From this we can derive the necessary condition: $2a = 0$. However, this implies either that a is O (which we have already excluded), or that a is ½v. Now, by convention (see above, page 91), only half of the set of blocks generated by {O, ½v} is included in the design. So there are in fact no circuits of length two arising in this way.

<u>Case (ii)</u>: only the first line comes from block IB_a.

Here, by considering this first line, p_1 is a or $\dot{-}a$; and then, by considering the second line, $0 = p_1 \dot{+} b$ or $0 = p_1 \dot{-} b$. This gives four possible equations, one of which must hold for any such circuit:

$$0 = a \dot{+} b \quad ;$$

$$0 = a \dot{-} b \quad ;$$

$$0 = \dot{-}a \dot{+} b \quad ;$$

$$0 = \dot{-}a \dot{-} b \quad .$$

Conversely, each of these yields v circuits of length two.

Case (iii): only the second line comes from block IB_a.

The roles of a and b are interchanged here from Case (ii), and so we get four equations:

$$0 = b \dotplus a \quad ;$$

$$0 = b \mathbin{\dot{-}} a \quad ;$$

$$0 = \dot{-}b \dotplus a \quad ;$$

$$0 = \dot{-}b \mathbin{\dot{-}} a \quad .$$

Hence, bringing together Cases (ii) and (iii), we have that each of the two equations:

$$0 = a \dotplus b \tag{A1}$$

$$\text{and } 0 = a \mathbin{\dot{-}} b \tag{A2}$$

yields 4v circuits of length two.

Case (iv): no lines come from block IB_a.

This parallels Case (i) (b being substituted for a), and so no circuits arise.

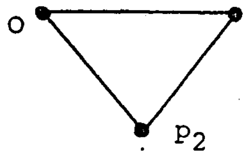
So, in summary, if there are n(A1) equations of the form (A1), and n(A2) of the form (A2), then there are 4v(n(A1) + n(A2)) circuits of length two in which the two lines are different, and so 4v + 4v(n(A1) + n(A2)) circuits of length two altogether.

3.4.2 Enumeration of circuits of length three

Matters become much more complicated when we are enumerating circuits of lengths greater than two. We can, however, simplify the description somewhat by making use of a certain symmetry which exists between a and b. It will be noticed in the enumeration of circuits of length two above that Cases (i) and (ii) are mirrored by, respectively, Cases (iv) and (iii) - in the sense that (iv) can be obtained from (i), and (iii) from (ii), by interchanging a and b. So we could have enumerated circuits of length two in two stages. We would have given the replicates involved in the two lines the general labels y_0 and (if appropriate - that is, if the second one is different) y_1. Then the first stage would have been to allocate one of a and b to y_0, and the other to y_1; and the second stage would have involved enumerating, for each allocation - and by means of general equations analogous to (A1) and (A2) in terms of y_0 and y_1 - the corresponding number of circuits that would have been generated.

There would have been little benefit in doing this for circuits of length two: it would have yielded no significant simplification. However, such an approach will help clarify the enumeration of circuits of lengths three (in this Section 3.4.2) and four (in the next Section 3.4.3); it will, moreover, and much more importantly (as we will mention at the end, in Section 3.4.5), suggest how the material we present could be generalised for block size greater than 2, and for more than two initial blocks.

Any circuit of length three (or triangle as we will often call it) must be proper; that is, its lines must be distinct. (This is simply to state an obvious fact of geometry.) The paradigmatic triangle starting at O and traversing clockwise can be depicted as:



;

that is, this represents the circuit $0 - p_1 - p_2 - 0$. Each line here corresponds to the addition or subtraction of one of the quantities a or b. We will use the label y_0 to refer to the quantity (a or b) which is associated with the first line; if the other quantity is associated with some line, we will call it y_1. Then we will classify the circuits according to the pattern of appearances of these labels - that is, according to their order of occurrence and the signs (addition or subtraction) associated with them (in the direction of the circuit). This will enable us to derive general formulas for the numbers of triangles arising out of each allocation of the quantities a and b to the labels y_0 and y_1.

So in this scheme there are four Cases:

(i) all three lines have the same label;

(ii) the first and second lines have the label y_0, and the third has the label y_1;

(iii) the first and third lines have the label y_0, and the second has the label y_1;

(iv) the first line has the label y_0, and the second and third have the label y_1.

We discuss each of these in turn.

Case (i): all three lines correspond to the same block.

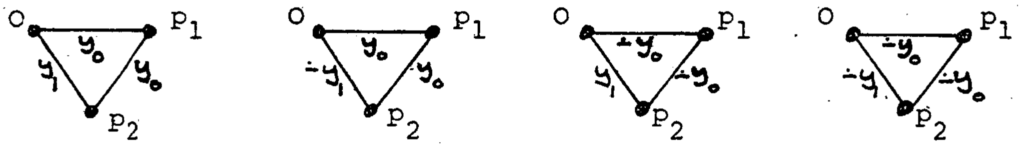
Here, since the points in the triangle are distinct, either all three lines correspond to addition of y_0, or else all three correspond to subtraction of y_0. (If, for instance, the first line were to correspond to addition of y_0 and the second to subtraction of y_0, then the point p_2 would be $y_0 \dot{-} y_0$ which is 0.) Either way we have that:

$$3y_0 = 0 \ . \qquad (B1)$$

That is, if there are n(a, B1) equations (B1) for $y_0 = a$ and n(b, B1) for $y_0 = b$, then there are altogether 2v(n(a, B1) + n(b, B1)) triangles all three of whose lines arise from the same initial block. (Of course, n(y_0, B1) is either 0 or 1 for each value of y_0.)

Case (ii): first and second lines have the label y_0 and third line has the label y_1.

In this Case, either both the first and the second lines correspond to addition of y_0, or both correspond to subtraction of y_0; and for each of these, the third line can correspond to either addition or subtraction of y_1. So the triangles here can be of the following four types:



For the first and fourth of these we have the equation:

$$2y_0 \dotplus y_1 = 0 \ , \qquad (B2)$$

and for the second and third, the equation:

$$2y_0 \dot{-} y_1 = 0 \ . \qquad (B3)$$

So, if there are n(y_0, y_1, B2) equations of the form (B2) and n(y_0, y_1, B3) equations of the form (B3) (for the labels y_0 and y_1), then the number of triangles arising under this Case (ii) is:

2v(n(a, b, B2) + n(b, a, B2) + n(a, b, B3) + n(b, a, B3)).

Cases (iii) and (iv): the theory for these cases is similar to what we have already done. The number arising under each turns out to be, as in Case (ii):

$$2v(n(a, b, B2) + n(b, a, B2) + n(a, b, B3) + n(b, a, B3)).$$

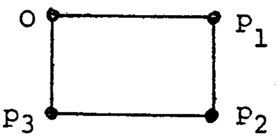
In summary, therefore, we have that the total number of triangles is:

$$2v(n(a, B1) + n(b, B1)) +$$
$$6v(n(a, b, B2) + n(b, a, B2) + n(a, b, B3) + n(b, a, B3)).$$

3.4.3 Enumeration of circuits of length four

Here, to simplify matters further, we will confine our attention to designs which have no multiple concurrences (and so no multiple lines in their variety concurrence graphs); and we will also assume that neither a nor b equals $\frac{1}{2}v$. The material could be extended to more general types of design; but that would tend to sacrifise illustrative clarity.

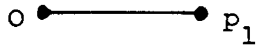
We will often call circuits of length four squares, and we will depict the typical square as:



(starting at O and traversing clockwise: that is, O - p_1 - p_2 - p_3 - O). Immediately, a new complication arises that was not present with triangles: these four vertices need not be distinct. Certainly adjacent vertices must be different; but it remains possible that O is the same as p_2 or that p_1 is the same as p_3. We will deal with each of the resulting cases in turn, and we will find that in all but those cases where all the vertices are different, the number is a simple function of v, r, and k.

(i) $p_2 = O$ and $p_1 = p_3$

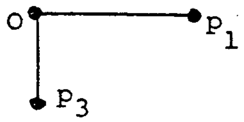
Here, since there are no multiple lines, the circuit involves traversing the same line four times:



There are altogether r(k-1) - that is, 4 - lines which O lies on, and so the total number of such circuits is 4v.

(ii) $p_2 = O$ and $p_1 \neq p_3$

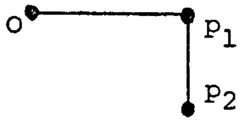
In this case, again since there are no multiple lines, the circuit looks like:



The number of these starting at O is the same as the number of pairs of distinct lines starting at O, which is r(k-1)(r(k-1) - 1), and this equals 12. So the total number of such circuits is 12v.

(iii) $p_2 \neq O$ and $p_1 = p_3$

This time, the circuit is:



Bearing in mind again that there are no multiple lines, there are r(k-1) lines joining O to points p_1, and there are, for each such p_1, a further r(k-1) - 1 lines joining p_1 to a point p_2 different from O. So the total number of such circuits is vr(k-1)(r(k-1) - 1), which equals 12v again.

(iv) $p_2 \neq O$ and $p_1 \neq p_3$

It is in this case that difficulties arise: the number of squares is not simply a function of v, r, and k. Again, as for triangles, we will divide the enumeration into two stages: the first stage will involve allocating the quantities a and b to the general labels y_0 and y_1; and, then, the second stage will enumerate the number of squares for each allocation, by means of general formulas based on counting certain relationships between y_0 and y_1 which we will derive from the appropriate square diagrams. Again, too, y_0 will always be the label associated with the first line.

There are four general Cases (not exactly analogous to the types we formed for triangles):

(a) all four lines have the same label;

(b) three lines have the label y_0 and one has the label y_1;

(c) two lines have the label y_0 and two have the label y_1;

(d) one line has the label y_0 and three have the label y_1.

We will deal with each of these in turn, and further sub-divide them as appropriate.

<u>Case (a)</u>: all four lines have the same label.

Here, since the vertices of the square are distinct, either all four lines correspond to addition of y_0, or else all correspond to subtraction of y_0. So if there are $n(y_0, C1)$ equations of the following type:

$$4y_0 = 0 , \tag{C1}$$

then there are $2v(n(a, C1) + n(b, C1))$ squares arising under this Case (a). (Notice that the reason why such equations do yield squares of the appropriate type is that $y_0 \neq \frac{1}{2}v$.)

<u>Case (b)</u>: three lines have label y_0 and one has label y_1.

It will be convenient to sub-divide this Case into three sub-cases, according to which line has the label y_1:

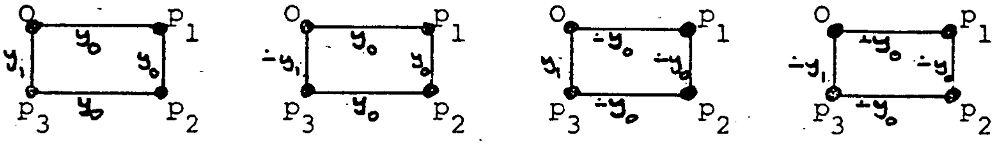
I the fourth line has the label y_1;

II the third line has the label y_1;

III the second line has the label y_1.

(Of course, the first line cannot have the label y_1: our system of labelling presupposes that the first line has the label y_0.)

I. Since the vertices are distinct, either there are three appearances of $\dot{+}y_0$ or three appearances of $\dot{-}y_0$; and for each of these, the fourth line can correspond either to y_1 or to $\dot{-}y_1$. That is, there are four possible squares here:



The first and fourth of these arise from the equation:

$$3y_0 \dot{+} y_1 = 0 , \qquad \text{(C2)}$$

and the second and third from the equation:

$$3y_0 \dot{-} y_1 = 0 . \qquad \text{(C3)}$$

So if there are $n(y_0, y_1, C2)$ equations like (C2) and $n(y_0, y_1, C3)$ equations like (C3), then the total number of squares here is:

$$2v(n(a, b, C2) + n(b, a, C2) + n(a, b, C3) + n(b, a, C3)).$$

II. and III. We get, by similar arguments, exactly the same number for each of these other two sub-cases.

Case (c): two lines have the label y_0 and two have the label y_1.

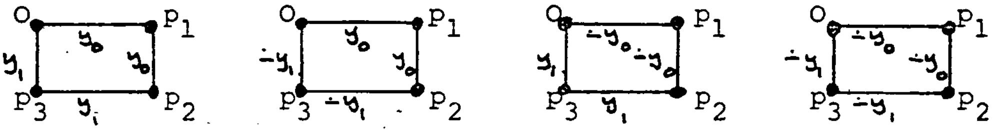
Again, it will be useful to sub-divide this Case into three sub-cases:

I the first two lines have the label y_0 and the last two have the label y_1;

II the first and third lines have the label y_0 and the second and fourth have the label y_1;

III the first and fourth lines have the label y_0 and the second and third have the label y_1.

I. Again since the vertices are distinct, there are four possibilities:



The first and fourth of these arise from the equation:

$$2y_0 \dot{+} 2y_1 = 0 \quad ; \qquad \text{(C4)}$$

and the second and the third from the equation:

$$2y_0 \dot{-} 2y_1 = 0 \quad . \qquad \text{(C5)}$$

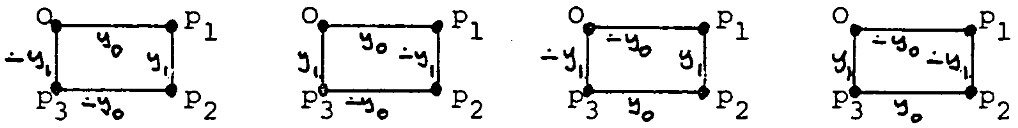
So, using notation analogous to that which we have already employed, we have that the total number of squares here is:

$$2v(n(a, b, C4) + n(b, a, C4) + n(a, b, C5) + n(b, a, C5)).$$

Obviously, moreover, there is a symmetry in these equations between a and b, and so n(a, b, C4) = n(b, a, C4) and n(a, b, C5) = n(b, a, C5). So the number of squares in this sub-case is:

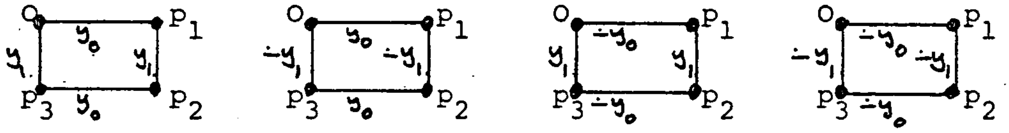
$$4v(n(a, b, C4) + n(a, b, C5)).$$

II. This is the most complicated of all the sub-cases we have to deal with. However, certain combinations of the signs $\dot{+}$ and $\dot{-}$ with the labels can be excluded immediately: any combination, to be precise, which gives rise to equations $2y_0 = 0$ or $2y_1 = 0$, since we have stipulated that a and b do not equal 0 or ½v. Moreover, certain other combinations of signs with labels correspond to unavoidable squares: that is, they give rise to squares regardless of the values of a and b. These unavoidable squares are:



(That is, there are v squares of each of these forms, once the various possible starting points are taken into account. So there are altogether - taking account also of the two possible allocations of a and b to y_0 and y_1 - 8v unavoidable squares of this type.)

Then, again using arguments very similar to those we have already employed, we find that the other possibilities which can arise are of the forms:



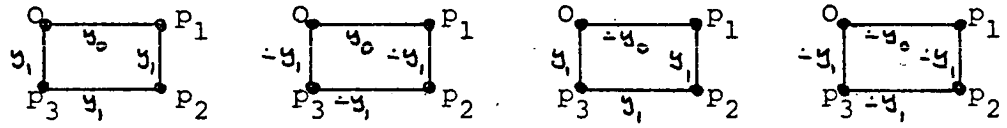
The first and fourth of these arise from equation (C4) again, and the second and third from equation (C5). So, as in I, we get that the number of squares is:

$$4v(n(a, b, C4) + n(b, a, C5)).$$

III. This is similar to I above, and we get exactly the same number here as there.

<u>Case (d)</u>: one line has the label y_0 and three have the label y_1.

Here, since y_0 must occur with the first line, and since, again, all the vertices are distinct, the only four possibilities are:



The first and fourth arise from the equation:

$$y_0 \dotplus 3y_1 = 0 \quad , \tag{C6}$$

and the second and third from the equation:

$$y_0 \dot{-} 3y_1 = 0 \quad . \tag{C7}$$

Moreover, the numbers of these equations are related to the numbers of equations of the forms (C2) and (C3) by interchanging a and b: thus, for example, $n(a, b, C6)$ is the same as $n(b, a, C2)$. So the total number here is:

$$2v(n(b, a, C2) + n(a, b, C2) + n(b, a, C3) + n(a, b, C3)).$$

So, summarising, we have enumerated circuits of length four as follows. From (i), (ii) and (iii) (page 107), and the formula for unavoidable squares in Case (c) II (page 111), there are altogether $4v + 12v + 12v + 8v$ - that is, $36v$ - unavoidable circuits of length four. From Case (a) (page 108), there are

$$2v(n(a, C1) + n(b, C1))$$

squares arising from equations like (C1). From Case (b) I, II, and III (page 109) and from Case (d) (this page), there are

$$8v(n(a, b, C2) + n(b, a, C2) + n(b, a, C3) + n(a, b, C3))$$

squares arising from equations like (C2) and (C3). And, finally, from Case (c) I and III (pages 110 and 111) and from the last part of Case (c) II (page 111), there are

$$12v(n(a, b, C4) + n(a, b, C5))$$

squares arising from equations like (C4) and (C5).

3.4.4 Application to John's method of searching for efficient cyclic designs

We can facilitate the application of these formulas to the problem raised by John (which we mentioned at the beginning, page 99) by bringing them together in Table 2 (page 113). The point of doing this is to make it easier to see exactly how many squares arise from each of the various relationships. (We have also adapted the numbering of the relationships, for ease of reference.) Further to the circuits listed in the Table there are, of course, 4v unavoidable circuits of length two (containing only one line), and 36v unavoidable circuits of length four.

Table 2

Enumeration of circuits of lengths 2, 3, and 4 in cyclic designs for v varieties with initial blocks {0, a} and {0, b}.

		Relationship	Number of circuits	
(A)	Length two	$a \dotplus b = 0$	4v	(A1)
		$a \dot{-} b = 0$	4v	(A2)
(B)	Length three	$3a = 0$	2v	$(B1)_1$
		$3b = 0$	2v	$(B1)_2$
		$2a \dotplus b = 0$	6v	$(B2)_1$
		$2b \dotplus a = 0$	6v	$(B2)_2$
		$2a \dot{-} b = 0$	6v	$(B3)_1$
		$2b \dot{-} a = 0$	6v	$(B3)_2$
(C)	Length four	$4a = 0$	2v	$(C1)_1$
		$4b = 0$	2v	$(C1)_2$
		$3a \dotplus b = 0$	8v	$(C2)_1$
		$3b \dotplus a = 0$	8v	$(C2)_2$
		$3a \dot{-} b = 0$	8v	$(C3)_1$
		$3b \dot{-} a = 0$	8v	$(C3)_2$
		$2a \dotplus 2b = 0$	12v	(C4)
		$2a \dot{-} 2b = 0$	12v	(C5)

We now describe three examples of how these rules can be used in the choice of the two initial blocks of size two. We will, as always (motivated by the conjecture in 2.5), invoke the graphical criteria sequentially: first, we will find those designs which have a minimal number of circuits of length two; then, among these, we will find those designs which have a minimal number of circuits of length three; and, finally, among these, we will find those designs which have a minimal number of circuits of length four. Each of these processes of selection will use the relevant rules in the Table: calculating numbers of circuits of length n by means of these rules will, in general, be much simpler than finding $Tr(A^n)$ by unrefined computation (or, even, frequently, by taking advantage of the savings made possible by equation (3.6)).

To shorten the examples, we will consider only designs in which a is 1 (the analysis for other values of a being similar); and we will restrict b to the set {1, 2,....., m} (this being all that is necessary: all other choices of b correspond to the negatives, modulo v, of elements in this set, and the form of the relationships in the Table obviously implies that the numbers of circuits of each of the three lengths for a second initial block {0, -b} are the same as the corresponding numbers for a second initial block {0, b}).

Example (1)

Here, we look for a design with 8 varieties replicated 4 times. So m is 4, and hence we will choose b in the set {1, 2, 3}. (We would not choose b equal to 4, since {0, 4} would provide only three replications of each variety.)

Criterion $\underline{C_2}$: if a is 1, then to avoid proper circuits of length two, we must (by (A) in the Table 2) choose b in the set {2, 3}.

Criterion $\underline{C_3}$: (B1) cannot hold (since 3 is not a factor of 8), and $(B2)_2$ cannot hold since a is 1. 2a = 2, and so $(B2)_1$ will not hold for $b \leq 3$. To avoid $(B3)_1$, it is necessary to choose b not equal to 2. Hence we must choose b equal to 3. (Notice that then $(B3)_2$ also is avoided.)

Thus, avoiding circuits of length two and three leads us to the choice b = 3. That is, the two initial blocks are {0, 1}, and {0, 3} . (This is the design given by John, Wolock and David (1972).)

Example (2)

This one is for 9 varieties each replicated 4 times. So m is 4, and hence we will choose b in the set {1, 2, 3, 4} .

Criterion $\underline{C_2}$: if a is 1, then to avoid proper circuits of length two we must (by (A)) choose b in the set {2, 3, 4} .

Criterion $\underline{C_3}$: in this example (unlike the previous one), we cannot completely eliminate circuits of length three, for $(B3)_1$ would exclude b = 2, $(B1)_2$ would exclude b = 3, and $(B2)_2$ would exclude b = 4. Accordingly, unable to avoid circuits of length three, the best we can do is to make that choice of b in {2, 3, 4} which gives rise to as few such circuits as possible. Both of b equal to 2 and b equal to 4 give 6v - that is, 54 - circuits of length three. But b equal to 3 gives only 2v - that is, 18 - circuits of length three.

So we choose b equal to 3, giving initial blocks $\{0, 1\}$ and $\{0, 3\}$. (Again, this is the one listed by John, Wolock and David.)

Example (3)

10 varieties each replicated 4 times. So m is 5, and hence we will choose b in the set $\{1, 2, 3, 4\}$.

Criterion $\underline{\underline{C}}_2$: if a is 1, then to avoid proper circuits of length two, we choose b in the set $\{2, 3, 4\}$.

Criterion $\underline{\underline{C}}_3$: $(B1)_2$ cannot hold (since 3 is not a factor of 10). $2a = 2$, and so $(B2)_1$ will not hold if $b \leq 4$. Neither will $(B2)_2$ (since 2 is not a factor of 9), or $(B3)_2$ (since 2 is not a factor of 11). Moreover, to avoid $(B3)_1$, it is necessary to choose b not equal to 2. So to avoid triangles, we must choose b in the set $\{3, 4\}$.

Thus, in this example, applying criteria $\underline{\underline{C}}_2$ and $\underline{\underline{C}}_3$ does not exhaust all choices. So we turn to $\underline{\underline{C}}_4$ to choose between b equal to 3 and b equal to 4.

Criterion $\underline{\underline{C}}_4$: it is not possible to avoid completely circuits of length four, since $(C3)_1$ excludes b equal to 3 and (C4) excludes b equal to 4. So we have to find which of the two choices of b yields the smaller number of such circuits. If b is 3, then $(C2)_2$ and $(C3)_1$ hold, and so there are $8v + 8v$ - that is, 160 - squares. But if b is 4, then only (C4) holds, and then there are $12v$ - that is, 120 - squares. So the criterion $\underline{\underline{C}}_4$ leads us to choose $b = 4$. (The number of unavoidable squares referred to on page 113 is, of course, the same for any choice of b: namely, $36v$ which equals 360.)

So, applying these three graphical criteria with a equal to 1 leads us to choose the design with initial blocks {0, 1} and {0, 4}. (This, too, is the one listed in the catalogue.)

3.4.5 Extension of the theory to general cyclic designs

The theory and examples which we have presented illustrate the way in which the first few graphical criteria can be used to derive a set of simple rules (Table 2) for choosing cyclic designs with two initial blocks of size two. We have worked through the examples in some detail in order to show how these rules operate in practice. Clearly, however, they can be programmed on a computer into a very simple algorithm for choosing such designs.

Moreover, these rules could be extended to the case of more than two initial blocks, and to block size greater than two. To conclude this Chapter, we describe very briefly how this could be done. That this extension is quite straightforward is a consequence of, above all, the two-stage approach to enumeration which we will continue to employ.

Any line in a general cyclic design corresponds to addition or subtraction from a given endpoint of one of the differences listed in the paired-difference set for the design (which we defined on page 90). For the case we have already dealt with - that is, where there are two initial blocks {0, a} and {0, b} - the difference set is simply {±a, ±b}(the signs depending on whether a and b are less than or greater than m). Just as the first stage of the enumeration for this special case involved allocating a and b to the labels y_0 and y_1, so, for the general case, the first stage will involve allocating differences to labels y_0, y_1, and also (where appropriate) y_2 and y_3. The second stage will then - analogously to what we have already done - invoke general formulas for the numbers of circuits arising out of each allocation. The derivation of these general formulas is rather more complicated than - but not fundamentally different from - the special case we have looked at: there is, for example, one extra type of triangle, in which the three lines each have a different label.

And, of course, once the enumeration had been generalised in this way, it would be possible to devise rules for adding or extending initial blocks in this general case too. The approach by means of graph theory which we have advocated here could, therefore, provide a general means of facilitating John's method of searching for cyclic designs.

CHAPTER 4

APPLICATION OF THE GRAPHICAL CRITERIA TO α DESIGNS

4.1 Introduction: the number of circuits of length two

The second area of application of the graphical criteria which we discuss concerns the selection of efficient α designs. Since the variety concurrence graph of an α design is, in general, more complicated than that of a cyclic design, the material in this Chapter will be correspondingly more complex and lengthy than that of Chapter 3.

Throughout, we will be concerned with α designs for $v = ks$ varieties in $b = rs$ blocks, each variety being replicated r times, and each block having k plots. There will be s blocks in each replicate.[†]

Our main aim will be to simplify the calculation of the numbers of circuits of various lengths in various types of α design; it will be by building on this that, in Chapter 5, we will be able to devise algorithms for generating efficient α designs. Recurrently here, we will use the graph theory basis of the criteria to show that they can be calculated by means of certain simple functions of the generating array α.

In this respect, indeed, we are extending a result already obtained by Patterson and Williams (1976b). They show (page 85) that the pattern of concurrences of an α design can be found from this

[†] Full details of the definition and construction of α designs are in Patterson and Williams (1976b).

generating array. As an introduction to our later enumeration of circuits of lengths three and four, we will describe first how their theory can be used to enumerate proper circuits of length two. We will start by explaining this for a simple example, and then we will deal more generally with all α designs. (Circuits of length two that are not proper are, of course, easy to enumerate: there are, as we observed on page 72, exactly vr(k-1) of them.)

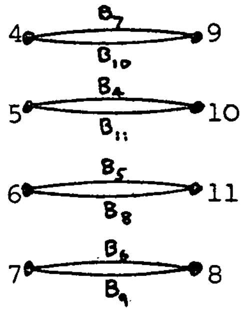
The example we look at is for 12 varieties in 3 replicates and in blocks of size 3. The generating array is:

$$\alpha = \begin{matrix} 0 & 0 & 0 \\ 0 & 1 & 2 \\ 0 & 2 & 3 \end{matrix} \quad .$$

Then the full design is (rows in each replicate being blocks):

Replicate 0	Replicate 1	Replicate 2
B_0: 0 4 8	B_4: 0 5 10	B_8 : 0 6 11
B_1: 1 5 9	B_5: 1 6 11	B_9 : 1 7 8
B_2: 2 6 10	B_6: 2 7 8	B_{10}: 2 4 9
B_3: 3 7 11	B_7: 3 4 9	B_{11}: 3 5 10

Now, in the variety concurrence graph of this design, proper circuits of length two arise as follows:



Thus, bearing in mind that starting point and orientation are significant, there are altogether 16 proper circuits of length two. What is important for our purposes here is that these circuits can be detected from the generating array α: the reason why, for example, the varieties 5 and 10 are joined by two lines - that is, concur twice - is that the corresponding entries of the array (namely those in the last two columns) determine that 5 and 10 will occur in the same row in both replicate 1 and replicate 2. It is this kind of observation that the theory of Patterson and Williams generalises.

They actually show something slightly different, but equivalent: if the generating array has entries α_{um} (for $0 \leq u \leq r-1$ and $0 \leq m \leq k-1$), then a necessary and sufficient condition for the design to have no multiple concurrences is that for $u \neq u'$ and $m \neq m'$,

$$\alpha_{mu} \dot{-} \alpha_{mu'} \neq \alpha_{m'u} \dot{-} \alpha_{m'u'} \;{}^{\dagger} . \tag{4.1}$$

This result can be, as it were, turned round to allow us to count the number of proper circuits of length two in any α design: each equation of the following form yields exactly s such circuits:

$$\alpha_{mu} \dot{-} \alpha_{mu'} = \alpha_{m'u} \dot{-} \alpha_{m'u'} \quad . \tag{4.2}$$

Moreover, every proper circuit of length two arises from one such equation, if we specify that the starting point is the one in column u and that the first line is the one corresponding to a block in replicate m: the opposite orientation is obtained by the different equation which arises by interchanging the left- and right-hand sides of (4.2), and the other starting point by multiplying both sides of (4.2) by $\dot{-}1$. Thus, in the example, there

[†] In this and all succeeding Chapters, a dot above the operator indicates modulo s.

are four different equations, as follows:

$$1 \dot{-} 2 = 2 \dot{-} 3$$

$$2 \dot{-} 3 = 1 \dot{-} 2$$

$$2 \dot{-} 1 = 3 \dot{-} 2$$

$$3 \dot{-} 2 = 2 \dot{-} 1 ,$$

and each of these gives rise to s - that is, 4 - circuits of length two. For instance, the first of them gives rise to four circuits which can be represented by the diagram on page 120, the starting point being at the left-hand end (since $m = 1$), and the starting line being the one from replicate 1 (since $u = 1$).

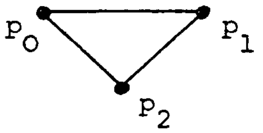
Later, in 5.2, we develop an algorithm for generating $\alpha(0,1)$ designs which relies on equation (4.2). Meanwhile, in the rest of this Chapter, we show how the number of circuits of lengths three and four (which, as in Chapter 3, we will often call, respectively, triangles and squares) can, in a similar fashion, be found from the generating array α.

4.2 The number of triangles in an α design

First of all, then, we describe how the number of triangles in an α design can be calculated very straightforwardly from the generating array α. The formulas which we will develop will be applicable to all α designs, and we will provide a fairly broad range of examples to illustrate this. However, we will also describe how the method can be simplified for $\alpha(0,1)$ designs, and for α designs in which the number of replicates is two or three. In particular, the case of three-replicate $\alpha(0,1)$ designs is so simple that we will use it as an introduction to the more general cases. Of course, the application to such designs could be deduced as a corollary of the general results (and we will also show later how this can be done). But we present this particular application first in order to make the general results more accessible: the clearest way of explicating complex theory is through preparing the way for it by means of a simpler special case.

4.2.1 Introduction: three-replicate $\alpha(0,1)$ designs

The only way a circuit of length three can arise is - as we noted in Chapter 3, page 103 - as a proper triangle. That is, none of the lines in it can be repeated. In what follows, the paradigmatic circuit of length three will be denoted by the diagram:



,

where this notation will mean that the starting point is p_0, and that the orientation is clockwise: that is, $p_0 - p_1 - p_2 - p_0$.

Now, each of these lines corresponds to a concurrence of the varieties at each end of it, and so to a block in some replicate. Obviously, the following are the only possibilities:

I all the blocks lie in the same replicate;

II the blocks lie in exactly two different replicates;

III the blocks lie in three different replicates.

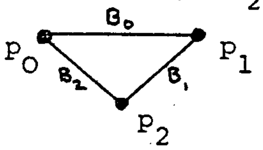
We will call these, respectively, single-replicate triangles, two-replicate triangles, and three-replicate triangles.

We show first (in (i)) that there are no two-replicate triangles in an $\alpha(0,1)$ design. Then (in (ii)), we show that the number of single-replicate triangles is a simple function of the numbers s, k, and r (which, of course, equals 3). Finally (in (iii)), we describe how three-replicate triangles can be enumerated in three-replicate $\alpha(0,1)$ designs. The arguments we use here, although shaped specifically for such designs, are typical of the arguments we will use later for the general case.

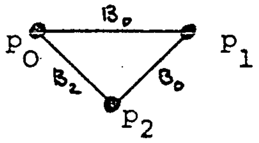
(i) Two-replicate triangles

In order to show that there are no two-replicate triangles in an $\alpha(0,1)$ design, it is sufficient to show that, if two of the replicates involved in the triangle are the same, then the three blocks are identical.

Let the line joining the varieties p_0 and p_1 correspond to concurrence in block B_0, the line joining p_1 and p_2 to concurrence in block B_1, and the line joining p_2 and p_0 to concurrence in block B_2:



Suppose, for example, that blocks B_0 and B_1 lie in the same replicate. Then they are actually the same block, since variety p_1 lies in only one block of that replicate. So the triangle looks like:



So p_0 and p_2 both occur in block B_0, and hence concur in this block. So, if B_0 and B_2 were different blocks, then p_0 and p_1 would be joined by two lines, which would contradict the fact that the design has no double concurrences.

Hence all three blocks are the same (the argument being similar if the initial two blocks which lay in the same replicate were B_0 and B_2 or B_1 and B_2 instead of B_0 and B_1).

(ii) Single-replicate triangles

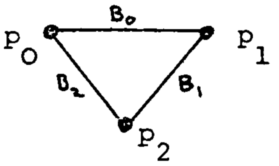
Triangles of this type arise out of three concurrences within the same block, since none of the varieties lies in more than one block of the same replicate. Each block gives rise to $k(k-1)(k-2)$ such triangles (since, of course, orientation and starting point are significant). So, since there are rs - that is, 3s - blocks altogether, the total number of such triangles is

$$3sk(k-1)(k-2), \qquad (4.3)$$

which we will call T_1.

(iii) Three-replicate triangles

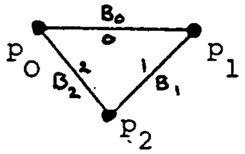
It follows from (i) and (ii) that differences in numbers of triangles among $\alpha(0,1)$ designs for three replicates must arise from differences in numbers of three-replicate triangles. In this type, all the blocks are distinct:



These three blocks lie in different replicates; let the replicate in which block B_0 lies be labelled i_0, that in which B_1 lies be labelled i_1, and that in which B_2 lies be labelled i_2. Then the enumeration of such triangles will be simplified if we divide it into two stages (reminiscent of, but not precisely analogous to, the two stages we used when we were enumerating triangles in cyclic designs (page 104)). The first stage involves allocating the replicate numbers 0, 1, and 2 in some order to the labels i_0, i_1, and i_2. There are six ways of doing this. Then the second stage involves enumerating, for each such allocation, the number of three-replicate triangles with these values of i_0, i_1, and i_2. The reason why this approach leads to a simplification is that it enables us to use general rules for enumerating three-replicate triangles for general labels i_0, i_1, and i_2. The simplification becomes even more pronounced in the case of more general α designs, as we will show later. But, even for three-replicate α(0,1) designs, it transpires that matters are made markedly more straightforward if we proceed in this way.

Indeed, this approach leads naturally to the observation that we only need enumerate triangles for one such allocation, since each of the six allocations yields the same number of three-replicate triangles. (This is a consequence of the fact that re-allocating the three replicates 0, 1, and 2 to the labels i_0, i_1, and i_2 simply corresponds to some combination of starting at a different vertex and traversing the triangle in the opposite direction. Thus, for example, the triangles arising when the allocation is $i_0 = 0$, $i_1 = 1$, and $i_2 = 2$ stand in a one-to-one relationship with the triangles when $i_0 = 1$, $i_1 = 0$, and $i_2 = 2$, the relationship being set up by the transformation of starting point from p_0 to p_2, and the transformation of direction from clockwise to anti-clockwise.) Only having to do the calculations for one of the six allocations, we choose to do it for the obvious one: $i_0 = 0$, $i_1 = 1$, and $i_2 = 2$.[†] (Then, in finding the total number of three-replicate triangles, we multiply the number for this allocation by six.)

Henceforth, therefore, we will assume that block B_u lies in replicate u (for u = 0, 1, and 2):



More precisely, let the block B_u be the x_u^{th} block in replicate u. Further, let the variety p_u lie in column j_u. (The varieties of an α design can be divided into k columns, the j^{th} column being the set {js, js+1,....., js+(s-1)} .) So the numbers j_u are distinct, since no pair of varieties in the same column can concur.

The varieties in column j of replicate u occur in the order:

$$js + (x \dotplus \alpha_{uj})$$

for $0 \leq j \leq k-1$ and $0 \leq x \leq s-1$. So the varieties in block x_v of replicate u are:

$$js + (x_v \dotplus \alpha_{uj}) \text{ for } 0 \leq j \leq k-1 \ . \qquad (4.4)$$

This, along with the triangle diagram immediately above, enables us to express each variety p_u in two different ways. For example, variety p_0 lies in block B_0; this means that p_0 lies in column j_0 of block x_0 of replicate 0, and so that:

$$p_0 = j_0 s + (x_0 \dotplus \alpha_{0j_0}) \ . \qquad (4.5)$$

Then, since variety p_1 lies in block B_0, it lies in column j_1 of block x_0 of replicate 0, and so:

[†] We also assume in this Section 4.2.1 that the array α is in standard form: that is, that $\alpha_{0j} = 0 = \alpha_{j0}$ for each j.

$$p_1 = j_1 s + (x_0 \dotplus \alpha_{0j_1}) \quad . \tag{4.6}$$

Continuing in this manner, we get a further four equations:

from block B_1:

$$p_1 = j_1 s + (x_1 \dotplus \alpha_{1j_1}) \ ; \tag{4.7}$$

$$p_2 = j_2 s + (x_1 \dotplus \alpha_{1j_2}) \ ; \tag{4.8}$$

from block B_2:

$$p_2 = j_2 s + (x_2 \dotplus \alpha_{2j_2}) \ ; \tag{4.9}$$

$$p_0 = j_0 s + (x_2 \dotplus \alpha_{2j_0}) \quad . \tag{4.10}$$

From these equations, we can derive a relationship among the α_{uj_v}. This set of six equations (4.5) - (4.10) holds if, and only if, the following set of three holds:

$$x_0 \dotplus \alpha_{0j_0} = x_2 \dotplus \alpha_{2j_0} \quad ; \tag{4.11}$$

$$x_0 \dotplus \alpha_{0j_1} = x_1 \dotplus \alpha_{1j_1} \quad ; \tag{4.12}$$

$$x_1 \dotplus \alpha_{1j_2} = x_2 \dotplus \alpha_{2j_2} \quad . \tag{4.13}$$

((4.11) is derived by equating the two expressions (4.5) and (4.10) for p_0, (4.12) by equating the two expressions for p_1, and (4.13) by equating the two expressions for p_2.)

Now, if equations (4.11) - (4.13) are valid, then, by a simple process of eliminating the numbers x_i (and bearing in mind that, since the array is in standard form, $\alpha_{0j} = 0$ for each value of j) we can derive the following equation:

$$\alpha_{1j_2} \mathbin{\dot{-}} \alpha_{1j_1} = \alpha_{2j_2} \mathbin{\dot{-}} \alpha_{2j_0} \tag{4.14}$$

(where the j_u are distinct). Moreover, conversely, each equation like (4.14) gives rise to s distinct triangles of the form depicted at the top of page 127: an equation like (4.14) specifies the columns to which the three vertices belong; there are, then, s choices for p_0, and that fixes all three vertices. That no two equations like (4.14) yield the same three-replicate triangle in this way results from the fact that the terms in the equation are fully determined by the columns to which the three ordered vertices belong. (The ordering is important, and is relevant because orientation and starting point are significant. As we mentioned above (page 126), other orientations and other starting points arise as a result of re-allocating the replicates to the labels i_0, i_1, and i_2.)

So, if there are $t_3(0, 1, 2)$[†] equations of the form (4.14), then there are $st_3(0, 1, 2)$ three-replicate triangles as in the diagram at the top of page 127, and so a total of $6st_3(0, 1, 2)$ three-replicate triangles altogether (taking into account the other possible allocations of replicates to labels).

What this means is that there are $6t_3(0, 1, 2)$ series of s triangles each (arising from the s choices of vertex p_0). These series can be grouped into $t_3(0, 1, 2)$ sets of six, the series in each set of six being obtained from each other by the various choices of starting point and orientation. (We will illustrate these groupings below in the Example (1).)

To facilitate the counting of equations like (4.14), we define two k×k arrays G_1 and G_2, whose $(i,j)^{th}$ entries are, respectively, $\alpha_{1i} \dot{-} \alpha_{1j}$ and $\alpha_{2i} \dot{-} \alpha_{2j}$. In enumerating such equations, the

[†] The notation $t_3(0, 1, 2)$ will be explained later when we deal with general α designs. It indicates, essentially, that the allocation referred to has $i_0 = 0$, $i_1 = 1$, and $i_2 = 2$.

condition that the suffices j_0, j_1, and j_2 be distinct is equivalent to the condition that some pair of them be distinct (since if any two of them are equal, then condition (4.1) ensures that all three of them will be equal). Hence, for any given value of j_2, the number of equations of the form (4.14) with that particular value of j_2 is the number of equalities between an off-diagonal element of row j_2 of the array G_1 and an off-diagonal element of row j_2 of the array G_2.†

Some further notation will simplify the description of the formula for enumerating these equalities. For each row j ($0 \leq j \leq k-1$), and for each x in {0, 1,....., s-1} , let $h_{1j}(x)$ be the number of occurrences of x in the row j of G_1, and let $h_{2j}(x)$ be the number of occurrences of x in row j of G_2. Then the number of equalities between an off-diagonal element of row j of G_1 and an off-diagonal element of row j of G_2 is:

$$\sum_{x=1}^{s-1} h_{1j}(x)\ h_{2j}(x).$$

(The summation starts at 1 since zeroes occur only on the diagonals.) So, summing over j, the total number of equalities is:

$$t_3(0,\ 1,\ 2) \quad = \quad \sum_{j=0}^{k-1} \sum_{x=1}^{s-1} h_{1j}(x)\ h_{2j}(x) \ . \qquad (4.15)$$

Bringing together this equation with the comments concerning the number of three-replicate triangles that we made on page 129,

†The forms of these arrays G_1 and G_2 bear some resemblance to the structure of what are known in combinatorial theory as perfect difference sets. We investigate this similarity further in the final Section 4.4.3 of this Chapter on α designs.

we find that the total number of these triangles is:

$$6s \sum_{j=0}^{k-1} \sum_{x=1}^{s-1} h_{1j}(x)\, h_{2j}(x) \ , \tag{4.16}$$

which we will call T_3.

Finally, recalling that the number of single-replicate triangles is (by equation (4.3)) 3sk(k - 1)(k - 2), and that for an α(0,1) design there are no two-replicate triangles, we have that for an α(0,1) design in three replicates, the total number of triangles is:

$$\underline{\underline{C}}_3 = 3sk(k-1)(k-2) + 6s \sum_{j=0}^{k-1} \sum_{x=1}^{s-1} h_{1j}(x)\, h_{2j}(x) \ . \tag{4.17}$$

(In Chapter 5, we describe how this formula can be used as the basis of an algorithm for constructing efficient three-replicate α(0,1) designs.)

We now turn to specific examples of these calculations. The first example illustrates how this theory can be used to enumerate circuits of length three, and the second how the number of three-replicate triangles can be used to compare three-replicate α(0,1) designs for fixed values of s and k. (Thus the first example is of purely mathematical interest; the second has a bearing on practical statistical applications.)

Example (1)

For s = 8 and k = 4, the best three-replicate α(0,1) design has generating array:

$$\alpha = \begin{matrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 5 \\ 0 & 3 & 7 & 1 \end{matrix} \ .$$

The arrays G_1 and G_2 are then:

G_1	0	1	2	5
0	0	7	6	3
1	1	0	7	4
2	2	1	0	5
5	5	4	3	0

G_2	0	3	7	1
0	0	5	1	7
3	3	0	4	2
7	7	4	0	6
1	1	6	2	0

Then, by formula (4.15), $t_3(0, 1, 2) = 2$, and so the number of three-replicate triangles is (by equation (4.16)) 96.

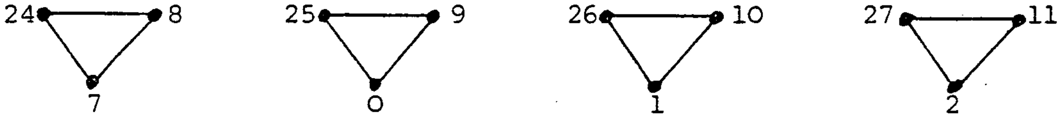
There are, by equation (4.3), $3sk(k - 1)(k - 2) = 576$ single-replicate triangles.

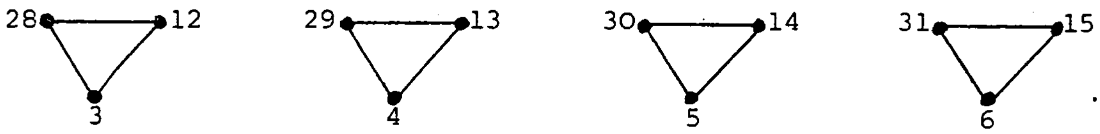
The three-replicate triangles can be listed by examining the design, which is in full:

Replicate 0				Replicate 1				Replicate 2			
0	8	16	24	0	9	18	29	0	11	23	25
1	9	17	25	1	10	19	30	1	12	16	26
2	10	18	26	2	11	20	31	2	13	17	27
3	11	19	27	3	12	21	24	3	14	18	28
4	12	20	28	4	13	22	25	4	15	19	29
5	13	21	29	5	14	23	26	5	8	20	30
6	14	22	30	6	15	16	27	6	9	21	31
7	15	23	31	7	8	17	28	7	10	22	24

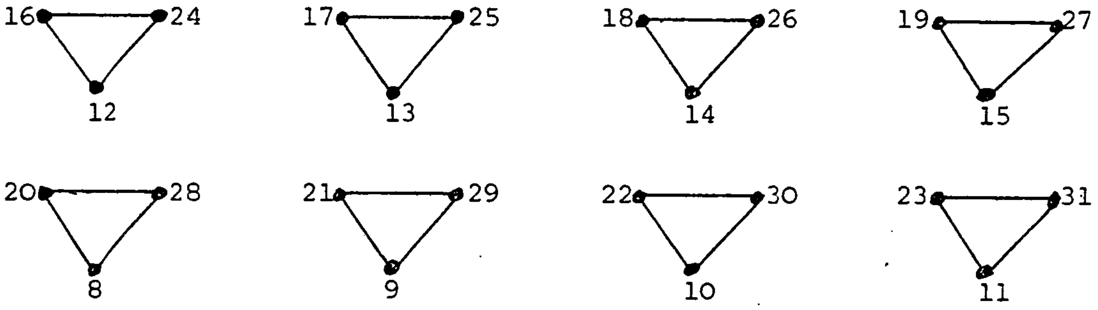
As we explained earlier, there are $t_3(0, 1, 2)$ series of s triangles each, the series arising out of $t_3(0, 1, 2)$ - that is, 2 - underlying triangle diagrams by means of different starting points and orientations. The two underlying series are as follows:

(1) $j_0 = 3$; $j_1 = 1$; $j_2 = 0$.





(2) $j_0 = 2;\ j_1 = 3;\ j_2 = 1.$



Each of these sixteen diagrams represents six circuits of length three when different starting points and orientations are taken into account. Notice that successive terms among the eight in a series are obtained by repeatedly adding 1 (modulo 8) to the vertices.

Example (2)

We consider all $\alpha(0,1)$ designs for three replicates in blocks of size 4, with 6 blocks in each replicate, which have the following as the first two rows in their generating arrays:

$$\begin{matrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 3 \end{matrix}.$$

Then each of the possible extensions of these two rows to an $\alpha(0,1)$ design in three replicates has $3sk(k-1)(k-2) = 432$ single-replicate triangles. The numbers of three-replicate triangles, along with the corresponding values of the harmonic mean efficiency factor $\underline{\underline{A}}$, are given in Table 3 (page 134). Notice that the more three-replicate triangles there are, the lower is the value of the harmonic mean efficiency factor $\underline{\underline{A}}$ - though, as we might expect from equation (2.5), the ordering induced by $\underline{\underline{A}}$ is more subtle than that induced by the numbers of three-replicate triangles: the number of such triangles is 252 for both the second and third groups in the Table. The next graphical criterion - $\underline{\underline{C}}_4$, or numbers of squares - serves to distinguish between these two groups: those designs in the first group (with harmonic mean efficiency factor .7239) have 7824 squares, while those in the second group (with harmonic mean efficiency factor .7229) have 7920 squares. Observations such as these tend to lend credibility to the conjecture in 2.5.

Table 3

Three-replicate α designs for 24 varieties in blocks of size 4.

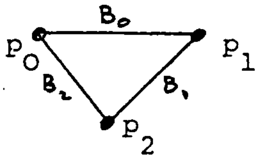
Comparison of harmonic mean efficiency factor with number of three-replicate triangles.

Third row of		Number of
generating array α.	$\underline{\underline{A}}$	three-replicate triangles.
0 5 3 2	.7265	216
0 4 1 5	.7265	216
0 3 1 4	.7265	216
0 2 5 1	.7265	216
0 5 4 2	.7239	252
0 4 3 2	.7239	252
0 3 5 1	.7239	252
0 2 4 1	.7239	252
0 5 1 4	.7229	252
0 4 3 5	.7229	252
0 3 5 4	.7229	252
0 2 1 5	.7229	252
0 4 3 1	.7196	288
0 3 5 2	.7196	288

4.2.2 Generalisation to all α designs

The generalisation to all α designs of these results on enumerating triangles is now a fairly straightforward process, and involves no ideas that are fundamentally different from those we have encountered hitherto.

As in the case of three-replicate α(0,1) designs, we deal with the typical triangle:



(starting at p_0 and traversing the lines clockwise). Again, each of the lines corresponds to a concurrence in a block in some replicate, and, as before, we have the following types:

I single-replicate triangles: all the blocks are in the same replicate;

II two-replicate triangles: the blocks lie in exactly two different replicates;

III three-replicate triangles: the blocks lie in three different replicates.

We deal with each type separately, and bring together on page 150 a summary of the formulas which we derive.

I. Single-replicate triangles

The number of single-replicate triangles is

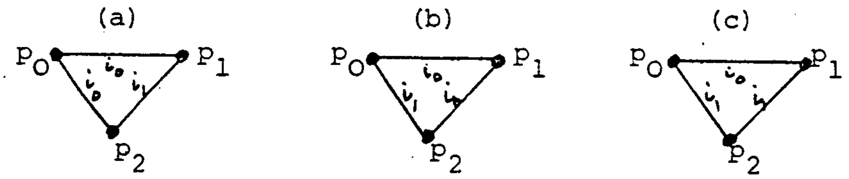
$$rsk(k - 1)(k - 2) , \qquad (4.18)$$

which we will denote again by T_1. (This can be established by arguing exactly as in (ii) on page 125: the reasoning there generalises in an obvious way to all α designs.)

II. Two-replicate triangles

Although we have to deal in some more detail with two-replicate and three-replicate triangles, it will nevertheless be apparent that the techniques we used for three-replicate $\alpha(0,1)$ designs are essentially all that are needed here too. Moreover, it will also become clear that the enumeration of triangles of these types can be based on arrays like G_1 and G_2.

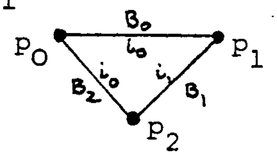
The two replicates involved in a two-replicate triangle we will label as i_0 and i_1 - and we will adopt the convention that the first replicate to occur is labelled i_0. Then any triangle of this type is of one of the following three forms:



As in the case of three-replicate triangles in three-replicate $\alpha(0,1)$ designs, the enumeration of two-replicate triangles here will be facilitated if we divide it into two stages. The first stage involves allocating two of the replicate numbers 0, 1,....., r-1 to the labels i_0 and i_1. (There are $r(r-1)$ ways of doing this.) Then the second stage involves enumerating, for each such allocation, the number of triangles of each of the three forms (a), (b), and (c) having these particular values of i_0 and i_1.

We show that the number of two-replicate triangles for any given pair of replicates i_0 and i_1 can be calculated from the corresponding two rows of the generating array α. To simplify this further, we stipulate that after selecting a pair of replicates i_0 and i_1, we will reduce the row corresponding to i_0 to zeroes by adding constants to columns of the array α. The effect of this operation is, simply, to re-label the varieties, and so the number of two-replicate triangles arising from the new rows i_0 and i_1 is exactly the same as the number arising from the old ones. (Re-labelling the vertices of a graph G transforms it into a graph G' that is isomorphic to G: that is, which has the same structural properties as G.) In other words, instead of assuming (as we have done up till now) that the array α is in standard form, we will assume here that $\alpha_{i_0 j} = 0$ for each j.[†]

First, consider two-replicate triangles of the form (a) (page 136). Let the block which corresponds to the line $p_0 - p_1$ be B_0, the block corresponding to the line $p_1 - p_2$ be B_1, and the block corresponding to the line $p_2 - p_0$ be B_2. So B_0 and B_2 are in replicate i_0, and B_1 is in replicate i_1, as follows:



For each u = 0, 1, and 2, let B_u be the x_u^{th} block in its replicate. Also, let the variety p_u lie in column j_u. So, arguing exactly as

[†] Implicitly postulated here is the practice of using the same notation for all the forms of the original array α which can be obtained by reducing some row to zeroes. This should not lead to any confusion provided it is always remembered that, having chosen any two replicates i_0 and i_1, the first thing we do is reduce the row corresponding to i_0 to zeroes. The alternative practice would have been to have added another suffix to the α_{ij} to indicate which of the r rows is zero; that, however, would have been awkward and excessively complicated

we did for three-replicate triangles in three-replicate $\alpha(0,1)$ designs, variety p_0 lies in column j_0, and in block x_0 of replicate i_0; so:

$$p_0 \;=\; j_0 s + (x_0 \dotplus \alpha_{i_0 j_0}) \quad . \tag{4.19}$$

Similarly, from block B_0:

$$p_1 \;=\; j_1 s + (x_0 \dotplus \alpha_{i_0 j_1}) \quad ; \tag{4.20}$$

from block B_1:

$$p_1 \;=\; j_1 s + (x_1 \dotplus \alpha_{i_1 j_1}) \quad ; \tag{4.21}$$

$$p_2 \;=\; j_2 s + (x_1 \dotplus \alpha_{i_1 j_2}) \quad ; \tag{4.22}$$

from block B_2:

$$p_2 \;=\; j_2 s + (x_2 \dotplus \alpha_{i_0 j_2}) \quad ; \tag{4.23}$$

$$p_0 \;=\; j_0 s + (x_2 \dotplus \alpha_{i_0 j_0}) \quad . \tag{4.24}$$

Then, proceeding in the same way as earlier, we can derive from these equations a relationship among the $\alpha_{i_u j_w}$. This set of six holds if, and only if, the following set of three holds:

$$x_2 \dotplus \alpha_{i_0 j_0} \;=\; x_0 \dotplus \alpha_{i_0 j_0} \quad ; \tag{4.25}$$

$$x_0 \dotplus \alpha_{i_0 j_1} \;=\; x_1 \dotplus \alpha_{i_1 j_1} \quad ; \tag{4.26}$$

$$x_1 \dotplus \alpha_{i_1 j_2} \;=\; x_2 \dotplus \alpha_{i_0 j_2} \quad . \tag{4.27}$$

((4.25) is obtained by equating the two expressions (4.19) and (4.24) for p_0, and so on.)

Now we use the fact that $\alpha_{i_0 j} = 0$ for each j. We get:

$$x_2 \;=\; x_0 \quad ;$$

(that is, blocks B_0 and B_1 are the same);

$$x_0 \dot{-} x_1 = \alpha_{i_1 j_1} ;$$

$$x_2 \dot{-} x_1 = \alpha_{i_1 j_2} .$$

Hence: $$\alpha_{i_1 j_1} \dot{-} \alpha_{i_1 j_2} = 0 . \qquad (4.28)$$

Let there be $t_2(i_0, i_1)$ equations of this form (enumerating over ordered pairs of distinct columns j_1 and j_2). The suffices i_0 and i_1 here indicate which two of the rows of the array α are involved, and what is the order of their occurrence.

It now follows that there are $s(k - 2)t_2(i_0, i_1)$ two-replicate triangles for this choice of two replicates i_0 and i_1. Demonstrating this is very similar to the analogous proof for three-replicate triangles in three-replicate $\alpha(0,1)$ designs. An equation like (4.28) specifies the columns to which the varieties p_1 and p_2 belong. There are, then, s choices for p_1. (That is, s choices for the number x_1.) This choice fixes the value of p_2. Further, having chosen p_1, there are $k - 2$ choices available for p_0 in the block $B_0 = B_2$. So an equation like (4.28) yields $s(k - 2)$ two-replicate triangles. Moreover, no two equations like (4.28) yield the same two-replicate triangle (since the terms in equation (4.28) are fully determined by the columns to which the second and third points of the triangle belong.) Hence, indeed, the number of two-replicate triangles for this choice of two replicates i_0 and i_1 is:

$$s(k - 2)t_2(i_0, i_1) . \qquad (4.29)$$

Two-replicate triangles of the forms (b) and (c) on page 136 can be treated very similarly, and it turns out that the number of each of them is, again, $s(k - 2)t_2(i_0, i_1)$. (For (b), we derive, in place of the equation (4.28), the equation:

$$\alpha_{i_1 j_0} \doteq \alpha_{i_1 j_2} = 0 ;$$

for (c) we derive: $\alpha_{i_1 j_0} \doteq \alpha_{i_1 j_1} = 0$.)

So, bringing together the results for two-replicate triangles of the forms (a), (b), and (c), we have that the number of two-replicate triangles for a particular choice of two replicates i_0 and i_1 is:

$$3s(k - 2)t_2(i_0, i_1) . \qquad (4.30)$$

So the total number of two-replicate triangles (over all choices of two replicates) is:

$$3s(k - 2) \sum_{i_0 \neq i_1} t_2(i_0, i_1) . \qquad (4.31)$$

There are several simplifications which we can introduce to the calculation of the quantity (4.31). The most useful stems from the observation that the quantities t_2 are symmetrical in i_0 and i_1: to be precise, $t_2(i_0, i_1) = t_2(i_1, i_0)$. This is due to the fact that if row i_1 is reduced to zeroes instead of row i_0, then row i_0 gets filled with the negatives of the former entries of row i_1. It follows from this and the expression (4.31) that the number of two-replicate triangles is:

$$6s(k - 2) \sum_{i_0 < i_1} t_2(i_0, i_1) . \qquad (4.32)$$

This we will denote by T_2.

In practice, the calculation of these quantities $t_2(i_0, i_1)$ can be simplified by means of arrays analogous to the arrays G_1 and

G_2 which we used for three-replicate $\alpha(0,1)$ designs. Define the $k \times k$ array $G_{i_0 i_1}$ to have as its $(j_1, j_2)^{th}$ entry the quantity:

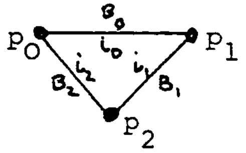
$$\alpha_{i_1 j_1} \dot{-} \alpha_{i_1 j_2} .$$

(The suffices i_0 and i_1 in $G_{i_0 i_1}$ refer to the two rows of the array α which are involved, the first of these rows being, as always, reduced to zeroes. Notice that with this notation, the two arrays G_1 and G_2 defined for three-replicate $\alpha(0,1)$ designs would be G_{01} and G_{02}.) Then $t_2(i_0, i_1)$, which is the number of equalities of the form $\alpha_{i_1 j_1} \dot{-} \alpha_{i_1 j_2} = 0$ $(j_1 \neq j_2)$ is, simply, the number of zeroes among the off-diagonal entries of the array $G_{i_0 i_1}$.

III. Three-replicate triangles

The enumeration of three-replicate triangles for general α designs is similar to the enumeration of such triangles in three-replicate $\alpha(0,1)$ designs.

Let the three replicates in the triangle be labelled i_0, i_1, and i_2, occurring in that order. So, with our customary notation, we have the typical configuration:



Again, we divide the enumeration into two stages. In the first stage, we allocate three of the replicate numbers 0, 1,....., r-1 to the labels i_0, i_1, i_2. There are $r(r - 1)(r - 2)$ ways of doing this. Then in the second stage we enumerate, for each such allocation, the number of three-replicate triangles having these particular values of i_0, i_1, and i_2. We show that this second stage can be based on the corresponding rows of the generating array α, with the usual simplification that the row corresponding to the replicate i_0 has been reduced to zeroes. (As in the case of two-replicate triangles, this operation does not alter the essential structure of the variety concurrence graph.)

Arguing as before, we derive from the diagram on page 141 the following set of three equations (analogous to (4.11) - (4.13), or to (4.25) - (4.27)):

$$\begin{aligned} &\text{from variety } p_0\text{:} \quad x_2 \dotplus \alpha_{i_2 j_0} = x_0 \dotplus \alpha_{i_0 j_0} \; ; \\ &\text{from variety } p_1\text{:} \quad x_0 \dotplus \alpha_{i_0 j_1} = x_1 \dotplus \alpha_{i_1 j_1} \; ; \\ &\text{from variety } p_2\text{:} \quad x_1 \dotplus \alpha_{i_1 j_2} = x_2 \dotplus \alpha_{i_2 j_2} \; . \end{aligned}$$

Then, since $\alpha_{i_0 j} = 0$ for each j, we have:

$$\begin{aligned} x_0 \mathbin{\dot{-}} x_2 &= \alpha_{i_2 j_0} \; ; \\ x_0 \mathbin{\dot{-}} x_1 &= \alpha_{i_1 j_1} \; ; \\ x_1 \mathbin{\dot{-}} x_2 &= \alpha_{i_2 j_2} \mathbin{\dot{-}} \alpha_{i_1 j_2} \; . \end{aligned}$$

It follows from this that:

$$\alpha_{i_1 j_2} \mathbin{\dot{-}} \alpha_{i_1 j_1} = \alpha_{i_2 j_2} \mathbin{\dot{-}} \alpha_{i_2 j_0} \tag{4.33}$$

(where the suffices j_0, j_1, and j_2 are distinct).

Let there are be $t_3(i_0, i_1, i_2)$ equations of this form (counting over ordered triples of distinct columns j_0, j_1, and j_2). Again, the suffices i_0, i_1, and i_2 indicate which rows of the array α are involved, and what is the order of their occurrence. (This accounts for the notation which we used for three-replicate $\alpha(0,1)$

designs: see the footnote on page 129.)

We show now that there are, then, $st_3(i_0, i_1, i_2)$ three-replicate triangles for this choice of replicates i_0, i_1, and i_2. The argument is, once more, similar to the analogous proof for three-replicate triangles in three-replicate $\alpha(0,1)$ designs. An equation like (4.33) specifies the columns to which the vertices belong. Then there are s choices for p_0, and that fixes all three vertices. No two equations like (4.33) give the same three-replicate triangle, since (as with equation (4.14) before) the terms of equation (4.33) are fully determined by the columns which contain the three vertices of the triangle. (The starting vertex gives the value of j_0, the second vertex in the orientation gives the value of j_1, and the third vertex gives the value of j_2.)

So the total number of three-replicate triangles is:

$$s \Sigma\ t_3(i_0, i_1, i_2) \quad , \tag{4.34}$$

where this sum is over sets of distinct replicates i_0, i_1, and i_2.

The calculation of this quantity can be made considerably easier by means of several different simplifications, all analogous to simplifications we have used previously. On the one hand, we will show that the range of the summation can be restricted to replicates with $i_0 < i_1 < i_2$. On the other hand, we will derive expressions (equation (4.36) below) for the quantities $t_3(i_0, i_1, i_2)$ which will be based entirely on the arrays G_{uw} which we introduced above in the context of enumerating two-replicate triangles.

First, then, we simplify the summation: we show that the number of three-replicate triangles is:

$$6s \sum_{i_0<i_1<i_2} t_3(i_0, i_1, i_2) \quad . \qquad (4.35)$$

(Compare equation (4.16) or equation (4.32).)

Consider any three replicates u, w, and z. Then there are, in principle, six ways of allocating these three to the labels i_0, i_1, and i_2 in the first stage of enumerating the three-replicate triangles. But, in fact, each of the six allocations yields the same number of triangles. This is because (as with the analogous situation for three-replicate $\alpha(0,1)$ designs), re-allocating these replicates u, w, and z to labels simply corresponds to some combination of starting at a different vertex and traversing the triangle in the opposite direction. Thus, for instance, the triangles arising when the allocation is $i_0 = u$, $i_1 = w$, and $i_2 = z$ stand in a one-to-one relationship with the triangles when $i_0 = w$, $i_1 = u$, and $i_2 = z$, the relationship being set up by the rule that the starting point is p_2 (instead of p_0), and the direction is anticlockwise (instead of clockwise).

It follows that the expression (4.35) does, indeed, count all the three-replicate triangles. We therefore denote it by T_3.

The other important simplification that we can introduce into the enumeration of three-replicate triangles concerns the way in which we calculate the quantity $t_3(i_0, i_1, i_2)$. We will derive a formula for this which will be based on the arrays $G_{i_0i_1}$, $G_{i_0i_2}$, and $G_{i_1i_2}$, and which will be seen to be a generalisation of the formula (4.15) which we developed for use in enumerating three-replicate triangles in three-replicate $\alpha(0,1)$ designs.

In order to present this formula, we have to introduce some new notation. We have already defined the arrays G_{uw}. Now we define quantities that will record the numbers of occurrences of each of the residues 0, 1,....., s-1 in the different rows of these arrays. For each such array G_{uw} (where $0 \leq u < w \leq r-1$), let $h_{uwj}(x)$ denote the number of occurrences of x in the row j of G_{uw} (where $0 \leq x \leq s-1$, and $0 \leq j \leq k-1$). (Thus, just as the arrays G_1 and G_2 for three-replicate $\alpha(0,1)$ designs became in this notation G_{01} and G_{02}, so the quantities $h_{1j}(x)$ and $h_{2j}(x)$ have become $h_{01j}(x)$ and $h_{02j}(x)$ respectively.)

Then the formula which we will establish for $t_3(i_0, i_1, i_2)$ is:

$$t_3(i_0, i_1, i_2) = \sum_{j=0}^{k-1} \sum_{x=0}^{s-1} h_{i_0 i_1 j}(x) h_{i_0 i_2 j}(x) - t_2(i_0,i_1) - t_2(i_0,i_2) - t_2(i_1,i_2) - k \tag{4.36}$$

(We will show later that this is a generalisation of the expression (4.15) for $t_3(0, 1, 2)$ in three-replicate $\alpha(0,1)$ designs.)

Now, $t_3(i_0, i_1, i_2)$ is the number of equalities of the form (4.33), which is the number of equalities between an off-diagonal antry of the array $G_{i_0 i_1}$ and an off-diagonal entry of the array $G_{i_0 i_2}$ such that these entries are to be in the same row but not in the same column. So far, this is close to the observations we made at a similar stage of the development for three-replicate $\alpha(0,1)$ designs (page 130). However, there now arises a new complication. In the case of three-replicate $\alpha(0,1)$ designs, we were able simply to count the numbers of appearances of each of the quantities x in the rows of the two arrays, and then multiply them together. This we could do because of the absence of multiple lines, which ensured that there would not be equality between entries in the same place in both arrays. (We show below that this simplification applies to all α designs in which there are no multiple lines.) However, for the general α designs we are dealing with here, we have to be more careful: there can be equalities between entries in the same place in the two arrays. We get round these difficulties in the proof of equation (4.36) by means of the following quantities $y_{uwj_0j_1}$, which act as indicator variables for the cells of the array G_{uw}: define $y_{uwj_0j_1}(x)$ to be 1 if $G_{uw}(j_0,j_1) = x$, and to be 0 if it does not. Then the number of equalities of the form (4.33) which arise from a particular entry $G_{i_0i_1}(j_2,j_1)$ $(j_2 \neq j_1)$ is:

$$\sum_{x=0}^{s-1} y_{i_0i_1j_2j_1}(x) \sum_{\substack{j_0=0\\ j_0\neq j_1,j_2}}^{k-1} y_{i_0i_2j_2j_0}(x) \quad .$$

($y_{i_0i_1j_2j_1}(x)$ is non-zero for exactly one value of x, when it is 1, and then the inner sum records the number of occurrences of that value of x in the row j_2 of $G_{i_0i_2}$, off-diagonal (since $j_0 \neq j_2$) and not in the cell $(j_2,\ j_1)$ (since $j_0 \neq j_1$).)

For convenience, we label this quantity $X_{j_2j_1}$. Then, since $j_2 \neq j_1$, we have that $X_{j_2j_1}$ equals:

$$\sum_{x=0}^{s-1} y_{i_0i_1j_2j_1}(x) \sum_{j_0=0}^{k-1} y_{i_0i_2j_2j_0}(x) - \sum_{x=0}^{s-1} y_{i_0i_1j_2j_1}(x) y_{i_0i_2j_2j_1}(x) - \sum_{x=0}^{s-1} y_{i_0i_1j_2j_1}(x) y_{i_0i_2j_2j_2}(x) \ . \tag{4.37}$$

Now, the sum

$$\sum_{j_0=0}^{k-1} y_{i_0 i_2 j_2 j_0}(x)$$

is, simply, the number of occurrences of x in the row j_2 of $G_{i_0 i_2}$, which we have previously defined as $h_{i_0 i_2 j_2}(x)$. (See page 145.) We can substitute this in the first term in the expression (4.37). Also, we can simplify the third term by using the fact that $y_{i_0 i_2 j_2 j_2}(x)$ is 1 for $x = 0$ and is 0 for $x \geq 1$. So we get the following simplified version of the expression (4.37) for $X_{j_2 j_1}$:

$$\sum_{x=0}^{s-1} y_{i_0 i_1 j_2 j_1}(x) h_{i_0 i_2 j_2}(x) - \sum_{x=0}^{s-1} y_{i_0 i_1 j_2 j_1}(x) y_{i_0 i_2 j_2 j_1}(x)$$
$$- y_{i_0 i_1 j_2 j_1}(0).$$

So, summing over the whole range of values $j_1 \neq j_2$, we find that the number of equalities of the form (4.33) which arise out of the row j_2 of $G_{i_0 i_1}$ is:

$$\sum_{\substack{j_1=0 \\ j_1 \neq j_2}}^{k-1} X_{j_2 j_1},$$

which equals:

$$\sum_{x=0}^{s-1} \sum_{\substack{j_1=0 \\ j_1 \neq j_2}}^{k-1} y_{i_0 i_1 j_2 j_1}(x) h_{i_0 i_2 j_2}(x) - \sum_{x=0}^{s-1} \sum_{\substack{j_1=0 \\ j_1 \neq j_2}}^{k-1} y_{i_0 i_1 j_2 j_1}(x) y_{i_0 i_2 j_2 j_1}(x)$$
$$- \sum_{\substack{j_1=0 \\ j_1 \neq j_2}}^{k-1} y_{i_0 i_1 j_2 j_1}(0).$$

Simplifying this by the same techniques as we used immediately above on the expression (4.37), and then summing over j_2, we get that the total number of equalities of the form (4.33) - in other words, $t_3(i_0, i_1, i_2)$ - is:

$$\sum_{j_2=0}^{k-1} \sum_{x=0}^{s-1} h_{i_0 i_1 j_2}(x) h_{i_0 i_2 j_2}(x) - \sum_{j_2=0}^{k-1} h_{i_0 i_1 j_2}(0) - \sum_{j_2=0}^{k-1} h_{i_0 i_2 j_2}(0) + 2k - \sum_{x=0}^{s-1} \sum_{j_2, j_1=0}^{k-1} y_{i_0 i_1 j_2 j_1}(x) y_{i_0 i_2 j_2 j_1}(x) . \qquad (4.38)$$

Now, as we pointed out earlier when we were dealing with two-replicate triangles, $t_2(i_0, i_1)$ is the number of zeroes among the off-diagonal entries of the array $G_{i_0 i_1}$. On the other hand, the sum:

$$\sum_{j_2=0}^{k-1} h_{i_0 i_1 j_2}(0)$$

is the total number of zeroes in the array $G_{i_0 i_1}$, including the k zeroes which inevitably appear on the leading diagonal. Hence:

$$\sum_{j_2=0}^{k-1} h_{i_0 i_1 j_2}(0) - k = t_2(i_0, i_1) . \qquad (4.39)$$

Similarly, we get that:

$$\sum_{j_2=0}^{k-1} h_{i_0 i_2 j_2}(0) - k = t_2(i_0, i_2) . \qquad (4.40)$$

Furthermore, the final expression in (4.38) - namely:

$$\sum_{x=0}^{s-1} \sum_{j_2, j_1=0}^{k-1} y_{i_0 i_1 j_2 j_1}(x) y_{i_0 i_2 j_2 j_1}(x)$$

- is the number of equalities between entries $G_{i_0 i_1}(j_2, j_1)$ and $G_{i_0 i_2}(j_2, j_1)$. (That is, between cells in the same position in the two arrays.) It is not difficult to see that such equalities identify the positions of the off-diagonal zeroes in the array $G_{i_1i_2}$. Indeed, it is almost obvious. Let α denote the generating array with its row i_0 reduced to zeroes, and let α' denote it with row i_1 reduced to zeroes. That is, for each i and j:

$$\alpha'_{ij} = \alpha_{ij} \dot{-} \alpha_{i_1j} .$$

Then equality between the $(j_2, j_1)^{th}$ entries of $G_{i_0i_1}$ and $G_{i_0i_2}$ means that:

$$\alpha_{i_1j_2} \dot{-} \alpha_{i_1j_1} = \alpha_{i_2j_2} \dot{-} \alpha_{i_2j_1} .$$

So:

$$(\alpha_{i_1j_2} \dot{-} \alpha_{i_1j_2}) \dot{-} (\alpha_{i_1j_1} \dot{-} \alpha_{i_1j_1}) = (\alpha_{i_2j_2} \dot{-} \alpha_{i_1j_2}) \dot{-} (\alpha_{i_2j_1} \dot{-} \alpha_{i_1j_1})$$

which implies that:

$$0 = \alpha'_{i_2j_2} \dot{-} \alpha'_{i_2j_1} .$$

That is, $0 = G_{i_1i_2}(j_2, j_1)$, or, in other words, the $(j_2, j_1)^{th}$ entry of the array $G_{i_1i_2}$ is zero. This argument can be reversed to show that each off-diagonal zero in the array $G_{i_1i_2}$ corresponds to an equality between entries in the same position in arrays $G_{i_0i_1}$ and $G_{i_0i_2}$.

Hence, again allowing for the fact that $G_{i_0i_1}$ and $G_{i_0i_2}$ both have k zeroes down their leading diagonals, we find that we can simplify the final expression in (4.38) as follows:

$$\sum_{x=0}^{s-1} \sum_{j_2,j_1=0}^{k-1} y_{i_0i_1j_2j_1}(x) y_{i_0i_2j_2j_1}(x) = t_2(i_1, i_2) + k . \qquad (4.41)$$

Substituting in (4.38) from (4.39), (4.40), and (4.41) establishes the validity of the equation (4.36).

This completes the simplifications that are possible in enumerating three-replicate triangles.

Summary of formulas for enumerating triangles in α designs

Before we describe how all this theory can be used in practice, we bring together the important definitions and results in the following summary. The summary contains nothing that we have not already presented; it simply collates the material in a form that is more concise, and therefore can be more easily referred to.

For $0 \leq u < w \leq r-1$, we define the $k \times k$ array G_{uw} by: first, reduce the array α so that the row corresponding to replicate u consists entirely of zeroes; then, with α in this form, let the (j_0, j_1) entry of G_{uw} be $\alpha_{wj_0} \dot{-} \alpha_{wj_1}$ (for $0 \leq j_0, j_1 \leq k-1$). Further, let $t_2(u,w)$ be the number of zeroes among the off-diagonal entries of G_{uw}, let $h_{uwj}(x)$ be the number of occurrences of x in the row j of G_{uw} ($0 \leq x \leq s-1$ and $0 \leq j \leq k-1$), and then define $t_3(i_0, i_1, i_2)$ by the equation (4.36).

Then we have the following formulas for the number of triangles of the three different types.

I. Single-replicate triangles

$$T_1 = rsk(k-1)(k-2). \quad \text{(See equation (4.18).)}$$

II. Two-replicate triangles

$$T_2 = 6s(k-2) \sum_{i_0<i_1} t_2(i_0, i_1). \quad \text{(See equation (4.32).)}$$

III. Three-replicate triangles

$$T_3 = 6s \sum_{i_0<i_1<i_2} t_3(i_0, i_1, i_2). \quad \text{(See equation (4.35).)}$$

Notice that each of these formulas contains a factor s. This reflects the fact that we have been able to base the enumeration on the array α, or, put differently, that the triangles occur in series of length s, the members of each series being generated by repeatedly adding 1 modulo s.

It remains to mention some aspects of the practical applications of these formulas. We have already described rules for enumerating triangles in three-replicate $\alpha(0,1)$ designs, and we will indicate here how these rules arise as particular cases of the general formulas given in the above summary. Also worthy of comment are the simplifications that are possible when the formulas are applied to $\alpha(0,1)$ designs in general, and to α designs in two replicates, and so we will look at these cases as well. Subsequently, we will provide some more examples.

A natural way of looking for simplifications of the formulas in the summary is to investigate designs in which some or other of the two-replicate and three-replicate triangles are absent altogether (the number of single-replicate triangles being fixed for given values of r, s, and k). Obviously, on the one hand, when there are only two replicates, then there are no three-replicate triangles, for an indispensible first requirement for the existence of such triangles is the existence of three distinct replicates.

If, on the other hand, there are no multiple lines, then there are no two-replicate triangles. For, in this case, there are no equations of the form (4.28). (Any such equation produces double lines joining varieties in column j_1 to varieties in column j_2, one line corresponding to a block in replicate i_0, and one to a block in replicate i_1.) So, for $\alpha(0,1)$ designs, $t_2(i_0, i_1)$ (which records the number of such equations) is zero for every pair of replicates i_0 and i_1. So, in such designs, there are no two-replicate triangles. (One consequence of this is that, in comparing two $\alpha(0,1)$ designs D and D', we should compare the numbers of three-replicate triangles, since the number of single-replicate triangles is the same for both (being $rsk(k - 1)(k - 2)$). If the designs have T_3 and T_3' such triangles respectively, then D is better than D' on the graphical criterion $\underline{C}_3$ if $T_3 < T_3'$.)

The enumeration of three-replicate triangles in $\alpha(0,1)$ designs can be simplified too. The quantities $t_3(i_0, i_1, i_2)$ which are required for this enumeration are given by equation (4.36). Now, as we have just shown, $t_2(u, w) = 0$ for any two replicates u and w in a designs with no multiple lines. Hence, for such a design, equation (4.36) becomes:

$$t_3(i_0, i_1, i_2) = \sum_{j=0}^{k-1} \sum_{x=0}^{s-1} h_{i_0 i_1 j}(x)\, h_{i_0 i_2 j}(x) - k . \qquad (4.42)$$

Furthermore, the absence of multiple lines means that the only zeroes in the arrays $G_{i_0 i_1}$ and $G_{i_0 i_2}$ are the k zeroes down the leading diagonal, and so:

$$t_3(i_0, i_1, i_2) = \sum_{j=0}^{k-1} \sum_{x=1}^{s-1} h_{i_0 i_1 j}(x)\, h_{i_0 i_2 j}(x) . \qquad (4.43)$$

(It is worth noting that, since the design has no multiple lines, $h_{uwj}(x)$ is either 0 or 1 for each of the values of u, w, x, and j.)

The formula (4.17) which we derived independently for the number of triangles in a three-replicate $\alpha(0,1)$ design can be deduced from this as follows. Since the design has no multiple lines, it has no two-replicate triangles. The number of single-replicate triangles is $rsk(k-1)(k-2) = 3sk(k-1)(k-2)$. The number of three-replicate triangles is

$$6s \sum_{i_0<i_1<i_2} t_3(i_0, i_1, i_2) ,$$

and, since there are only three replicates, this equals $6st_3(0, 1, 2)$. Moreover, equation (4.43) above gives:

$$t_3(0, 1, 2) = \sum_{j=0}^{k-1} \sum_{x=1}^{s-1} h_{01j}(x)\, h_{02j}(x) ,$$

and so the formula (4.17) follows (bearing in mind that for $u = 0$ and 1, G_{0u} is the same as the earlier G_u and h_{0u} is the same as the earlier h_u.)

Finally in this Section 4.2 on triangles in α designs, we turn to some specific examples of these calculations. We will not, here, be interested in three-replicate $\alpha(0,1)$ designs, since we described examples of them earlier. What we will look at is the enumeration of triangles in, first, an $\alpha(0,1,2)$ design, and, second, an $\alpha(0,1)$ design which has four replicates.

Example (1)

This is an $\alpha(0,1,2)$ design in three replicates for 28 varieties in blocks of size 4. (So $s = 7$.) The generating array is:

$$\alpha = \begin{matrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 3 \\ 0 & 2 & 3 & 6 \end{matrix} .$$

In line with the way we presented the theory, we will deal with the three different types of triangle in turn. The formulas come from the summary on page 150 above.

I. Single-replicate triangles.

The number of such triangles is calculated easily: it is $rsk(k - 1)(k - 2)$, which is 504.

II. Two-replicate triangles.

By the formula in the summary, the number of such triangles is:

$$6s(k - 2)(t_2(0, 1) + t_2(0, 2) + t_2(1, 2)).$$

The terms in this expression can be found by forming the arrays G_{01}, G_{02}, G_{12}, and then counting the numbers of zeroes off their diagonals. The arrays are:

$$\begin{array}{c|cccc} G_{01} & 0 & 1 & 2 & 3 \\ \hline 0 & 0 & 6 & 5 & 4 \\ 1 & 1 & 0 & 6 & 5 \\ 2 & 2 & 1 & 0 & 6 \\ 3 & 3 & 2 & 1 & 0 \end{array} \qquad \begin{array}{c|cccc} G_{02} & 0 & 2 & 3 & 6 \\ \hline 0 & 0 & 5 & 4 & 1 \\ 2 & 2 & 0 & 6 & 3 \\ 3 & 3 & 1 & 0 & 4 \\ 6 & 6 & 4 & 3 & 0 \end{array} \qquad \begin{array}{c|cccc} G_{12} & 0 & 1 & 1 & 3 \\ \hline 0 & 0 & 6 & 6 & 4 \\ 1 & 1 & 0 & 0 & 5 \\ 1 & 1 & 0 & 0 & 5 \\ 3 & 3 & 2 & 2 & 0 \end{array} .$$

(The definitions of these arrays G_{uw} are given on page 150. In forming G_{01} and G_{02}, the row $u = 0$ is already a row of zeroes, For G_{12}, however, we first have to reduce row $u = 1$ to zeroes, which causes row $w = 2$ to become 0 1 1 3.)

There are no off-diagonal zeroes in G_{01} or in G_{02}, and there are 2 in G_{12}. So $t_2(0, 1) = t_2(0, 2) = 0$, and $t_2(1, 2) = 2$. So the total number of two-replicate triangles is $6s(k - 2)(0 + 0 + 2)$, which is 168.

III. Three-replicate triangles.

Referring again to the summary on page 150, we find that the number of three-replicate triangles is $6st_3(0, 1, 2)$. The quantity $t_3(0, 1, 2)$ is found from the arrays G_{01} and G_{02} given above, by means of the formula (4.36). That is, $t_3(0, 1, 2)$ is

$$\sum_{j=0}^{k-1} \sum_{x=0}^{s-1} h_{01j}(x)\, h_{02j}(x) \; - \; t_2(0, 1) \; - \; t_2(0, 2) \; - \; t_2(1, 2) \; - \; k,$$

where for u = 1 and 2, $h_{0uj}(x)$ is (as defined on page 150) the number of occurrences of x in row j of the array G_{0u}. Hence

$$t_3(0, 1, 2) \;=\; 9 - 0 - 0 - 2 - 4 ,$$

which equals 3, and so the total number of three-replicate triangles is $6s\times 3 \;=\; 126$. (As with the three-replicate triangles in the example on pages 131 - 133, the two-replicate and three-replicate triangles here could be listed by examining the design.)

Example (2)

So far, we have concentrated exclusively in the examples in this Section 4.2 on α designs in three replicates. However, the formulas on page 150 are also, of course, applicable to α designs which have a larger number of replicates. Here, therefore, we broaden our scope to a four-replicate design. The one we will look at is the best α(0,1) design in the ARCUS catalogue for 40 varieties in four replicates and blocks of size 5. The generating array is:

$$\alpha \;=\; \begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 3 & 5 & 6 \\ 0 & 2 & 7 & 3 & 5 \\ 0 & 6 & 1 & 4 & 2 \end{matrix} \; .$$

The analysis can be simplified by means of the points about α(0,1) designs which we made on pages 151 - 153. There are no two-replicate triangles, and the number of single-replicate triangles is rsk(k - 1)(k - 2) = 1920 (since s = 8). The number of three-replicate triangles is:

$$6s \sum_{i_0<i_1<i_2} t_3(i_0, i_1, i_2) ,$$

which is the same as:

$6s(t_3(0, 1, 2) + t_3(0, 1, 3) + t_3(0, 2, 3) + t_3(1, 2, 3))$.

The quantities $t_3(i_0, i_1, i_2)$ here can be calculated using equation (4.43), which, in turn, requires us to construct the arrays $G_{i_0 i_1}$ and $G_{i_0 i_2}$ for the various combinations of values of i_0, i_1, and i_2. For example, for $t_3(0, 1, 2)$, we first form the arrays G_{01} and G_{02}:

G_{01}	0	1	3	5	6
0	0	7	5	3	2
1	1	0	6	4	3
3	3	2	0	6	5
5	5	4	2	0	7
6	6	5	3	1	0

G_{02}	0	2	7	3	5
0	0	6	1	5	3
2	2	0	3	7	5
7	7	5	0	4	2
3	3	1	4	0	6
5	5	3	6	2	0

.

Then, using equation (4.43), we get that $t_3(0, 1, 2) = 9$. (That is, for this allocation of replicates to the numbers i_0, i_1, and i_2, there are 9 series of s (that is, 8) three-replicate triangles. Similarly, we get $t_3(0, 1, 3) = 10$, $t_3(0, 2, 3) = 10$, and $t_3(1, 2, 3) = 10$. So the total number of three-replicate triangles is 6s(9 + 10 + 10 + 10), which equals 1872.

4.3 The number of squares in an α design

Although the graphical criterion $\underline{\underline{C}}_3$ can provide a reasonably accurate measure of the efficiency of an α design, it is, nevertheless, frequently not as subtle as we might require.

Certainly, it can be used to exclude designs that are blatantly inefficient (and this is a point we return to in 5.3 when we construct an algorithm to generate α designs with small numbers of triangles). But even among α designs with few triangles there can be a fairly pronounced degree of variation in the values of the harmonic mean efficiency factor. This seems to be particularly true for designs with large numbers of varieties: it is then possible to exclude altogether two-replicate and three-replicate triangles and yet still fall short of the best attainable value of the harmonic mean efficiency factor. Moreover, as we observed earlier (page 150), a two-replicate $\alpha(0,1)$ design has no two-replicate and three-replicate triangles at all, and so the graphical criterion $\underline{\underline{C}}_3$ cannot tell us anything very much about its efficiency.[†]

In such cases, it becomes necessary to proceed to the next graphical criterion, $\underline{\underline{C}}_4$. That is, numbers of squares will be used to distinguish between designs with the same numbers of triangles. The purpose of this Section 4.3 is to explain how the numbers of squares in an α design can be calculated on the basis of the generating array α. The development will parallel closely the exposition for triangles, the main difference being that we will deal only with two specific series of designs. First of all, we will introduce some of the techniques and methods of argument in the context of discussing two-replicate $\alpha(0,1)$ designs. Then we will generalise slightly, to three-replicate $\alpha(0,1)$ designs, in the

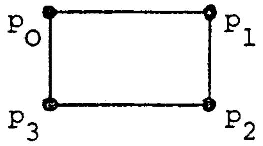
[†] This is, in fact, a particular case of the more general result on two-replicate resolvable designs which we established in 2.6.1

course of which we will show how the numbers of squares of certain types are closely related to the numbers of three-replicate triangles. We restrict our attention to these two series partly because further generalisation leads us into daunting compexity with little reward at the end of it (it being unlikely, for example - bearing in mind the conjecture in 2.5 and our decision to use the graphical criteria sequentially - that we would ever need to invoke the $\underline{\underline{C}}_4$ criterion for designs with double lines), and partly because these two series are of especial interest in themselves (being widely used in practice).

We will illustrate the theory with examples in order to elucidate how the various formulas can be applied.

4.3.1 Two-replicate $\alpha(0,1)$ designs

Our paradigmatic square can be drawn as:



(Compare the triangle on page 123.) We will (as in Chapter 3) adopt as a notational convention that such a circuit starts at the top left-hand vertex (that is, at p_0) and is traversed clockwise (that is, $p_0 - p_1 - p_2 - p_3 - p_0$).

Immediately, we encounter the same complication as arose when we were counting squares in cyclic designs (page 106): these four vertices need not be distinct; to be precise, it remains possible that p_0 is the same as p_2 or that p_1 is the same as p_3. At various stages in the development, we will have to deal with these possibilities separately. We will observe, however, that the only kinds of square that can vary in number are the proper ones: the ones in which all the vertices are distinct. We will bring together these observations at the end of the exposition (page 171).

Now, as with triangles, each line of this square corresponds to a concurrence of the varieties at each end of it, and so to a block in some replicate. Then, since there are only two replicates, the following are the only possibilities:

I Single-replicate squares: all the blocks lie in the same replicate.

II Two-replicate squares: the blocks lie in two different replicates.

We will discuss each of these types in turn, and we will find that complications arise only in the case of certain kinds of two-replicate squares. (We will be using, again, many of the techniques which we developed in the course of counting triangles.)

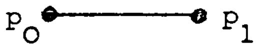
We will label the blocks B_0, B_1, B_2, and B_3 (where B_0 corresponds to the line $p_0 - p_1$, and so on round the square clockwise, and where these blocks are not necessarily, at this stage, distinct).

I. <u>Single-replicate squares</u>.

All the blocks lie in the same replicate. So, in fact, they are all the same block, since no variety lies in two blocks of any replicate. For each block, we can enumerate the number of such squares as follows, the classification (into (a) - (d)) depending on whether or not p_0 is the same as p_2 and p_1 is the same as p_3.

(a) Only two distinct vertices: $p_0 = p_2$ and $p_1 = p_3$.

Since each concurrence in a block yields exactly one line, this means that the same line starting at p_0 is traversed four times:

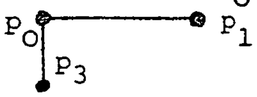


.

There are k choices for p_0, and then k - 1 choices for p_1 (which must be different from p_0, being adjacent to it). So the number here is k(k - 1).

(b) First, second and fourth vertices distinct: $p_0 = p_2$ and $p_1 \neq p_3$.

Two different lines starting at p_0 are traversed twice each:

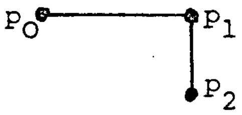


.

There are k choices for p_0, and then k - 1 for p_1. The point p_3 must be different from each of the two points p_0 and p_1, and so there are k - 2 choices for it. So the number here is k(k - 1)(k - 2).

(c) First, second and third vertices distinct: $p_0 \neq p_2$ and $p_1 = p_3$.

A path of length two which joins p_0 to a point different from p_0 is traversed twice:



.

This is similar to (b): there are k choices for p_0, then k - 1 for p_1. The next point p_2 must be different from p_1 and p_0, which are distinct from each other, and so there are k - 2 choices for p_2. So the number here is k(k - 1)(k - 2).

(d) Proper squares: all the vertices are distinct: $p_0 \neq p_2$ and $p_1 \neq p_3$.

There are k choices for p_0, then k - 1 for p_1. The vertex p_2 must be different from p_1 and p_0, which are different from each other, and so there are k - 2 choices for p_2. Then p_3 must be different from each of the three distinct points p_0, p_1, and p_2, and so there are k - 3 choices for it. So the number here is $k(k - 1)(k - 2)(k - 3)$.

Hence, bringing together the results of (a), (b), (c), and (d), the total number of circuits of length four which arise from the one block is:

$k(k - 1) + k(k - 1)(k - 2) + k(k - 1)(k - 2) + k(k - 1)(k - 2)(k - 3)$

which equals:

$$k(k - 1)(k^2 - 3k + 3).$$

So the overall total of single-replicate squares is, since there are 2s blocks altogether:

$$2sk(k - 1)(k^2 - 3k + 3) \quad . \tag{4.44}$$

II. Two-replicate squares

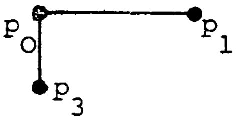
Extending the notation we used earlier for triangles, we let the two replicates involved be labelled i_0 and i_1, with i_0 being the one which occurs first: that is, the replicate in which block B_0 lies will always be labelled i_0. Again, as with single-replicate squares, we deal with four different cases (a) - (d), depending on whether or not p_0 is the same as p_2 and p_1 is the same as p_3.

(a) Only two distinct vertices: $p_0 = p_2$ and $p_1 = p_3$.

This means that each of the four lines in the circuit joins the two points p_0 and p_1. Now, some pair of these lines must be different, since not all the replicates are the same. It would follow if such a square existed that p_0 and p_1 were joined by two different lines, which would be a contradiction. So there are no circuits of this type at all.

(b) First, second and fourth vertices distinct: $p_0 = p_2$ and $p_1 \neq p_3$.

Since no pair of vertices are joined by more than one line, we have that the circuit must consist of two different lines starting at p_0, each being traversed twice:

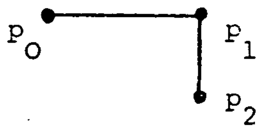


That is, $B_0 = B_1$ and $B_2 = B_3$. So, since the circuit involves two replicates (the first being i_0), the line $p_0 - p_1$ must be from the replicate i_0, and the line $p_0 - p_3$ must be from the replicate i_1. Each variety p_0 lies on k - 1 lines arising from replicate i_0, and on k - 1 lines arising from replicate i_1. Moreover, none of the lines arising from replicate i_1 joins p_0 to the same vertex as a line arising from replicate i_0 (since the design has no multiple lines). So, for each ordered pair of replicates i_0 and i_1, there are $(k - 1)^2$ circuits of this type starting at p_0, and hence $ks(k - 1)^2$ such circuits altogether (taking into account the ks possible starting points). There are two ways of allocating the replicates 0 and 1 to the labels i_0 and i_1, and so the total number of circuits here is:

$$2ks(k - 1)^2 \quad . \qquad (4.45)$$

(c) First, second and third vertices distinct: $p_0 \neq p_2$ and $p_1 = p_3$.

Again using the fact that no pair of vertices are joined by more than one line, we have that in this circuit a path of length two which joins p_0 to a point p_2 different from p_0 is traversed twice:



That is, $B_0 = B_3$ and $B_1 = B_2$. Hence, since two replicates are involved (and the first is, as always, i_0), we have that the first line is in replicate i_0, and that the second line is in replicate i_1. Each variety p_0 is joined to k - 1 varieties p_1 by a line from replicate i_0. Then p_1 is joined to k - 1 varieties p_2 (all different from p_0 since the design has no multiple lines) by a line from replicate i_1. So for each allocation of replicates to labels, there are $(k - 1)^2$ circuits of this type starting at each variety, and so $ks(k - 1)^2$ such circuits altogether. Hence, since there are exactly two ways of allocating replicates to labels, the total number of squares of this type is, again:

$$2ks(k - 1)^2 \ . \qquad (4.46)$$

(d) Two-replicate proper squares: all the vertices are distinct: $p_0 \neq p_2$ and $p_1 \neq p_3$.

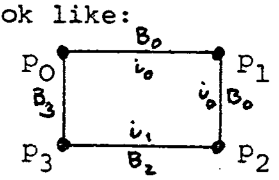
We will find that the enumeration of these squares is rather more complicated, and that, unlike single-replicate squares and two-replicate squares that are not proper, the number of two-replicate proper squares is not simply a function of s and k. Thus, as we mentioned at the beginning (page 159), when we are comparing designs, the quantities that can vary, and therefore the quantities that matter, are the numbers of these proper squares.

We show first that blocks B_0 and B_2 must lie in the replicate i_0, and that blocks B_1 and B_3 must lie in the replicate i_1.

Now, by our initial stipulation, block B_0 must lie in replicate

i_0. Suppose - towards a contradiction - that block B_2 lay in replicate i_1. We show that, then, B_1 could not lie in either replicate i_0 or replicate i_1.

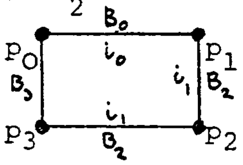
If B_1 lay in replicate i_0, then it would equal B_0 (since variety p_1 does not lie in two different blocks of that replicate). So the square would look like:



There would now be two possibilities for B_3, both of which lead directly to contradictions. (It should be emphasised that these contradictions will be subsidiary to - or nested with - the main contradiction which we are aiming at, and which we specified at the top of this page.) On the one hand, B_3 could be in replicate i_0. In that case, it would equal B_0, since variety p_0 lies in both blocks B_0 and B_3. So varieties p_3 and p_2 would both lie in block B_0, which is in replicate i_0, and would both lie in block B_2, which is in replicate i_1. This would mean that since p_3 and p_2 are distinct they would be joined by two different lines, which is a contradiction. On the other hand, B_3 could be in replicate i_1. But then B_3 would have to equal B_2, and so we would get the same kind of contradiction, with p_0 in place of p_3. (The argument works again precisely because p_0 and p_2 are distinct.)

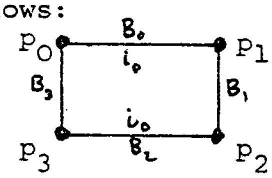
So, if B_1 lay in replicate i_0, then both these possibilities for B_3 would have resulted in a contradiction.

So - still supposing that block B_2 lies in replicate i_1 - it would have to be the case that block B_1 lay in replicate i_1 too, and consequently equalled B_2:



Then there would be two possibilities for block B_3, both of which would lead to contradictions in exactly the same way as before, with p_1 playing the role of p_2 above, p_0 the role of p_3, and p_3 the role of p_0. (We use the facts that p_1 and p_0 are distinct and that p_1 and p_3 are distinct.)

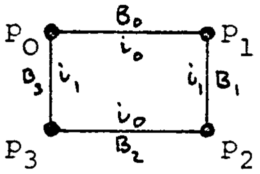
So we have shown that if block B_2 did not lie in replicate i_0 (and consequently lay in replicate i_1), then B_1 could not lie in either replicate i_0 or i_1, and that is a contradiction. Hence, we have established that block B_2 must lie in replicate i_0. So the square must be as follows:



Second, we show that blocks B_1 and B_3 both lie in replicate i_1.

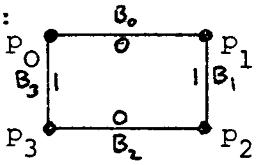
Now, if block B_1 lay in replicate i_0, then, to avoid having two lines joining varieties p_0 and p_3, we would have to have block B_3 in replicate i_0 as well. But then the square would not involve two replicates. So block B_1 must lie in replicate i_1. Likewise, to avoid a double line joining p_1 and p_2, block B_3 must lie in replicate i_1.

This completes the proof that in any proper two-replicate square, blocks B_0 and B_2 lie in replicate i_0, and blocks B_1 and B_3 lie in replicate i_1. Hence we have established that the square can be depicted as:



(4.47)

So the enumeration of proper two-replicate squares comes down to the enumeration of squares of this form (4.47). We describe next how this can be carried out. As in the case of three-replicate triangles (and of squares in cyclic designs), we will divide the enumeration here into two stages. The first stage will involve the allocations of the replicate numbers 0 and 1 to the labels i_0 and i_1. Then the second stage will entail enumerating the number of squares for each allocation. However, again as with the enumeration of three-replicate triangles for three-replicate $\alpha(0,1)$ designs, the first of these stages can be dispensed with, since the re-allocation of replicates to labels simply corresponds to choosing a new starting point; in consequence, the number of squares for each allocation is the same. (That is, the squares for the allocation $i_0 = 0$ and $i_1 = 1$ stand in a one-to-one relationship with the squares for $i_0 = 1$ and $i_1 = 0$, the relationship being defined by starting at p_1 instead of p_0.) So, in describing how to enumerate these squares, we will assume that $i_0 = 0$ and $i_1 = 1$, and we will remember to double the results at the end. So the square we will be working with looks like:



(4.48)

We set about devising rules for counting such squares by using techniques that we have adopted several times before. First, we derive two equations for each of the varieties p_u. Let the column in which p_u lies be j_u, and let B_u be the x_u^{th} block in its replicate. Then to say that varieties p_0 and p_1 concur in block B_0 is to say that they occur in the same row x_0 of the array of blocks from replicate 0. Thus:

$$p_0 = j_0 s + (x_0 \dotplus \alpha_{0j_0})$$

$$p_1 = j_1 s + (x_0 \dotplus \alpha_{0j_1})$$

Similarly, from each of the other three lines we get:

$$\begin{aligned}
\text{from } B_1\text{:} \quad p_1 &= j_1 s + (x_1 \dotplus \alpha_{1j_1}) \; ; \\
p_2 &= j_2 s + (x_1 \dotplus \alpha_{1j_2}) \; ; \\
\text{from } B_2\text{:} \quad p_2 &= j_2 s + (x_2 \dotplus \alpha_{0j_2}) \; ; \\
p_3 &= j_3 s + (x_2 \dotplus \alpha_{0j_3}) \; ; \\
\text{from } B_3\text{:} \quad p_3 &= j_3 s + (x_3 \dotplus \alpha_{1j_3}) \; ; \\
p_0 &= j_0 s + (x_3 \dotplus \alpha_{1j_0}) \; .
\end{aligned} \qquad (4.49)$$

Next, equating the pairs of expressions for each of the four varieties, we find that this set (4.49) is equivalent to:

$$\begin{aligned}
\text{from } p_0\text{:} \quad x_0 \dot{-} x_3 &= \alpha_{1j_0} \; ; \\
\text{from } p_1\text{:} \quad x_0 \dot{-} x_1 &= \alpha_{1j_1} \; ; \\
\text{from } p_2\text{:} \quad x_2 \dot{-} x_1 &= \alpha_{1j_2} \; ; \\
\text{from } p_3\text{:} \quad x_2 \dot{-} x_3 &= \alpha_{1j_3} \; .
\end{aligned} \qquad (4.50)$$

Now, if this set of equations holds, then, by eliminating the numbers x_i, we find that the following relationship amongst the

α_{1j_i} must obtain:

$$\alpha_{1j_0} \dot{-} \alpha_{1j_1} = \alpha_{1j_3} \dot{-} \alpha_{1j_2} \ . \tag{4.51}$$

Moreover, since varieties in the same column do not concur (and so are not adjacent), the j_i corresponding to adjacent vertices must be different. So we can add to (4.51) the condition:

$$\{j_0, j_2\} \cap \{j_1, j_3\} = \emptyset \ . \tag{4.51a}$$

(Henceforth, whenever equation (4.51) is referred to, it will be implicitly assumed to be governed by equation (4.51a).)

Still proceeding as we did when we were enumerating triangles, we show that each equation of the form (4.51) yields exactly s proper two-replicate squares, and that no two such equations give rise to the same square. An equation like (4.51) specifies the columns to which the vertices belong. There are, then, s choices for p_0 (or, put differently but equivalently, s choices for x_0), and this, along with the equation, fixes all four vertices. That no two equations like (4.51) yield the same proper two-replicate square in this fashion is a consequence of the fact that the terms of the equation are fully determined by the columns to which the four ordered vertices belong.

So, if there are c(0, 1) equations like (4.51), then there are sc(0, 1) proper two-replicate squares that look like diagram (4.48), and so a total of

$$2sc(0, 1) \tag{4.52}$$

proper two-replicate squares altogether (taking into account the other possible allocation of replicates to labels).

We can facilitate the calculation of $c(0, 1)$ (that is, the enumeration of equations like (4.51)) by means of the same arrays G_{uw} that were useful in enumerating triangles.[†] In fact, all we need here is the array G_{01}, whose $(j_0, j_1)^{th}$ entry is, of course, $\alpha_{1j_0} \dot{-} \alpha_{1j_1}$. Then an equation like (4.51) arises whenever there is equality between a pair of off-diagonal elements of G_{01} which lie in different rows and different columns. (These restrictions ensure that the second part of (4.51) is satisfied: namely, condition (4.51a).) The number of such equalities is not difficult to calculate. For each x in the set $\{0, 1, \ldots\ldots, s-1\}$, let $H(x)$ denote the total number of occurrences of x in the array G_{01}. Then for each x, the number of equalities of the form (4.51) which arise between cells containing an entry x is, simply, $H(x)(H(x) - 1)$. (That this is so is due to the fact that the design has no multiple lines, and hence that two different cells which contain x must lie in different rows and different columns: for example, if they lay in the same column j, and in rows u and v, then we would have $\alpha_{1u} = \alpha_{1v}$, and so, by the condition (4.1), u would equal v, and the cells would be the same.) To obtain the total number of equalities like (4.51), we need only sum over $x \in \{1, 2, \ldots\ldots, s-1\}$,

[†] As we mentioned before (in the footnote to page 130), these arrays G_{uw} are reminiscent of the structure of perfect difference sets. When we discuss this connection further in 4.4.3, we will, in particular, show that it can be used to yield lower bounds for the numbers of squares in two-replicate α designs, and, moreover, that in certain special cases, a design which actually attains this lower bound is optimal on the $\underline{\underline{A}}$-criterion as well.

for the only zero entries in the array G_{01} are the k zeroes down the leading diagonal. In other words,

$$c(0,\ 1) \quad = \quad \sum_{x=1}^{s-1} H(x)^2 - k^2 + k \qquad (4.53)$$

(using also the fact that, for the same reason, $\sum_{x=1}^{s-1} H(x) = k(k - 1)$).

So, bringing together equations (4.52) and (4.53), we have that the number of proper two-replicate squares is:

$$2s(\sum_{x=1}^{s-1} H(x)^2 - k^2 + k) \ . \qquad (4.54)$$

Notice also that in the notation we adopted when we were enumerating triangles:

$$H(x) \quad = \quad \sum_{j=0}^{k-1} h_{01j}(x)$$

($h_{01j}(x)$ being the number of occurrences of x in the row j of G_{01}).

Summary of formulas for enumerating squares in two-replicate α(0,1) designs

This completes the enumeration of squares in a two-replicate α(0,1) design. Before we describe some examples of this enumeration, we bring together the various results in the following summary.

I. Single-replicate squares

$$2ks(k - 1)(k^2 - 3k + 3) \ .$$

II. Two-replicate squares

(a) Only two distinct vertices:

None.

(b) First, second and fourth vertices distinct:

$$2ks(k - 1)^2.$$

(c) First, second and third vertices distinct:

$$2ks(k - 1)^2.$$

(d) Proper squares:

$$2s\left(\sum_{x=1}^{s-1} H(x)^2 - k^2 + k\right) .$$

As was the case with triangles, each of these expressions contains a factor s, reflecting the fact that the squares occur in series of length s, the members of which are generated by repeatedly adding 1 modulo s to the vertices. Notice also that the only type of square which can vary in number is II (d). These are proper squares: so, as we mentioned above (page 159), all circuits of length four that really matter are proper squares.

Finally in connection with two-replicate $\alpha(0,1)$ designs, we give four examples of the application of this theory. The first example shows how it can be used to enumerate circuits of length four, and the other three concern its use in comparing α designs. (The last example provides powerful evidence of the general efficacy of the graphical criterion $\underline{\underline{C}}_4$ when applied to α designs in conjuction with the conjecture in 2.5.)

<u>Example (1)</u>

We consider a two-replicate $\alpha(0,1)$ design for 40 varieties in blocks of size 4. (So s = 10.) The generating array is:

$$\alpha = \begin{matrix} 0 & 0 & 0 & 0 \\ 0 & 5 & 8 & 9 \end{matrix} .$$

So, referring to the summary above, the number of single-replicate squares is $2ks(k - 1)(k^2 - 3k + 3)$, which equals 1680. There are, as always, no two-replicate squares with only two distinct vertices, and there are $2ks(k - 1)^2$ - that is, 720 - of each of the types II (b) and (c) in the summary. In order to calculate the number of proper two-replicate squares, we need the array G_{01} (defined on page 169):

G_{01}	0	5	8	9
0	0	5	2	1
5	5	0	7	6
8	8	3	0	9
9	9	4	1	0

We then calculate the number of proper two-replicate squares by means of the equation (4.54). This requires finding the quantities $H(x)$: that is, the number of times x occurs in the array G_{01}. These quantities are given in the following table:

x	1	2	3	4	5	6	7	8	9
H(x)	2	1	1	1	2	1	1	1	2

So $\sum_{x=1}^{s-1} H(x)^2 = 18$, and, therefore, by equation (4.54), the number of proper two-replicate squares is $2s(18 - 16 + 4)$, which equals 120. So the total number of squares is:

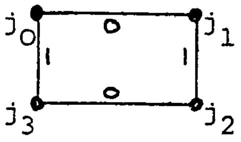
$$1680 + 720 + 720 + 120,$$

which is 3240.

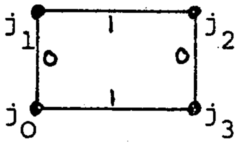
Following the development of the theory, the collection of proper two-replicate squares can be divided into 6 (that is, 18 - 16 + 4) groups of 20 (that is, 2s) each. Each group corresponds to a particular set of numbers j_u (arising from a particular equation of the form (4.51)). For each group there are 10 (that is, s) circuits with their first line in replicate 0 (that is, the allocation $i_0 = 0$ and $i_1 = 1$), and 10 circuits with their first line in replicate 1 (that is, the allocation $i_0 = 1$ and $i_1 = 0$). The six equations (4.51) are as follows:

$$
\begin{array}{lrcl|cccc}
 & & & & j_0 & j_1 & j_2 & j_3 \\ \cline{5-8}
x = 1 & \left\{ \begin{array}{l} 0 \dot{-} 9 \\ 9 \dot{-} 8 \end{array} \right. & \begin{array}{c} = \\ = \end{array} & \begin{array}{l} 9 \dot{-} 8 \\ 0 \dot{-} 9 \end{array} & \begin{array}{c} 0 \\ 3 \end{array} & \begin{array}{c} 3 \\ 2 \end{array} & \begin{array}{c} 2 \\ 3 \end{array} & \begin{array}{c} 3 \\ 0 \end{array} \\
x = 5 & \left\{ \begin{array}{l} 0 \dot{-} 5 \\ 5 \dot{-} 0 \end{array} \right. & \begin{array}{c} = \\ = \end{array} & \begin{array}{l} 5 \dot{-} 0 \\ 0 \dot{-} 5 \end{array} & \begin{array}{c} 0 \\ 1 \end{array} & \begin{array}{c} 1 \\ 0 \end{array} & \begin{array}{c} 0 \\ 1 \end{array} & \begin{array}{c} 1 \\ 0 \end{array} \\
x = 9 & \left\{ \begin{array}{l} 9 \dot{-} 0 \\ 8 \dot{-} 9 \end{array} \right. & \begin{array}{c} = \\ = \end{array} & \begin{array}{l} 8 \dot{-} 9 \\ 9 \dot{-} 0 \end{array} & \begin{array}{c} 3 \\ 2 \end{array} & \begin{array}{c} 0 \\ 3 \end{array} & \begin{array}{c} 3 \\ 0 \end{array} & \begin{array}{c} 2 \\ 3 \end{array}
\end{array} .
$$

For each of these equations, the first series of 10 has pattern of columns and replicates:



and the second series has the pattern:

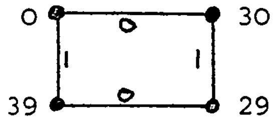


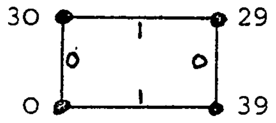
For example, for the first equation, the two series are:

(a) ($i_0 = 0$ and $i_1 = 1$)

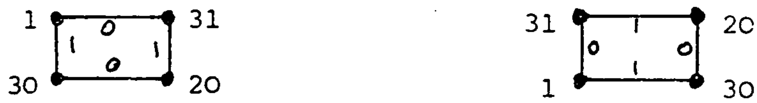
(b) ($i_0 = 1$ and $i_1 = 0$)

(i)

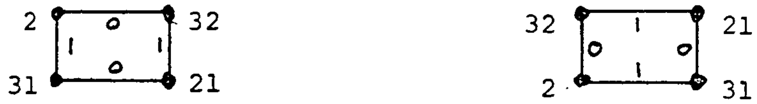




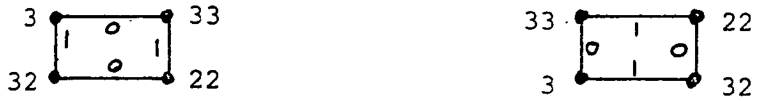
(ii)



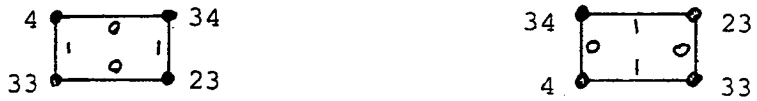
(iii)



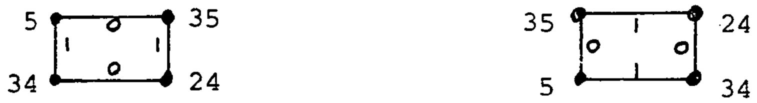
(iv)



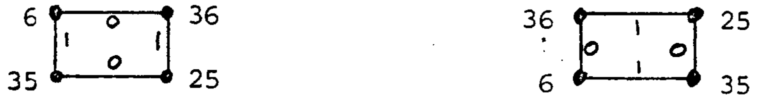
(v)



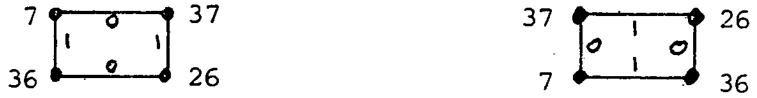
(vi)



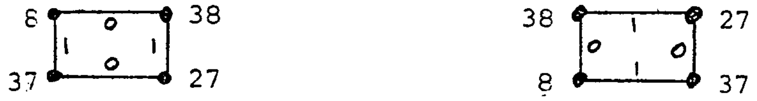
(vii)



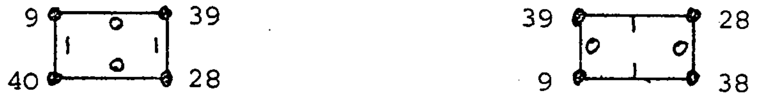
(viii)



(ix)



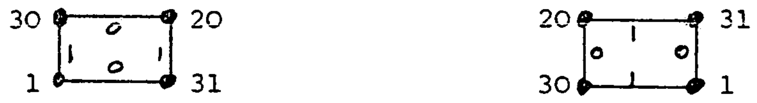
(x)



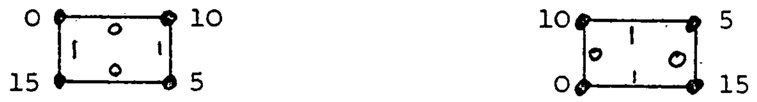
The remaining five equations have the following circuits as the first in their two series (the series being generated by repeatedly adding 1 modulo 10 to each of the vertices):

(a) (b)

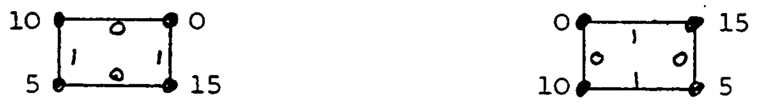
(2)



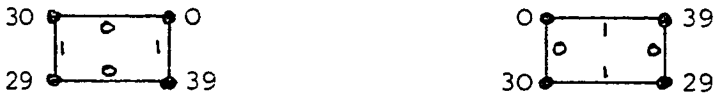
(3)



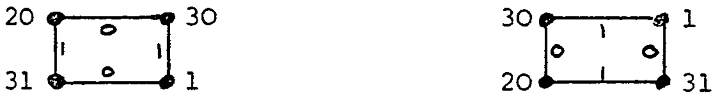
(4)



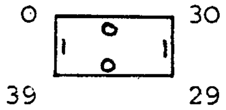
(5)



(6)



Notice how this brings out the importance of the fact that, in enumerating circuits, starting point and orientation are significant. For example (temporarily putting aside the conventions that all circuits start at their top left-hand corner and are traversed clockwise), the following basic diagram gives rise to eight circuits when the different combinations of starting points and orientations are taken into account:



The ways these different combinations arise are recorded in the following table (the ten circuits in each series being numbered (i), (ii),....., (x) as on pages 173 - 174):

Orientation	Starting point	Circuit
Clockwise	0	(1)(a)(i)
"	30	(1)(b)(i)
"	29	(6)(a)(x)
"	39	(6)(b)(x)
Anticlockwise	0	(5)(b)(i)
"	30	(5)(a)(i)
"	29	(2)(b)(x)
"	39	(2)(a)(x)

Examples (2)

Next, we look at two examples which illustrate how numbers of squares can be used to compare two-replicate designs. We will calculate the numbers of proper two-replicate squares, since these are (as we mentioned earlier, after the Summary on page 171). the only ones which can vary in number between designs.

(a) $s = 10$ and $k = 4$

In a full listing of all connected $\alpha(0,1)$ designs in two replicates for blocks of size 4 and with 10 blocks in each replicate, there are 8 different values of the harmonic mean efficiency factor $\underline{\underline{A}}$, listed in the Table 3 on page 177. The first column of the Table gives the second rows of the generating arrays of examples of designs with each of these values of $\underline{\underline{A}}$ (it being assumed that the first rows each consist entirely of zeroes), and the third column gives the corresponding number of proper two-replicate squares. (The first example is, of course, the one which we have already analysed in detail in Example (1) above.) In the listing of all $\alpha(0,1)$ designs for these values of s and k, we found that all designs with equal values of the harmonic mean efficiency factor $\underline{\underline{A}}$ have the same number of squares. However, the converse does not hold: for example, the third and fourth designs in the Table have the same numbers of squares, but different values of $\underline{\underline{A}}$. These two cannot be distinguished on the graphical criteria $\underline{\underline{C}}_2$ or $\underline{\underline{C}}_3$. (We would not, of course, be using the $\underline{\underline{C}}_4$- criterion if they could.) Neither, therefore - since they are resolvable two-replicate designs - can they be distinguished

Table 3

Two-replicate α(0,1) designs for 40 varieties in blocks of size 4: comparison of harmonic mean efficiency factor $\underline{\underline{A}}$ with number of proper two-replicate squares.

Second row of generating array α.	$\underline{\underline{A}}$	Number of proper two-replicate squares
0 5 8 9	.60241816	120
0 5 7 9	.60241652	120
0 6 7 9	.59688341	160
0 6 8 9	.59541331	160
0 2 7 9	.58945432	200
0 4 8 9	.58945275	200
0 7 8 9	.55755820	320
0 4 5 9	.53424658	400

on the $\underline{\underline{C}}_5$-criterion: this is a particular case of the result on such designs which we established in 2.6.1. (They each have, in fact, 13600 circuits of length 5.) However, they can be distinguished on the $\underline{\underline{C}}_6$-criterion, and (as we might hope from our conjecture in 2.5), the design 0 6 7 9 has fewer circuits of length six than the design 0 6 8 9: the numbers are, respectively, 74160 and 74280. (We found these values by the elementary, but laborious, method of calculating the matrices A^5 and A^6 , and then working out their Traces: we have not yet devised for circuits of length five or greater the kind of simple formulas that we have described for circuits of lengths three or four.) The tiny difference between the values of A for the first and second designs in the Table is reflected first in a difference in numbers of circuits of length 10, the first design having fewer than the second. For the fifth and sixth designs, the first difference in circuits is also at length 10: the fifth has fewer circuits of length 10 than the sixth.

(b) s = 10 and k = 5

Here, examples of all the distinct values of the harmonic mean efficiency factor which occur among the connected $\alpha(0,1)$ designs are listed (again according to decreasing value of A) in Table 4 below.

Table 4

Two-replicate $\alpha(0,1)$ designs for 50 varieties in blocks of size 5: comparison of harmonic mean efficiency factor A with number of proper two-replicate squares.

Second row of generating array α.	A	Number of proper two-replicate squares.
0 4 7 8 9	.68394157	520
0 3 5 8 9	.68213457	560
0 5 7 8 9	.68001546	600
0 4 6 8 9	.67931323	600
0 2 6 8 9	.67775399	640
0 4 5 8 9	.67228307	720
0 6 7 8 9	.66708365	800

Similar comments can be made about this Table as we made about the Table 3 in Example (2)(a). In particular, the third and fourth designs can be distinguished first on the $\underline{\underline{C}}_6$-criterion, with design 0 5 7 8 9 having fewer circuits (373240) of length six than design 0 4 6 8 9 (which has 373480).

Example (3)

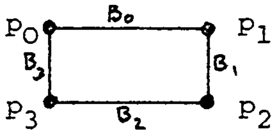
An illustration of the efficacy of the graphical criterion $\underline{\underline{C}}_4$ can be found in its application to the problem of finding the best $\alpha(0,1)$ design in two replicates for the range of s and k in the ARCUS catalogue of efficient α designs: that is,

$$\left.\begin{array}{r} 4 \le k \le s \\ \text{and } 5 \le s \le 15 \end{array}\right\} \text{ subject to } ks \le 100 \,. \tag{4.55}$$

We calculated the numbers of squares for all two-replicate $\alpha(0,1)$ designs for this range of s and k, and we found that if D and D' are any two such designs for the same values of s and k, then $\underline{\underline{A}} \le \underline{\underline{A}}'$ if, and only if, D has more squares than D' (the harmonic mean efficiency factors being, respectively, $\underline{\underline{A}}$ and $\underline{\underline{A}}'$). In other words, the graphical criterion $\underline{\underline{C}}_4$ did not contradict the $\underline{\underline{A}}$-criterion for $\alpha(0,1)$ designs in two replicates with s and k in the range (4.55): that is, the more general form of the conjecture in 2.5 was borne out for α designs with s and k in the range (4.55). This example therefore provides empirical confirmation of the power of the graphical criterion $\underline{\underline{C}}_4$ and of the validity of the conjecture. We develop these ideas further in 5.4, when we use the $\underline{\underline{C}}_4$-criterion as the basis of an algorithm for constructing efficient two-replicate $\alpha(0,1)$ designs outside the range specified in (4.55).

4.3.2 Three-replicate $\alpha(0,1)$ designs

We now generalise slightly, to calculate numbers of squares in three-replicate $\alpha(0,1)$ designs. Of course, numbers of triangles do tell us something about the efficiency of these designs, as we have already indicated in some detail. The purpose of enumerating squares will be to distinguish amongst designs which have the same numbers of triangles. Although the theory is rather more complicated than it was for two-replicate $\alpha(0,1)$ designs, we will, nevertheless, show that a great deal of this complication is of little significance for our purposes here. However, in order to reach this stage of simplification, we will, first of all, have to classify squares in the same kind of way as we did for two-replicate $\alpha(0,1)$ designs. We continue to take as our model of a typical square the diagram we set out with earlier:



As before, the classification will be according to the number of replicates the lines of the square involve. Since there are three replicates in the design, there are now three different types of square, the first two of which are obviously analogous to the two types for two-replicate $\alpha(0,1)$ designs:

I Single-replicate squares: all the blocks are in the same replicate;

II Two-replicate squares: the blocks lie in exactly two different replicates;

III Three-replicate squares: the blocks lie in three different replicates.

The number of single-replicate squares will be seen to be a function of s and k. As we have already done with triangles, and with two-replicate $\alpha(0,1)$ designs, we will divide the enumeration of two-replicate and three-replicate squares into two stages: for each type, we will, first, list the ways of choosing the appropriate replicates; than we will enumerate the number of squares of that type for each particular choice. The advantage of this two-stage approach will be largely as before: it will allow us to develop general rules for enumerating the squares for each choice of replicates. Indeed, it will transpire that this enumeration can be based on the same arrays G_{uw} that we defined earlier in the course of counting triangles. Since we will be interested in enumerating squares usually only after triangles have already been counted, these arrays will often, in practice, already have been formed, and so a great deal of work will be saved.

I. Single-replicate squares.

The counting of squares of this type is very similar to the counting of single-replicate squares in two-replicate designs. The only difference, in fact, is that there are 3s blocks instead of 2s, and so the number of single-replicate squares here is:

$$3ks(k - 1)(k^2 - 3k + 3) \; . \qquad (4.56)$$

II. Two-replicate squares.

We will adopt the same notational convention as we have employed several times before: the first replicate will be labelled i_0, and the next different one will be labelled i_1. Once we have chosen an unordered pair of replicates {u, w} to allocate to i_0 and i_1, the enumeration of two-replicate squares here is very similar to the enumeration of two-replicate squares in two-replicate designs, with u taking the place of 0 and w taking the place of 1. That is to say, first of all we reduce the array α so that the row corresponding to replicate u consists of zeroes. Then, we form the array G_{uw} whose $(j_0, j_1)^{th}$ entry is $\alpha_{wj_0} \dot{-} \alpha_{wj_1}$. Extending the notation we used earlier, we let $H_{uw}(x)$ be the number of occurrences of x in the array G_{uw} (for $x \varepsilon \{1, 2, \ldots\ldots, s-1\}$). Then it is clear that, by arguing in the same way as we did for two-replicate $\alpha(0,1)$ designs, and referring to the summary on page 170, the number of two-replicate squares for the two allocations $i_0 = u$, $i_1 = w$ and $i_0 = w$, $i_1 = u$ taken together is:

$$2s(2k(k-1)^2 - k^2 + k + \sum_{x=1}^{s-1} H_{uw}(x)^2) .$$

It follows that the total number of two-replicate squares is the sum of these quantities over all the three ways of selecting two unordered numbers from the set of three replicates {0, 1, 2} . This total is, therefore,

$$6s(2k(k-1)^2 - k^2 + k) + 2s \sum_{x=1}^{s-1} H_{01}(x)^2 + 2s \sum_{x=1}^{s-1} H_{02}(x)^2 + 2s \sum_{x=1}^{s-1} H_{12}(x)^2,$$

which is:

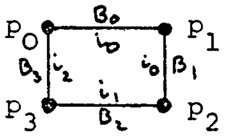
$$6ks(k-1)(2k-3) + 2s \sum_{x=1}^{s-1} (H_{01}(x)^2 + H_{02}(x)^2 + H_{12}(x)^2) . \qquad (4.57)$$

III. Three-replicate squares.

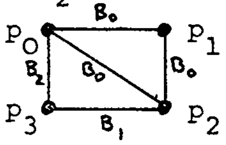
Here, all three replicates must appear among the lines; moreover, since there are four lines, some replicate must appear twice. We can simplify the enumeration of these squares by classifying them according to whether the replicate which appears twice occurs with adjacent lines or with opposite lines. The reason why this leads to a simplification is that the number of squares in the first of these classes turns out to be a function of the number of three-replicate triangles.

(a) The replicate which appears twice in the square occurs with adjacent lines.

This means that the square can be described as two triangles meeting along one of its diagonals, one triangle being of the single-replicate type and the other being of the three-replicate type. For example, if replicate i_0 is repeated on the first and second lines, then the square is of the form:



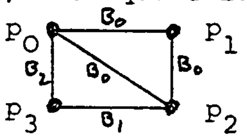
Then, since variety p_1 occurs in only one block of replicate i_0, the block B_0 is the same as the block B_1. So the block B_0 gives rise to a line joining p_0 and p_2, and hence the square can be drawn as:



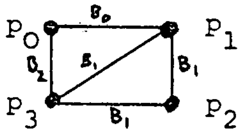
The triangle $p_0 - p_1 - p_2 - p_0$ involves only a single replicate, and the triangle $p_0 - p_2 - p_3 - p_0$ involves three replicates, and they meet along the diagonal $p_0 - p_2$.

This observation makes the enumeration of these squares very simple. First, we divide them into four mutually exclusive groups, according to which two of the four lines are associated with the repeated replicate; then we enumerate the squares in each group:

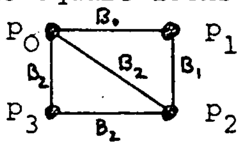
(i) First and second lines come from the same replicate. Here, since the first line must be associated with the replicate i_0, the repeated replicate is i_0, and (as we showed above), the square looks like:



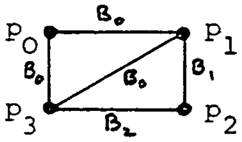
(ii) Second and third lines come from the same replicate. The first line must be associated with replicate i_0. So the repeated replicate must be i_1, and hence (arguing as above), the square looks like:



(iii) Third and fourth lines come from the same replicate. For the second line to be different from the first line, it must be associated with the replicate i_1. So the repeated replicate must be i_2, and hence (arguing, again, as above), the square looks like:



(iv) Fourth and first lines come from the same replicate. Then the repeated replicate must be i_0 (since i_0 must be associated with the first line). So the square looks like:



The number of squares in each of these four groups can be found by systematically going through all three-replicate triangles and constructing on the appropriate line of each of them a single-replicate triangle. Thus, in (i), the construction is on the first line of the triangle; in (ii), it is on the second line; in (iii), it is on the third line; and, in (iv), it is on the third line again. There are k - 2 ways of performing the construction on a given line (since the vertex that is added in this fashion to complete the square must be different from the two vertices that are joined by the line), and so if there are (as in the summary on page 150) T_3 three-replicate triangles, then there are $(k - 2)T_3$ squares in each group. Moreover, it is obvious that no square is enumerated in this way in two different groups, and so the total number of three-replicate squares in which the replicate which appears twice occurs with adjacent lines is:

$$4(k - 2)T_3 \ . \qquad (4.58)$$

(The number T_3 of three-replicate triangles can, of course, be calculated by means of the formulas which we derived earlier, and which are summarised on page 150: it equals:

$$6s \sum_{j=0}^{k-1} \sum_{x=1}^{s-1} h_{01j}(x)\ h_{02j}(x) \ ,$$

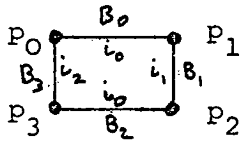
where $h_{uwj}(x)$ is the number of occurrences of x in the row j of G_{uw}.)

(b) <u>The replicate which appears twice in the square occurs with opposite lines</u>.

This is the final remaining possibility. There are two sub-cases, depending on which of the two replicates i_0 and i_1 occur twice. (Replicate i_2 cannot occur twice on opposite lines since it can only occur on the third or fourth lines to leave room for it to be preceeded by two different replicates.) It will turn out to be convenient for the development of the enumeration formulas if we stipulate that the row of the generating array α which is to be reduced to zeroes is the one corresponding to the replicate which occurs twice. This is a departure from our previous practice (where it was always the row i_0 that was reduced to zeroes). However, this will not make the formulas we derive any more complicated to apply, and it will greatly facilitate the theory, enabling us to establish rules for enumerating these squares which will be based, yet again, on the arrays G_{uw}.

(i) Replicate i_0 occurs twice.

Here, the square looks like:



Proceeding as we have done before, we let block B_u be the x_u^{th} block in its replicate (for u = 0, 1, 2, and 3). Then we can derive the following four equations from the above diagram by equating pairs of expressions for the varieties p_u:

from p_0: $x_3 \dotplus \alpha_{i_2 j_0} = x_0 \dotplus \alpha_{i_0 j_0}$;

from p_1: $\quad x_0 \dotplus \alpha_{i_0 j_1} = x_1 \dotplus \alpha_{i_1 j_1}$;

from p_2: $\quad x_1 \dotplus \alpha_{i_1 j_2} = x_2 \dotplus \alpha_{i_0 j_2}$;

from p_3: $\quad x_2 \dotplus \alpha_{i_0 j_3} = x_3 \dotplus \alpha_{i_2 j_3}$.

Then, using the fact that the row i_0 of α consists only of zeroes, we can eliminate the numbers x_u to get:

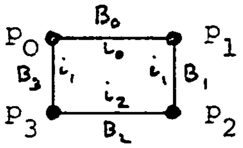
$$\alpha_{i_1 j_1} \mathbin{\dot{-}} \alpha_{i_1 j_2} = \alpha_{i_2 j_0} \mathbin{\dot{-}} \alpha_{i_2 j_3} , \qquad (4.59)$$

where $\{j_0, j_2\} \cap \{j_1, j_3\} = \emptyset$.

Conversely, each equation of the form (4.59) gives rise to exactly s three-replicate squares in which the replicate i_0 occurs with the first and third lines, and no two equations like (4.59) yield the same square of this form. (The proof of this is analogous to the many similar proofs we have given earlier.) So if there are $d_{i_0 i_1 i_2}$ equations of this form, then there are $sd_{i_0 i_1 i_2}$ three-replicate squares of this type for this allocation of replicate numbers 0, 1, and 2 to the labels i_0, i_1, and i_2. (The notation d_{uwz} is to be taken to mean that the row u is reduced to zeroes. Clearly, $d_{uzw} = d_{uwz}$. Shortly, in equation (4.62), we will derive further simplifications for these quantities.)

(ii) Replicate i_1 occurs twice.

Thus the square is of the form:



So, arguing as in (i), but this time assuming that row i_1 of the generating array α is reduced to zeroes (since it is replicate i_1 that occurs twice here), we derive the following equation, analogous to (4.59):

$$\alpha_{i_0 j_1} \dot{-} \alpha_{i_0 j_0} = \alpha_{i_2 j_2} \dot{-} \alpha_{i_2 j_3} , \qquad (4.60)$$

where $\{j_0, j_2\} \cap \{j_1, j_3\} = \emptyset$.

This equation is of exactly the same kind as equation (4.59), and so, in the same notation, there are $d_{i_1 i_0 i_2}$ equations of the form (4.60). We can show, as before, that there are, then, $sd_{i_1 i_0 i_2}$ three-replicate squares in which replicate i_1 occurs with the second and fourth lines.

The following table summarises the information about three-replicate squares in which the replicate that appears twice occurs with opposite lines; there are six allocations of replicates to labels.

			(i)	(ii)
i_0	i_1	i_2	replicate i_0 repeated	replicate i_1 repeated
0	1	2	sd_{012}	sd_{102}
0	2	1	sd_{021}	sd_{201}
1	0	2	sd_{102}	sd_{012}
1	2	0	sd_{120}	sd_{210}
2	0	1	sd_{201}	sd_{021}
2	1	0	sd_{210}	sd_{120}

The way we have defined the types (i) and (ii) in this table ensures that there is no overlap between them: every square in the first of these categories has its first replicate repeated on its third line, whereas every square in the second category has its second replicate repeated on its fourth line. So we can find the total number of three-replicate squares in which the repeated replicate occurs with opposite lines by simply summing the entries in the final two columns of the table. Furthermore, since $d_{uwz} = d_{uzw}$ for each u, w, and z, this establishes that the number of such squares is:

$$4s(d_{012} + d_{102} + d_{201}) \ . \qquad (4.61)$$

We are left with the task of calculating the quantities d_{uwz}, and to this end we derive the formula in the next equation (4.62). The notation is the same as we used earlier: the number of occurrences of x in row j of the array G_{uw} is denoted by $h_{uwj}(x)$, and the total number of occurrences of x in this array is denoted by $H_{uw}(x)$. Then the formula which we establish is:

$$d_{uwz} = \sum_{x=1}^{s-1} H_{uw}(x)H_{uz}(x) - 2\sum_{j=0}^{k-1}\sum_{x=1}^{s-1} h_{uwj}(x)h_{uzj}(x) \ . \qquad (4.62)$$

The argument which we will use to demonstrate the validity of this will be very similar to the one we followed when we were proving (4.35). In particular, the crucial tools will be, again, the indicator variables $y_{uwj_0j_1}$ which we defined on page 146. That is, $y_{uwj_0j_1}(x)$ is 1 if $G_{uw}(j_0, j_1) = x$, and is 0 if not. (Of course, u and w lie in the set {0, 1, 2}, j_0 and j_1 in the set {0, 1,..., k-1}, and x in the set {0, 1,....., s-1}.)

Now, d_{uwz} is the number of equations of the following form which arise when row u of the generating array α is reduced to zeroes:

$$\alpha_{wj_1} \dot{-} \alpha_{wj_2} = \alpha_{zj_0} \dot{-} \alpha_{zj_3} , \qquad \text{where } \{j_0, j_2\} \cap \{j_1, j_3\} = \emptyset . \tag{4.63}$$

So d_{uwz} is the number of equalities between an off-diagonal cell of G_{uw} and an off-diagonal cell of G_{uz} such that these two cells are in different rows and in different columns. The number of such equalities which arise from a particular entry $G_{uw}(j_0, j_1)$ (for $j_0 \neq j_1$) is:

$$\sum_{x=1}^{s-1} y_{uwj_0j_1}(x) \sum_{\substack{j_2,j_3=0\\ j_2\neq j_0\\ j_3\neq j_1}}^{k-1} y_{uzj_2j_3}(x) ,$$

which equals:

$$\sum_{x=1}^{s-1} y_{uwj_0j_1}(x) \sum_{j_2,j_3=0}^{k-1} y_{uzj_2j_3}(x)$$
$$- \sum_{x=1}^{s-1} y_{uwj_0j_1}(x) \sum_{\substack{j_2=0\\ j_2\neq j_0}}^{k-1} y_{uzj_2j_1}(x) - \sum_{x=1}^{s-1} y_{uwj_0j_1}(x) \sum_{\substack{j_3=0\\ j_3\neq j_1}}^{k-1} y_{uzj_0j_3}(x) .$$

Now, since the number of occurrences of x in column j of G_{uw} is the same as the number of occurrences of $\dot{-}x$ in row j of that same array G_{uw}, this equals:

$$\sum_{x=1}^{s-1} y_{uwj_0j_1}(x) H_{uz}(x) - \sum_{x=1}^{s-1} y_{uwj_0j_1}(x) h_{uzj_1}(\dot{-}x) - \sum_{x=1}^{s-1} y_{uwj_0j_1}(x) h_{uzj_0}(x) .$$

Hence, summing over j_0, the total number of equalities altogether is:

$$d_{uwz} =$$

$$\sum_{x=1}^{s-1} H_{uw}(x)H_{uz}(x) - \sum_{x=1}^{s-1}\sum_{j_1=0}^{k-1} h_{uwj_1}(\dot{-}x)h_{uzj_1}(\dot{-}x) - \sum_{x=1}^{s-1}\sum_{j_0=0}^{k-1} h_{uwj_0}(x)h_{uzj_0}(x).$$

(Again, we use the fact that the number of appearances of x in the column j is the same as the number of appearances of $\dot{-}x$ in the row j.)

Finally, for each value of j_1, the quantity:

$$\sum_{x=1}^{s-1} h_{uwj_1}(\dot{-}x)\ h_{uzj_1}(\dot{-}x)$$

can be expressed alternatively as:

$$\sum_{x=1}^{s-1} h_{uwj_1}(x)\ h_{uzj_1}(x)\ .$$

This observation establishes the validity of equation (4.62).

The calculation of these quantities d_{uwz} can be further facilitated by noticing that for each u and w,

$$H_{uw}(x) = H_{wu}(x)\ ,$$

$$h_{uwj}(x) = h_{wuj}(\dot{-}x).$$

This enables us to base the calculation of all the d_{uwz} on the three arrays G_{01}, G_{02} and G_{12}. (Thus in the entire enumeration of squares, the only arrays we will have to form will be these three.)

Bringing together these observations with equation (4.62), we have the following expressions for use in calculating formula (4.61):

$$d_{012} = \sum_{x=1}^{s-1} H_{01}(x)H_{02}(x) - 2\sum_{j=0}^{k-1}\sum_{x=1}^{s-1} h_{01j}(x)h_{02j}(x)\ ; \tag{4.64}$$

$$d_{102} = \sum_{x=1}^{s-1} H_{01}(x)H_{12}(x) - 2\sum_{j=0}^{k-1}\sum_{x=1}^{s-1} h_{01j}(\dot{-}x)h_{12j}(x)\ ; \tag{4.65}$$

$$d_{201} = \sum_{x=1}^{s-1} H_{02}(x)H_{12}(x) - 2\sum_{j=0}^{k-1}\sum_{x=1}^{s-1} h_{02j}(x)h_{12j}(x)\ . \tag{4.66}$$

(In establishing these equations, we have also used the fact that for each u and w, $H_{uw}(\dot{-}x) = H_{uw}(x)$. This is a consequence of the fact that the array G_{uw} is skew-symmetric modulo s: that is, $G_{uw}(j_0, j_1) = \dot{-}G_{uw}(j_1, j_0)$.)

<u>Summary of formulas for enumerating squares in three-replicate $\alpha(0,1)$ designs</u>.

Before describing an example of the use of these formulas, we summarise them as follows (the notation being summarised afterwards):

I. <u>Single-replicate squares</u>

By equation (4.56), the number of single-replicate squares is: $3ks(k - 1)(k^2 - 3k + 3)$.

II. <u>Two-replicate squares</u>

By equation (4.57), the number of two-replicate squares is:

$$6ks(k - 1)(2k - 3) + 2s \sum_{x=1}^{s-1} (H_{01}(x)^2 + H_{02}(x)^2 + H_{12}(x)^2) \ .$$

III. <u>Three-replicate squares</u>

(a) The repeated replicate occurs with adjacent lines.

By equation (4.58), the number here is:

$$4(k - 2)T_3 \ .$$

(b) The repeated replicate occurs with opposite lines.

By equation (4.61), the number here is:

$$4s(d_{012} + d_{102} + d_{201}) \ .$$

<u>Notation</u>

The array G_{uw} is defined on page 182. Then, for each x, $H_{uw}(x)$ is the number of occurrences of x in this array, and $h_{uwj}(x)$ is the number of occurrences of x in its row j. T_3 is the number of three-replicate triangles (for which, see the summary on page 150). And d_{uwz} is the number of equations like (4.63), which can be expressed in the simpler form (4.62). (Further simplification is possible by means of equations (4.64) - (4.66).)

One point worthy of note in connection with this enumeration is the part played by proper squares. It will be recalled that when we were enumerating squares in two-replicate α(0,1) designs, we noted that the only types of square that could vary in number were all proper. (See, specifically, page 171.) The observation we make here is analogous: the variable components in the formulas summarised above all arise from squares that are proper. That this is true of the variable part of the formula for two-replicate squares can be demonstrated in exactly the same way as we did earlier for two-replicate designs; that the same is true of the formulas for three-replicate squares is a consequence of the fact that it is impossible for a square of this type not to be proper: the mere fact that there are three replicates involved ensures that the lines, and hence the vertices, must be distinct.

Example

Finally, we give an example to illustrate how the formulas can be used to enumerate squares in three-replicate α(0,1) designs.

The example is a three-replicate α(0,1) design for 50 varieties in blocks of size 5. (So s = 10.) The generating array is:

$$\alpha = \begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 3 & 4 \\ 0 & 9 & 8 & 7 & 6 \end{matrix} .$$

In order to apply the formulas, we will need the arrays G_{01}, G_{02}, and G_{12}. Referring to the definition on page 182, these are as follows:

G_{01}	0	1	2	3	4
0	0	9	8	7	6
1	1	0	9	8	7
2	2	1	0	9	8
3	3	2	1	0	9
4	4	3	2	1	0

G_{02}	0	9	8	7	6
0	0	1	2	3	4
9	9	0	1	2	3
8	8	9	0	1	2
7	7	8	9	0	1
6	6	7	8	9	0

G_{12}	0	8	6	4	2
0	0	2	4	6	8
8	8	0	2	4	6
6	6	8	0	2	4
4	4	6	8	0	2
2	2	4	6	8	0

Then we get the following values of the relevant quantities $H_{uw}(x)$:

x	1	2	3	4	5	6	7	8	9
$H_{01}(x)$	4	3	2	1	0	1	2	3	4
$H_{02}(x)$	4	3	2	1	0	1	2	3	4
$H_{12}(x)$	0	5	0	5	0	5	0	5	0

.

We are now in a position to apply the formulas contained in the summary on page 192.

I. The number of single-replicate squares is

$$3ks(k - 1)(k^2 - 3k + 3) ,$$

which is 7800.

II. The number of two-replicate squares is

$$6ks(k - 1)(2k - 3) + 2s \sum_{x=1}^{s-1} (H_{01}(x)^2 + H_{02}(x)^2 + H_{12}(x)^2) ,$$

which is 12800.

III.(a) The number of three-replicate squares in which the repeated replicate occurs with opposite lines is $4(k - 2)T_3$, which is:

$$24(k - 2)s \sum_{j=0}^{k-1} \sum_{x=1}^{s-1} h_{01j}(x)h_{02j}(x) \ ,$$

and this equals 5760.

(b) There are $4s(d_{012} + d_{102} + d_{201})$ three-replicate squares in which the repeated replicate occurs with adjacent lines. So the first step in enumerating them is the calculation of the quantities d_{uwz}. We can do this from the arrays G_{01}, G_{02}, and G_{12} by means of the equations (4.64), (4.65), and (4.66), and we get:

$$d_{012} = 44 \ ;$$

$$d_{102} = 24 \ ;$$

$$d_{201} = 24 \ .$$

So the total number of squares of this type is:

$$4s(44 + 24 + 24),$$

which is 3680.

Hence the total number of squares altogether is:

$$7800 + 12800 + 5760 + 3680,$$

and this is 30040.

4.4 Lower bounds for numbers of circuits of lengths two and four in an α design

4.4.1 Introduction

So far in this Chapter we have been concerned to devise formulas for calculating numbers of triangles and squares in various types of α designs. But this, in itself, is of little intrinsic importance for the statistician, however interesting it may be mathematically. Where it does become useful is as part of a search for efficient designs. That was the whole point of introducing graph theory in the first place, and it is to such searching that we now turn. In the next Chapter, we describe three algorithms for constructing efficient α designs which use the formulas we have derived in the earlier Sections of this Chapter. First of all, however, before we introduce these algorithms, there is one further theoretical aspect of the formulas which has relevance to this process of searching.

That aspect concerns the calculation of lower bounds for numbers of circuits of various lengths. The purpose of this is to facilitate the searching: if a lower bound for a certain graphical criterion is known, and if, moreover, it is also known that there do exist designs which attain that bound, then the scope of the search need be no wider than the collection of such designs.

Unfortunately, however, in only two particular cases have we been able to make useful progress on the level of theory: namely, finding lower bounds for the number of circuits of length two in quite a wide range of α designs, and for the number of squares in two-replicate α(0,1) designs. These lower bounds will form the subject of this Section 4.4. In other cases, the lower bounds which we have found by theoretical methods have been insufficiently tight: that is, they are not in general attainable. In consequence, as we will outline in the next Chapter, we have had to adopt in such cases rather more heuristic methods for incorporating lower bounds into algorithms. (See, in particular, page 235.)

4.4.2 Lower bound for the number of circuits of length two

First, a lower bound for the number of circuits of length two. This is particularly simple, and will not detain us long here: the lower bound which we will use is the one attained when there are no proper circuits of length two; that is, when there are no multiple lines. In other words, an α design will attain the lower bound if, and only if, it is a (0,1) design. (The fact that such designs do indeed have a minimal number of circuits of length two follows from the argument we presented on page 73, where we showed that a design in which the off-diagonal elements of the variety adjacency matrix differ by no more than one - that is, a Regular Graph Design - is optimal on the graphical criterion $\underline{\underline{C}}_2$.)

A necessary condition for the existence of such α designs is that $k \leq s$. That this is also in many cases sufficient (and so that the lower bound is generally useful) has been shown by Patterson and Williams (1976b): they demonstrate the existence of α(0,1) designs for the following fairly wide range of combinations of r, s, and k:

(i) $r = 2$ and $k \leq s$;

(ii) $r = 3$, s odd, and $k \leq s$;

(iii) $r = 3$, s even, and $k \leq s-1$;

(iv) $r = 4$, s odd and not a multiple of 3, and $k \leq s$.

Moreover, it is also known that α(0,1) designs do exist for many values of r, s, and k outwith this range. Consequently, it seems worthwhile to construct an algorithm which will search for α designs which have no double lines. This algorithm will be called the double-line algorithm, and will be described in Section 5.2.

4.4.3(a) Lower bound for the number of squares in a two-replicate $\alpha(0,1)$ design

The second area in which we calculate a lower bound is rather more complicated. First of all, here in (a), we derive a lower bound for the number of squares in two-replicate $\alpha(0,1)$ designs. (Recall from page 151 that the $\underline{\underline{C}}_4$ criterion is the first useful one for such designs: numbers of triangles tell us nothing much about them.) Then, in (b), we will examine the structure of two-replicate $\alpha(0,1)$ designs that contain a minimal number of squares. (In the next Chapter, we will show how the theory developed here can be used to devise an algorithm - to be known as the square algorithm - which will construct such designs.)

The lower bound follows from the formulas we developed in 4.3.1 for enumerating squares in two-replicate $\alpha(0,1)$ designs. We showed that the total number, $\underline{\underline{C}}_4$, of squares in such a design is:

$$\underline{\underline{C}}_4 = 2k^2s(k-1)^2 + 2s\sum_{x=1}^{s-1} H(x)^2 .$$

(This is found by summing together the number of squares in each of the categories in the summary on page 170. Recall that $H(x)$ is the number of occurrences of the quantity x in the array G_{01} whose $(j_0, j_1)^{th}$ entry is $\alpha_{1j_0} \dot{-} \alpha_{1j_1}$.)

Now, the only variable part of this formula is the sum

$$\sum_{x=1}^{s-1} H(x)^2 ,$$

So, finding a lower bound for $\underline{\underline{C}}_4$ is equivalent to finding a lower bound for this sum. We show that a lower bound for the sum is:

$$\rho(\delta + 1)^2 + (s - 1 - \rho)\delta^2 ,$$

where ρ and δ are the unique integers which satisfy the equation:

$$k(k - 1) = \delta(s - 1) + \rho , \quad 0 \leq \rho \leq s - 2 . \tag{4.67}$$

(That is, δ is the integer part of $\frac{k(k - 1)}{s - 1}$, and ρ is the remainder. The uniqueness of these integers is guaranteed by the Euclidean Algorithm: see, for example, Birkhoff and MacLane (1965).)

To demonstrate this, two points are important: first, that the numbers $H(x)$ are integers, and, second (as we showed on page 171), that their sum is $k(k - 1)$. We invoke again an observation from number theory which we have already referred to on page 73: the partition π of any integer z into u integer parts differing by at most one has smaller sum of squares than any other partition of z into u integer parts. Translating this into notation applicable to our problem here, we have that the partition π of the integer $k(k - 1)$ into $s - 1$ integer parts differing by at most one has smaller sum of squares than any other partition of $k(k - 1)$ into $s - 1$ integer parts. In particular, the partition π has no larger sum of squares than the partition consisting of the numbers $H(x)$.

So we can find a lower bound for the sum of squares

$$\sum_{x=1}^{s-1} H(x)^2$$

by finding the sum of squares corresponding to the partition π. The argument we use for this is very similar to the one we adopted on page 73. Let the two values of the numbers in π be $y + 1$ and y, and, further, let there be n values $y + 1$ and hence $s - 1 - n$ values y. From the fact that π is a partition of $k(k - 1)$, we have that the sum of the numbers in π is $k(k - 1)$; in other words, that:

$$n(y + 1) + (s - 1 - n)y = k(k - 1) .$$

That is,

$$k(k - 1) = y(s - 1) + n .$$

Now, y and n are both integers, and $0 \leq n \leq s-1$. So, if, on the one hand, n is actually strictly less than $s - 1$, then it follows from the uniqueness of the numbers δ and ρ that $y = \delta$ and $n = \rho$. In that case, the sum of squares of the partition π is, indeed,

$$\rho(\delta + 1)^2 + (s - 1 - \rho)\delta^2.$$

On the other hand, if n equals $s - 1$, then

$$k(k - 1) = (y + 1)(s - 1)$$

and the sum of squares of the partition π is $(s - 1)(y + 1)^2$ This, again, equals

$$\rho(\delta + 1)^2 + (s - 1 - \rho)\delta^2:$$

$s - 1$ is here actually a factor of $k(k - 1)$, and so $\rho = 0$ and $\delta = \frac{k(k - 1)}{s - 1} = y + 1$.

Bringing these points together, we have established that a lower bound for the sum of squares $\sum_{x=1}^{s-1} H(x)^2$ is:

$$\rho(\delta + 1)^2 + (s - 1 - \rho)\delta^2 , \qquad (4.68)$$

where δ and ρ are the unique integers satisfying (4.67).

This lower bound (4.68) for $\sum_{x=1}^{s-1} H(x)^2$ allows us to find, in the manner we outlined above, a lower bound L_4 (say) for the number of squares in a two-replicate $\alpha(0,1)$ design. It also allows us, more significantly, to develop an algorithm for the construction of efficient two-replicate $\alpha(0,1)$ designs; we describe this - the square algorithm - in 5.4. The algorithm searches for designs which have the minimal number L_4 of squares by searching for generating arrays α which give rise to quantities $H(x)$ that differ by no more than one: when that is the case, the sum $\sum_{x=1}^{s-1} H(x)^2$ attains the lower bound (4.68), and so the number of squares attains the lower bound L_4. Unfortunately, it is not true that for every combination of the numbers s and k there will necessarily exist designs which attain this lower bound: for example, when s = 20 and k = 5, the smallest value attained by the sum $\sum_{x=1}^{s-1} H(x)^2$ is 24, whereas the lower bound (4.68) is 22. (Other examples occur when s = 16 or 17 and k = 6, and when s = 14 and k = 7.) However, experience suggests that it will usually be worthwhile searching: such designs do exist for most combinations of s and k we have looked at - including, for instance, all combinations in the ARCUS catalogue except s = 14 and k = 7. Indeed, for all but that one combination of s and k in the catalogue, not only do designs exist which attain the lower bound, but, moreover, the best listed design is among them; this provides strong support for our conjecture that the best two-replicate $\alpha(0,1)$ design on the $\underline{\underline{A}}$-criterion will always be found among those which are best on the graphical criterion $\underline{\underline{C}}_4$. (See Section 2.5.) For example, taking s = 10 and k = 7, we have that $k(k - 1) = 42$ and $s - 1 = 9$, and so that $\delta = 4$ and $\rho = 6$. So the lower bound (4.68) for $\sum_{x=1}^{s-1} H(x)^2$ is 198. This is, in fact, the value of the sum of squares which is attained by the best listed design for these values of s and k. That design has generating array

$$\alpha = \begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 3 & 4 & 5 & 7 \end{matrix} ,$$

and so has the array G_{01} equal to:

	0	1	2	3	4	5	7
0	0	9	8	7	6	5	3
1	1	0	9	8	7	6	4
2	2	1	0	9	8	7	5
3	3	2	1	0	9	8	6
4	4	3	2	1	0	9	7
5	5	4	3	2	1	0	8
7	7	6	5	4	3	2	0

Then the values of the quantities H(x) (which count the numbers of appearances of each residue x in G_{01}) are as in the following table:

x	1	2	3	4	5	6	7	8	9
H(x)	5	5	5	4	4	4	5	5	5

Hence $\sum_{x=1}^{s-1} H(x)^2 = 198$ - the same as the lower bound.

4.4.3(b)(i) A class of two-replicate $\alpha(0,1)$ designs for which optimality on the graphical criterion $\underline{\underline{C}}_4$ implies optimality on the $\underline{\underline{A}}$-criterion: k(k - 1) is an integer multiple of s - 1

Of course, attaining the minimum number L_4 of squares does not, in general, guarantee optimality on any of the other criteria of efficiency. In particular, there can, amongst designs which do attain this minimum, be quite a wide range of values of the harmonic mean efficiency factor. However, there is one class of cases where a design with the minimal number of squares is, necessarily, also optimal on the $\underline{\underline{A}}$-criterion: namely, designs for which k(k - 1) is an integer multiple of s - 1. It is to the proof of this result that we now turn. A crucial part will be played in this proof by the contraction of the two-replicate design. (See Section 1.4.3(b).) It will also be interesting to observe here some correspondences with the theory of perfect difference sets.

In summary, the proof has two stages. First, we show that, in this case, a design which has the minimum number of squares has a contraction that is balanced. (It will be to this point that the connection with difference sets will be relevant, but we will postpone discussion of that connection until after the end of both stages of the proof.) Then we will use results obtained by Patterson and Williams (1976a) to show that this implies that the design is A-optimal.

So, first of all, we show that if $k(k - 1)$ is an integer multiple of $s - 1$, and if a two-replicate $\alpha(0,1)$ design for these values of s and k has the appropriate minimum number L_4 of squares, then the contraction of the design is balanced.

When $k(k - 1)$ is an integer multiple of $s - 1$, the remainder ρ is zero, and $\delta = \frac{k(k - 1)}{s - 1}$. So, for the design to have the minimum number of squares, each of the quantities $H(x)$ must equal δ. (Strictly speaking, we have not previously stated this explicitly; however, it follows in a way that is very closely analogous to our derivation of the lower bound (4.68) on page 200.) That is, each of the numbers x in the set $\{1, 2, \ldots\ldots, s-1\}$ occurs the same number (namely, δ) of times amongst the off-diagonal entries of the array G_{01}.

Now, the contraction of this two-replicate $\alpha(0,1)$ design is, as Williams, Patterson and John (1976) and Williams (1976) point out, a cyclic design for s varieties in blocks of size k, with initial block:

$$B_O = \{\alpha_{1O}, \alpha_{11}, \ldots\ldots, \alpha_{1,k-1}\} .$$

So, to show that the contraction is balanced, we have to show that this cyclic design is balanced. That involves showing that each pair of varieties in the cyclic design concur together the same number of times. We show that each pair concur δ times. If i and j are any two varieties in the cyclic design (that is, if i and j are distinct numbers in the set $\{O, 1, \ldots\ldots, s-1\}$), then the number of times they concur together in a block of the cyclic design is the number of times the quantity $i \dot{-} j$ occurs amongst the $k(k - 1)$ differences $\alpha_{1u} \dot{-} \alpha_{1v}$ arising from the initial block B_O. And that is the number of times $i \dot{-} j$ occurs amongst the off-diagonal elements of the array G_{O1}, which number is δ.

Therefore the cyclic design is, indeed, balanced (with each pair of its varieties concurring together δ times).

The second stage of the proof entails showing that the balanced structure of the contraction implies that the original two-replicate design itself is optimal according to the $\underline{\underline{A}}$-criterion. This involves little more than citing work done by other authors. On the one hand, it is a well known result that a balanced binary design is $\underline{\underline{A}}$-optimal. (See, for example, John (1971), page 247.) So the contraction is $\underline{\underline{A}}$-optimal in the class of all binary designs for s varieties in s blocks of size k. (That the contraction is binary follows from the fact that the two-replicate design has no double lines: this ensures, by requirement (4.1) on page 121, that all the numbers in the set $\{\alpha_{10}, \alpha_{11}, \ldots\ldots, \alpha_{1,k-1}\}$ are distinct, which, in turn, guarantees that in the contraction no variety occurs more than once in any block.)

But the A-optimality of the contraction immediately implies the A-optimality of the original two-replicate design, as noted by Williams, Patterson and John (1976), page 298 (deriving from a result of Patterson and Williams (1976a)). (See the formula which links the harmonic mean efficiency factors of the design and the contraction on page 42.)

Therefore we have shown what we set out to show: that if $k(k - 1)$ is an integer multiple of $s - 1$, then a two-replicate $\alpha(0,1)$ design with the minimal number L_4 of squares has a balanced contraction, and therefore is optimal according to the A-criterion.

Before we illustrate this result by means of an example, there is one further point which is worth bringing out here: namely, the correspondence between what we have just shown and the theory of perfect difference sets. A perfect difference set is a set - say $\{x_1, x_2, \ldots\ldots, x_n\}$ - of residues modulo s such that among the $n(n - 1)$ differences $x_i \dot{-} x_j$ (for $i \neq j$), each of the residues in $\{1, 2, \ldots\ldots, s-1\}$ occurs equally often. That is precisely what we have here in the case of those designs with the minimal number of squares. The set $\{\alpha_{10}, \alpha_{11}, \ldots\ldots, \alpha_{1,k-1}\}$ which formed the second row of the generating array α, and which also constituted the initial block of the cyclic contraction, is a perfect difference set. Of course, this is reflected in the fact that each of the $s - 1$

residues occurs equally often among the off-diagonal entries of the array G_{01}. Indeed, it is, furthermore, simply a re-phrasing of what we proved on page 204: namely, that the contraction is balanced. Perfect difference sets have long been known to yield balanced cyclic designs in this way: see, for example, John (1971) (pages 275ff) or Raghavarao (1971) (page 80). The combinatorial literature contains many methods of constructing perfect difference sets. (For instance, Ryser (1963).) These methods can, by means of the theory we have outlined in this Section, be used to construct two-replicate $\alpha(0,1)$ designs which have a minimal number of squares and which are, in consequence, $\underline{\underline{A}}$-optimal.

Example

As an illustration of this theory, we consider two-replicate $\alpha(0,1)$ designs having block size 4 and having 13 blocks in each replicate. So $k(k - 1) = 12$ and $s - 1 = 12$. That is, $k(k - 1)$ is an integer multiple of $s - 1$, and $\delta = \dfrac{k(k - 1)}{s - 1}$, and so equals 1. The best listed two-replicate $\alpha(0,1)$ design for these values of s and k has generating array

$$\alpha = \begin{array}{cccc} 0 & 0 & 0 & 0 \\ 0 & 1 & 3 & 9 \end{array} .$$

So the array G_{01} for this design is:

	0	1	3	9
0	0	12	10	4
1	1	0	11	5
3	3	2	0	7
9	9	8	6	0

Among the twelve off-diagonal entries of this array, each of the twelve residues in {1, 2,....., 12} occurs exactly once. In other words, among the differences $\alpha_{1i} \dot{-} \alpha_{1j}$ for this design, each of the residues occurs equally often. So the set {0, 1, 3, 9} is a perfect difference set modulo 13. The contraction is a cyclic design with initial block {0, 1, 3, 9} , and this design is balanced: each pair of the 13 varieties in it concur together $\delta = 1$ times.

4.4.3(b)(ii) Designs which attain the minimal number of squares in cases when k(k - 1) is not an integer multiple of s - 1

In general k(k - 1) will not be an integer multiple of s - 1, and so most of the theory in (i) above will not apply. However, one important aspect of that theory does remain relevant: namely, the use of the contraction to help in the search for A-optimal two-replicate designs. We show here that in this more general case, a design which attains the minimal number L_4 of squares has a cyclic contraction which has a minimal number of circuits of length two.

The argument here is very similar to the one we used in (i): the contraction is, as always for α designs, the cyclic design which has the initial block $\{0, \alpha_{11}, \ldots\ldots, \alpha_{1,k-1}\}$. Now, if the α design has the minimal number of squares, then the sum $\sum_{x=1}^{s-1} H(x)^2$ must equal the lower bound given by equation (4.68). But that implies that the quantities H(x) differ by at most one: the result from Williams, Patterson and John (1976) which we invoked earlier (page 199) proves that if the H(x) were to differ by more than one, then the sum of their squares could not be minimal.

This means that among the off-diagonal elements of the array G_{01}, ρ of the residues occur $\delta + 1$ times, and $s - 1 - \rho$ of them occur δ times (where ρ and δ are defined in (4.67)). So in the cyclic design which is the contraction, every pair of varieties concur either δ or $\delta + 1$ times; hence the contraction is a Regular Graph Design, and so has a minimal number of circuits of length two (as we showed on page 73).

These results enable us to bring together two conjectures relevant to the search for $\underline{\underline{A}}$-optimal two-replicate $\alpha(0,1)$ designs. On the one hand, suppose that in the class of symmetric cyclic designs for s varieties in s blocks of size k, there exist designs which are Regular Graph Designs: that is, in which every pair of varieties concur together λ or $\lambda + 1$ times (for some integer λ). Then the work of Mitchell and John (1976) would suggest the conjecture that all $\underline{\underline{A}}$-optimal cyclic designs for these parameters are to be found among those cyclic designs which are Regular Graph Designs. If this conjecture is true, then it follows that any $\underline{\underline{A}}$-optimal two-replicate $\alpha(0,1)$ design has the minimal number L_4 of squares. (For, the contraction of such an $\underline{\underline{A}}$-optimal α design must, by Williams, Patterson and John (1976), page 298, be an $\underline{\underline{A}}$-optimal cyclic design; so this contraction must, by the conjecture, be a Regular Graph Design, which implies that the original α design has L_4 squares.)

On the other hand, suppose that in the class of two-replicate $\alpha(0,1)$ designs for the parameters s and k, there exist designs which have the minimal number L_4 of squares. Then our comments in 2.5 would lead us to conjecture that the $\underline{\underline{A}}$-optimal two-replicate design for these parameters has that number of squares. If, now, this conjecture is true, then it follows that any $\underline{\underline{A}}$-optimal symmetric cyclic design for s varieties in s blocks of size k is a Regular Graph Design. (For, any such cyclic design is the contraction of a two-replicate $\alpha(0,1)$ design for parameters s and k; that α design must, therefore, be $\underline{\underline{A}}$-optimal, again by Williams, Patterson and John (1976), page 298; so it must, by this second conjecture, have the minimal number L_4 of squares, which, with the theory we have presented above, implies that the contraction is a Regular Graph Design.)

The equivalence of these two conjectures is a special case of the more general equivalence of analogous conjectures which we established in 2.6.2. We could, of course, have deduced this particular equivalence from that earlier result. However, that would have obscured some interesting features of the structure of the variety concurrence graphs of α designs which have been elicited by the method we have adopted here: for example, the connection with balanced cyclic designs, with Regular Graph Designs, and, above all, with the lower bound L_4 for the number of squares.

CHAPTER 5

CONSTRUCTING EFFICIENT α DESIGNS

5.1 Introduction: outline of general approach

We have seen now that the graphical criteria of efficiency are interesting at several different levels of mathematical abstraction and practical applicability. On the one hand, they have (as our initial exposition in Chapter 2 made clear) many strengths of a purely theoretical nature: the link which they provide between the combinatorial structure of the design and its efficiency is intuitively satisfying in that it makes some mathematical sense of our heuristic expectations.

But of rather greater importance, on the other hand, (as we mentioned in 2.5) are the implications of the graphical criteria for the selection of designs in practice; and we have already shown - in Chapters 3 and 4 - that they offer an attractive substitute for the A-criterion, involving, as they do, calculations that are - owing to the basis of the criteria in the variety concurrence graph - relatively straightforward.

Yet, simply to make that statement is (as we mentioned on page 196) to say very little of real use to the experimenter. It would be superfluous - if mathematically interesting and elegant - merely to have yet another series of measures for comparing designs that have already been constructed - no matter that these measures may be considerably easier to calculate than the ones employed hitherto. What is needed, besides, is some indication of how the graphical criteria can be incorporated into simple and rapid algorithms for the generation of designs with a reasonably high efficiency. Only once this has been done can graph theory be said to have been of some real practical benefit.

The purpose of this Chapter is to outline three ways in which this could be achieved. Of course, what we suggest here will be by no means the only method of using the graphical criteria to help in the generation of efficient designs. Indeed, as we will make clear (page 214), there is an obvious, if clumsy, way in which the criteria could be used to modify any plan of systematic searching so as to avoid designs that are, in some defined sense, inefficient. But the important feature of the algorithms that we will describe here is that they do use some of our earlier results in a rather more subtle manner, as part of their essential mechanism. Although they might not make the best possible use of these results (and we will indicate where they are particularly weak), they will, we hope, be a reasonable basis on which to build methods of greater refinement.

We describe three algorithms for constructing efficient α designs. The first is based on the graphical criterion $\underline{\underline{C}}_2$, and is applicable to all combinations of parameters r, s, and k. We will call it the double-line algorithm because its purpose is to construct designs with a minimal number of double-lines. The other two introduce some extra refinement, but for rather more restricted values of the parameters: the one invokes the graphical criteria $\underline{\underline{C}}_2$ and $\underline{\underline{C}}_3$ to construct efficient three-replicate α designs; since its purpose is to produce designs with a minimal number of triangles, we will call it the triangle algorithm. And the other invokes the graphical criteria $\underline{\underline{C}}_2$ and $\underline{\underline{C}}_4$ to construct efficient two-replicate designs; this one we call the square algorithm since it lists designs with a minimal number of squares. If the conjecture in 2.5 is true, then among the designs listed by these algorithms will be the best α designs on the $\underline{\underline{A}}$-criterion.

First of all, however, before going into details, we will discuss briefly some of the general ideas which have guided our approach.

The algorithms have a common underlying structure based on two tree searches. The first of these searches concerns the selection of replicates (or, in other words, rows of the generating array α): designs with r + 1 replicates will be produced by adding an $(r + 1)^{th}$ replicate to designs with r replicates (that is, by adding an $(r + 1)^{th}$ row to a given r rows of the array α). The second tree search (which is embedded in the first) concerns the choice of numbers to fill the $(r + 1)^{th}$ row of the array α: these are selected sequentially from the left, and, in principle (before, that is, any algebraic refinement is introduced), there are s ways of filling each of the k places in the row (from the set {0, 1,....., s-1}). In what follows in the rest of this Chapter, we do not modify in any way the first of these searches: we will always select replicates sequentially; in consequence, the algorithms would be more accurately described as algorithms for adding an $(r + 1)^{th}$ replicate to a given set of r replicates. Of course, this comes down to the same thing when constructing two-replicate designs - if, as we can assume, the first row of the generating array α always consists entirely of zeroes. However, for the other two algorithms - the double-line algorithm and the triangle algorithm - this sequential addition of replicates could, probably, be improved: perhaps, indeed, analogously to the refinements we introduce here into the selection of the $(r + 1)^{th}$ row.

So we are interested in adding an $(r + 1)^{th}$ row to a given set of r rows of the generating array α in such a way that the resulting design in r + 1 replicates has a reasonably high efficiency (according to the particular criterion we are invoking). Clearly, the crudest way of doing this would be to go through, systematically, all the s^k possible $(r + 1)^{th}$ rows and discard those with a value of $\underline{\underline{A}}$ less than a certain specified threshold. Such a method would, of course, be excessively laborious, and it is precisely our purpose here to show how this crude technique can be considerably refined by means of some of the theory we discussed in Chapter 4. But although we allude to this technique only to set about modifying it immediately, it remains, nevertheless, the underlying structure in all that follows.

5.2 Algorithm using the graphical criterion $\underline{\underline{C}}_2$: the double-line algorithm

One refinement of that crude structure would be to substitute for the $\underline{\underline{A}}$-criterion one (or more) of the graphical criteria. This would, certainly, reduce the amount of calculation that would have to be carried out. Indeed, any process of complete systematic searching could be refined in this obvious though clumsy way: namely, by calculating the number of circuits of some length h produced by each of the possible $(r + 1)^{th}$ rows, and then rejecting those rows which gave rise to a value of $\underline{\underline{C}}_h$ greater than a certain threshold. (This is the point which we anticipated above on page 211; we return to it on page 232 below.) But the algorithm we now describe makes rather more subtle use of the graphical criterion $\underline{\underline{C}}_2$. Instead of referring to the criterion only after the whole new row has been constructed, the algorithm will invoke it each time an individual element is chosen. That is, the $\underline{\underline{C}}_2$-criterion will be used to restrict the amount of searching that has to be gone through by excluding at as early a stage as possible certain possibilities for elements of the new row.

Expressed differently - in terms of the variety concurrence graph - what this amounts to is as follows. We want to construct designs with as few double lines in their graphs as possible. So we use this aim as the principle which guides the selection of each successive element of the new row. In particular, our interest here will be to find designs which have no double lines at all: that is, to find $\alpha(0,1)$ designs. So we use some of the theory we outlined in Chapter 4 to help us select elements for the added row which do not give rise to any double lines. It is the resulting mechanism which we will oall the double-line algorithm.

It will be illuminating if we carry with us a particular example: this will at each stage help to clarify our general points by making them more concrete, and, therefore, more apprehensible. The example we will use is quite simple. The initial generating array α to be extended is:

$$\alpha = \begin{array}{ccccc} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 3 & 5 & 2 \end{array} \quad ;$$

that is, there are two replicates and each block is of size 5. The value of s will be taken to be 6. So the design generated by the array α is a two-replicate $\alpha(0,1)$ design for 30 varieties. The aim of the algorithm when applied to this example will be, then, to add a third row to α so that the new array will generate a three-replicate $\alpha(0,1)$ design.

The particular piece of theory we adduce is on page 121: equation (4.2). We pointed out there that the number of double lines in the variety concurrence graph is s times the number of equalities of the following form among elements of the array α:

$$\alpha_{mu} \dot{-} \alpha_{mu'} = \alpha_{m'u} \dot{-} \alpha_{m'u'} \qquad \text{where } u \neq u' \text{ and } m \neq m' \; . \tag{5.1}$$

Thus, for a design to have no double lines, there must be no equalities of this type. In other words, an array α yields a (0,1) design if, and only if, for each pair u and u' of different columns of it, the r quantities $\alpha_{mu} \dot{-} \alpha_{mu'}$ are all distinct (for $0 \leq m \leq r-1$). Notice that the array α in our example yields no equalities of the form (5.1): the design has no double lines.

Our description of how this enters into the selection of elements for the new row will facilitated if we denote the collection of these r quantities $\alpha_{mu} \dot{-} \alpha_{mu'}$ (for $0 \leq m \leq r-1$) by a single symbol. This symbol we will choose to be $D_{uu'}$. That is,

$D_{uu'}$ is the set $\{\alpha_{mu} \dot{-} \alpha_{mu'} : 0 \leq m \leq r-1\}$. Then we will let D refer to the entire $r \times \frac{1}{2}k(k - 1)$ array consisting of the sets $D_{uu'}$ in the following pattern:

$$D_{01}, D_{02}, \ldots\ldots, D_{0,k-1}; D_{12}, D_{13}, \ldots\ldots, D_{1,k-1}; D_{23}, \ldots\ldots \text{etc.}$$

It will be appropriate - for obvious reasons - to call this array D the difference array of the design. So a necessary and sufficient condition for the design to have no double lines is that each of the columns of its difference array should contain r distinct elements.

In our example, D has 2 rows and 10 columns. Its first column, for instance, consists of the elements $0 \dot{-} 0$ and $0 \dot{-} 1$: that is, 0 and 5. The whole difference array is in this case:

D_{01}	D_{02}	D_{03}	D_{04}	D_{12}	D_{13}	D_{14}	D_{23}	D_{24}	D_{34} .
0	0	0	0	0	0	0	0	0	0
5	3	1	4	4	2	5	4	1	3

The reason why this notation simplifies the description is that when we add a row to the array α, the columns $D_{uu'}$ of the difference array D undergo only a very simple modification: they each acquire an extra element and are otherwise unchanged. That is, if the new difference array is called D', then its columns are related to those of the original difference array by the simple identity:

$$D'_{uu'} = D_{uu'} \cup \{\alpha'_{ru} \dot{-} \alpha'_{ru'}\}$$

(where the elements of the row added to the generating array have been denoted by $\alpha'_{r0}, \ldots\ldots, \alpha'_{r,k-1}$: since numbering of the rows is from 0, the first suffix here is r, not r + 1.)

Thus, for example, if we were to extend the array α we had above to the array:

$$\alpha' = \begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 3 & 5 & 2 \\ 0 & 2 & 1 & 4 & 5 \end{matrix} ,$$

then the new difference array D' would be:

D'_{01}	D'_{02}	D'_{03}	D'_{04}	D'_{12}	D'_{13}	D'_{14}	D'_{23}	D'_{24}	D'_{34}
0	0	0	0	0	0	0	0	0	0
5	3	1	4	4	2	5	4	1	3
4	5	2	1	1	4	3	3	2	5

This array is obtained from the original difference array D by adding appropriate elements $\alpha'_{2u} \dot{-} \alpha'_{2u'}$ to the bottom of the columns $D_{uu'}$: for example, D_{01} has added to it the element $\alpha'_{20} \dot{-} \alpha'_{21}$ which equals $0 \dot{-} 2$.

Now, as we noted at the top of the previous page, an extension of the given array α will generate a design with no double lines if, and only if, the numbers in any column of the new difference array D' are all distinct. Thus, in the example, that the extended generating array does yield an α(0,1) design is confirmed by the fact that no column of the corresponding difference array D' contains any element twice.

One necessary condition for this to happen is that each of the original columns $D_{uu'}$ should contain r distinct elements, or, in other words, that the array α which we started out with should generate a design that has no double lines. So, in what follows, this will always be assumed to be the case. Then, given that this holds, the extension will generate a design with no double lines if, and only if, each of the additional quantities $\alpha'_{ru} \dot{-} \alpha'_{ru'}$ is distinct from the elements already in the corresponding column of the original difference array D.

It is this re-phrased form of the condition (5.1) which we use as a guide in the selection of elements for the new row of the generating array α. The point of employing this alternative form of that condition is that it enables us to shift the problem from extending the array α to extending the array D - and allows us thereby to ensure the absence of double lines in a much more straightforward fashion. Of course, not every extension of D to $r + 1$ rows can be related to an extension of α: the extra row of D' must conform to the pattern defined by the differences $\alpha'_{ru} \dot{-} \alpha'_{ru'}$. Furthermore, amongst those extensions of D which do correspond to an extension of α, by no means all will yield a design that has no double lines. So, if we are to use the difference array D as a means of extending the array α, then we must establish necessary and sufficient conditions for an extension of D to correspond, first, to some extension of α, and, second and more particularly, to an extension of α that generates a design with no double lines.

We will deal with these two problems separately. First, then, what is the condition that must be satisfied by a row of numbers added to D in order for the new, larger, difference array D' to correspond to some extension of the generating array α? This question can be answered immediately: clearly, this condition is that there exist k numbers α'_{rj} (for $0 \leq j \leq k-1$) such that the final element in the column $D'_{uu'}$ of D' should be $\alpha'_{ru} \dot{-} \alpha'_{ru'}$ for each pair u and u' $(0 \leq u < u' \leq k-1)$: this is exactly what is meant by saying that the array D' must conform to the pattern of having arisen as the difference array of a generating array α'.

We will now express this in a slightly different form which will prove more useful in the development of the algorithm. All we are interested in, as a matter of fact, are generating arrays α' that are in standard form. So we can stipulate without any loss of generality (that is, without, essentially, missing any designs) that in the extended generating array α', the first element of the added row is zero: that is, that $\alpha'_{r0} = 0$. Hence, if the extension D' of the difference array D corresponds to an array α', then the first k - 1 entries of the added row of D' must be, for $1 \leq u' \leq k-1$, $\alpha'_{r0} \dot{-} \alpha'_{ru'}$, which, in consequence of our stipulation, equals $\dot{-}\alpha'_{ru'}$. Then, once we have chosen these k - 1 elements, the remainder of the last row of D is fully determined, since every remaining element in that row can be expressed as a difference of some pair of these first k - 1. Moreover, any choice of elements to fill the first k - 1 places corresponds to an extension of the array α.

In other words, the first condition (namely, that the extension of D should correspond to an extension of α) is satisfied if we confine ourselves to choosing these k - 1 elements, and then form the remainder of the new row accordingly.

Thus, in our example - which has initial difference array D defined on page 216 - if we choose 4 elements to be at the bottom of the 4 columns D_{01}, D_{02}, D_{03}, and D_{04}, and then form the rest of the third row according to the pattern of differences, then we will have a difference array D' corresponding to some α design in three replicates (though not necessarily, as yet, a design with no double lines). For instance, if we chose as the 4 elements the numbers 1, 2, 3, and 5 (that is, if we let $\alpha'_{21} = \dot{-}1$, $\alpha'_{22} = \dot{-}2$, $\alpha'_{23} = \dot{-}3$, and $\alpha'_{24} = \dot{-}5$), then we would get the extended difference array:

D'_{01}	D'_{02}	D'_{03}	D'_{04}	D'_{12}	D'_{13}	D'_{14}	D'_{23}	D'_{24}	D'_{34}
0	0	0	0	0	0	0	0	0	0
5	3	1	4	4	2	5	4	1	3
1	2	3	5	1	2	4	1	3	2

This method might not, at first sight, appear to be particularly helpful. In a sense, indeed, it is saying no more than that if the extended generating array α' is in standard form, then the sub-array of D' consisting of its first $k - 1$ columns contains simply the negatives modulo s of the entries of the array α'. However, the point of expressing things in this way is that allows us to incorporate very easily the second of the two conditions: namely, that the extension of α must generate a design with no double lines. That second condition requires that we must choose the $k - 1$ elements $\alpha'_{ru'}$, $(1 \leq u' \leq k-1)$ in such a way that, for each column $D_{uu'}$ of the original difference array D, the extra difference $\alpha'_{ru} \dot{-} \alpha'_{ru'}$ is distinct from the elements already in it.

For instance, in our example, the choice of 4 elements which we have just made (namely $\alpha'_{21} = 5$, $\alpha'_{22} = 4$, $\alpha'_{23} = 3$, and $\alpha'_{24} = 1$) does not yield an $\alpha(0,1)$ design, for the new difference array D' does not satisfy that necessary condition: the element 2 which is added to column D_{13} already appears in its second row.

We show now how this second condition can be incorporated as a set of rules for choosing the first k - 1 elements of the row that is to be added to the difference array D. It will be rather more straightforward if we talk about choosing the elements $\dot{-}\alpha'_{ru'}$, it being these, rather than their negatives, which appear at the feet of the first k - 1 columns of D'. So, for convenience, we will label $\dot{-}\alpha'_{ru'}$ as $x_{u'}$ $(0 \leq u' \leq k-1)$, where, of course, $x_0 = 0$. Then the condition for the design to have no double lines becomes: $x_u \dot{-} x_{u'}$ should not lie in column $D_{u'u}$ for $0 \leq u' < u \leq k-1$. The method which we propose will involve choosing each x_u in such a way that for every $u' < u$, $x_u \dot{-} x_{u'} \notin D_{u'u}$. Clearly this will be sufficient to ensure that the design has no double lines. (It would for example have prevented us choosing $x_3 = 3$ in the illustration on page 220.) Its advantage for our purpose here is that it defines rules for choosing x_u which depend only on the array D (which we start out with) and on those quantities $x_{u'}$ which have already been chosen. Thus this method provides an acceptable basis for an algorithm.

Before we describe the steps of this algorithm and illustrate its application to our example, however, we introduce one further simplification which will make it easier to operate (especially on a computer). This simplification takes further advantage of the fact that the rules for choosing x_u have been expressed entirely in terms of those $x_{u'}$ which have already been selected. The requirement that $x_u \dot{-} x_{u'}$ should not lie in the column $D_{u'u}$ of the original difference array D is equivalent to the requirement that x_u should not lie in the set $x_{u'} \dot{+} D_{u'u}$. So, after having chosen the element $x_{n'}$ at Step (n'), the algorithm will modify the difference array D by adding $x_{n'}$ to each of the columns $D_{n'v}$ for $v \geq n' + 1$. Then, at subsequent Steps (n), double lines will be avoided simply by choosing x_n so that it does not lie in any of the columns whose second suffix is n. That is, expressed more formally, the Steps of the double-line algorithm are:

Step (O):

Set $x_O = O$.

Step (1):

(a) Choose $x_1 \notin D_{O1}$.

(b) Add x_1 to all the elements in columns D_{1v} $(2 \leq v \leq k-1)$. Call the new array $D^{(1)}$, and its columns $D^{(1)}_{u'u}$.

Step (2):

(a) Choose $x_2 \notin D^{(1)}_{O2} \cup D^{(1)}_{12}$.

(b) Add x_2 to all the elements in the columns $D^{(1)}_{2v}$ $(3 \leq v \leq k-1)$. Call the new array $D^{(2)}$, and its columns $D^{(2)}_{u'u}$.

In general, Step (n) (for $n \leq k-2$):

(a) Choose $x_n \notin \bigcup_{u=o}^{n-1} D^{(n-1)}_{un}$.

(b) Add x_n to all the elements in the columns $D^{(n-1)}_{nu'}$ $(n+1 \leq u' \leq k-1)$. Call the new array $D^{(n)}$, and its columns $D^{(n)}_{u'u}$.

(Of course, the final Step (k-1) does not involve a part (b).)

Having chosen the numbers x_u $(O \leq u \leq k-1)$ in this fashion, we let $\alpha'_{ru} = \dot{-}x_u$.

These Steps will be made clearer if we illustrate their application to the example which we have been discussing all along. The original difference array D is given on page 216. We will follow through the Steps in some detail in order to expose fully how they work.

Step (0)

Let $x_0 = 0$.

Step (1)

(a) Choose $x_1 \notin D_{01}$. That is, choose x_1 so that it does not lie in the first column on the left of the array D. This column contains the elements 0 and 5. So we have to choose x_1 to lie in the set $\{1, 2, 3, 4\}$; say $x_1 = 1$.

(b) Form the array $D^{(1)}$. That is, add x_1 to all the elements in the columns D_{12}, D_{13}, and D_{14}. So the new array $D^{(1)}$ is (with $x_1 = 1$ written at the bottom of column $D^{(1)}_{01}$):

$D^{(1)}_{01}$	$D^{(1)}_{02}$	$D^{(1)}_{03}$	$D^{(1)}_{04}$	$D^{(1)}_{12}$	$D^{(1)}_{13}$	$D^{(1)}_{14}$	$D^{(1)}_{23}$	$D^{(1)}_{24}$	$D^{(1)}_{34}$
0	0	0	0	1	1	1	0	0	0
5	3	1	4	5	3	0	4	1	3
1									

Step (2)

(a) Choose $x_2 \notin D^{(1)}_{02} \cup D^{(1)}_{12}$. That is, choose x_2 so that it does not lie in the columns $D^{(1)}_{02}$ or $D^{(1)}_{12}$ of the array $D^{(1)}$. These columns contain the elements 0, 1, 3, and 5. So we must choose x_2 in the set $\{2, 4\}$. For example, $x_2 = 2$.

(b) Form the array $D^{(2)}$. That is, add x_2 to all the elements in the columns $D^{(1)}_{23}$ and $D^{(1)}_{24}$. Then $D^{(2)}$ is (with $x_2 = 2$

written at the feet of columns $D_{02}^{(2)}$ and $D_{12}^{(2)}$):

$D_{01}^{(2)}$	$D_{02}^{(2)}$	$D_{03}^{(2)}$	$D_{04}^{(2)}$	$D_{12}^{(2)}$	$D_{13}^{(2)}$	$D_{14}^{(2)}$	$D_{23}^{(2)}$	$D_{24}^{(2)}$	$D_{34}^{(2)}$
0	0	0	0	1	1	1	2	2	0
5	3	1	4	5	3	0	0	3	3
1	2			2					

Step (3)

(a) Choose $x_3 \notin D_{03}^{(2)} \cup D_{13}^{(2)} \cup D_{23}^{(2)}$. That is, choose $x_3 \notin \{0, 1, 2, 3\}$. So we must choose x_3 to lie in the set $\{4, 5\}$: say, $x_3 = 4$.

(b) Form the array $D^{(3)}$ by adding x_3 to all the elements in the column $D_{34}^{(2)}$. Then the array $D^{(3)}$ is:

$D_{01}^{(3)}$	$D_{02}^{(3)}$	$D_{03}^{(3)}$	$D_{04}^{(3)}$	$D_{12}^{(3)}$	$D_{13}^{(3)}$	$D_{14}^{(3)}$	$D_{23}^{(3)}$	$D_{24}^{(3)}$	$D_{34}^{(3)}$
0	0	0	0	1	1	1	2	2	4
5	3	1	4	5	3	0	0	3	1
1	2	4		2	4		4		

Step (4)

Finally, choose $x_4 \notin D_{04}^{(3)} \cup D_{14}^{(3)} \cup D_{24}^{(3)} \cup D_{34}^{(3)}$, which equals $\{0, 1, 2, 3, 4\}$. So we must choose $x_4 = 5$.

That is, we have $x_1 = 1$, $x_2 = 2$, $x_3 = 4$, and $x_4 = 5$. This means that the third row of the extended generating array α' is made up of the negatives of these numbers: namely, 5, 4, 2, and 1. So the new, extended, array α' which the algorithm has produced is:

$$\alpha' = \begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 3 & 5 & 2 \\ 0 & 5 & 4 & 2 & 1 \end{matrix} \; .$$

Next, we show how the theory we have presented and illustrated can be used as the basis of an algorithm for finding all extensions of a given r-replicate $\alpha(0,1)$ design to an $(r + 1)$-replicate $\alpha(0,1)$ design. As we mentioned earlier (page 212), this will involve incorporating what we have presented into a tree search: indeed, the above Steps become the means of limiting the number of branches of the tree that have to be searched.

Each branch of the tree corresponds to a particular choice of the numbers $x_1, x_2, \ldots\ldots,$ and x_{k-1}. Thus, before the theory is used to exclude certain combinations of numbers, there are s^{k-1} branches altogether. (It is s^{k-1} rather than the s^k which we mentioned earlier because we are now assuming that $x_0 = 0$.) The theory enables this number to be reduced by a large amount, by excluding certain branches before they are complete. To be precise, a branch which has not reached x_{k-1} can be terminated as soon as it is no longer possible to choose the next x_n at Step (n)(a): that is, as soon as the set:

$$\bigcup_{u=0}^{n-1} D_{un}^{(n-1)}$$

contains all the elements in $\{0, 1, \ldots\ldots, s-1\}$. In other words, a branch stops either when it has yielded $k - 1$ numbers $x_1, x_2, \ldots\ldots,$ and x_{k-1}, or else when it reaches a Step (n) at which the columns $D_{un}^{(n-1)}$ $(0 \leq u \leq n-1)$ contain between them all the numbers in the set $\{0, 1, \ldots\ldots, s-1\}$.

To illustrate this, we return to the example, and to the branch which starts by choosing $x_1 = 2$. (This is, of course, one of the permissible choices for x_2, as we noted on page 223 under Step (1).)

Then, for this value of x_1, the array $D^{(1)}$ is:

$D^{(1)}_{01}$	$D^{(1)}_{02}$	$D^{(1)}_{03}$	$D^{(1)}_{04}$	$D^{(1)}_{12}$	$D^{(1)}_{13}$	$D^{(1)}_{14}$	$D^{(1)}_{23}$	$D^{(1)}_{24}$	$D^{(1)}_{34}$
0	0	0	0	2	2	2	0	0	0
5	3	1	4	0	4	1	4	1	3
2									

Then, at Step (2), we must choose x_2 so that it does not equal any of the numbers already in the columns $D^{(1)}_{02}$ or $D^{(1)}_{12}$. This means that we must choose x_2 to be 1, 4, or 5. Suppose that we choose it to be 5. Then the array $D^{(2)}$ is:

$D^{(2)}_{01}$	$D^{(2)}_{02}$	$D^{(2)}_{03}$	$D^{(2)}_{04}$	$D^{(2)}_{12}$	$D^{(2)}_{13}$	$D^{(2)}_{14}$	$D^{(2)}_{23}$	$D^{(2)}_{24}$	$D^{(2)}_{34}$
0	0	0	0	2	2	2	5	5	0
5	3	1	4	0	4	1	3	0	3
2	5			5					

Next, we have to choose x_3 so that it does not equal any of the elements that are already in the columns $D^{(2)}_{03}$, $D^{(2)}_{13}$, or $D^{(2)}_{23}$. But this is impossible, since these three columns together already contain all six of the elements in the set {0, 1, 2, 3, 4, 5} . In other words, this branch of the tree is terminated here because there are no choices available for the next new element of the third row, x_3.

If we pursue this type of argument for each of the branches, then we get in this example the tree in Figure 2 (page 227). The eight completed branches of this tree correspond to the eight extensions of the given generating array

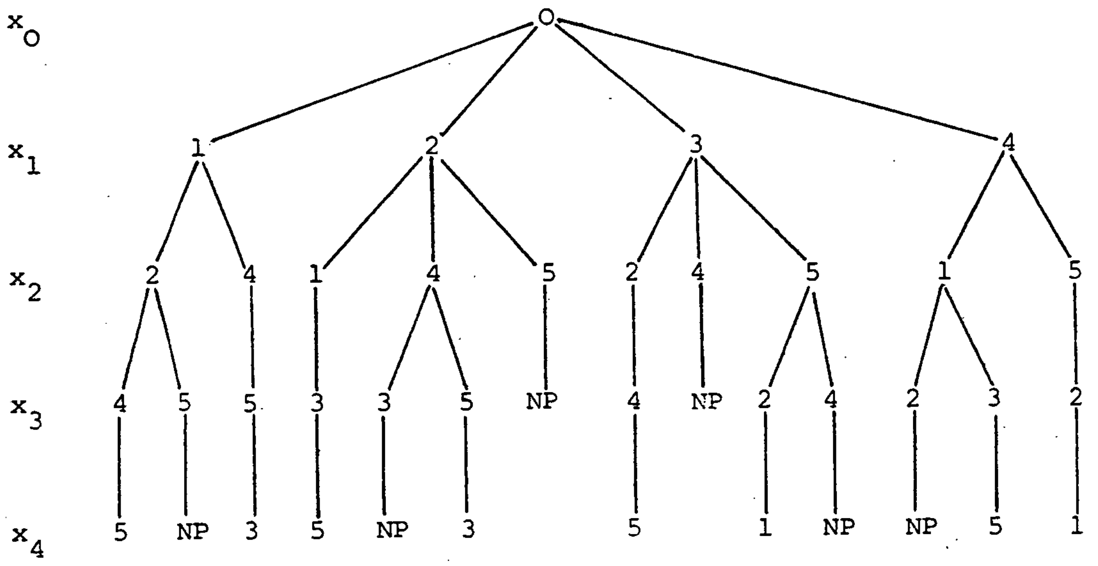
$$\alpha = \begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 3 & 5 & 2 \end{matrix}$$

FIGURE 2

Tree of choices produced by the double-line algorithm

when extending the array 0 0 0 0 0 .
0 1 3 5 2

(NP on a branch means that it is not possible to go any further.)



which generate α(0,1) designs. Thus, the double-line algorithm has yielded the following third rows (taking the negatives of the numbers on these completed branches): 0 5 4 2 1; 0 5 2 1 3; 0 4 5 3 1; 0 4 2 1 3; 0 3 4 2 1; 0 3 1 4 5; 0 2 5 3 1; and 0 2 1 4 5.

A rough measure of the efficacy of the algorithm can be had by comparing the number of Steps that are involved here with the number that would be involved in a complete tree search without any constraints on the choice of the individual numbers x_u. (That is, following the practice we suggested at the very beginning of this Section 5.2 (page 214), and testing whether the new design has no double lines only after a whole new row has been selected.) The number of Steps is, in fact, the number of line segments in the tree, including those segments which end with NP. So the total number of Steps involved in the above example is 40. On the other hand, in the complete tree, each node at each level would have six lines emanating from it downwards, and so the total number of lines would be 6 + 36 + 216 + 1296 = 1554. Even if we imposed the very obvious constraint on the complete tree that only x_0 is zero, there would still be 5 + 25 + 125 + 625 = 780 line segments.

Finally in connection with the double-line algorithm, we establish explicitly that it is reliable (that is, that each extension it yields will generate an $\alpha(0,1)$ design), and that it is exhaustive (that is, that it yields all extensions which generate an $\alpha(0,1)$ design).

The first of these results - that the algorithm is reliable - is obviously guaranteed by the theory which we presented at the outset of this Section 5.2, and which underlies its mechanism.

The second result, too, is already implicit in what we have said earlier. Any extension of the generating array α to an array α' which generates an $(r + 1)$-replicate $\alpha(0,1)$ design corresponds uniquely to a set $\{x_1, x_2, \ldots\ldots, x_{k-1}\}$. (That is, we can assume that the array α' is in standard form, and then x_u is defined to be $\dot{-}\alpha'_{ru}$ for $1 \leq u \leq k-1$.) We must show that every such set occurs as one of the completed branches of the tree. To show this, it is sufficient to show that at Step (n) (having already chosen the numbers $x_1, \ldots\ldots, x_{n-1}$), the number x_n is one of those which do not lie in the set:

$$\bigcup_{u'=0}^{n-1} D_{u'n}^{(n-1)}$$

(thus ensuring that at Step (n)(a), x_n is available as a choice outside this set). If this did not hold, then x_n would be contained in one of the sets $D_{u'n}^{(n-1)}$ for some u' in the range $0 \leq u' \leq n-1$. Now, by its definition, $D_{u'n}^{(n-1)}$ equals $\{\alpha_{mu'} \dot{-} \alpha_{mn} \dot{+} x_{u'}: 0 \leq m \leq r-1\}$. So, x_n would equal $\alpha_{mu'} \dot{-} \alpha_{mn} \dot{+} x_{u'}$ for some m. But then $\alpha_{ru'} \dot{-} \alpha_{rn}$, which equals $x_n \dot{-} x_{u'}$, would equal $\alpha_{mu'} \dot{-} \alpha_{mn}$, and this would contradict the hypothesis that the array α' generates a design with no multiple lines. (This hypothesis implies that the entries of the array α' satisfy the condition (4.1) on page 121.)

5.3 Algorithm using the graphical criterion C_3: the triangle algorithm

The algorithm which we have just described offers a systematic yet fairly rapid way of listing α(0,1) designs. This may, frequently, be enough: some experimenters consider it a sufficient guarantee of efficiency that the design should have no double lines; so they would simply take the first design which the algorithm came up with. However, usually some more refined measure of efficiency will also be required, to distinguish among the α(0,1) designs which are listed by the double-line algorithm. We have suggested already the crudest way of doing this: namely, by calculating for each of them the value of the harmonic mean efficiency factor $\mathbf{A}$. But one of the main themes of this thesis is that graph theory methods can to some extent be used in place of $\mathbf{A}$. (The conjecture on which this is based is stated in 2.5.) So, at the very least, there is no point in calculating $\mathbf{A}$ before invoking rather more of the graphical criteria than simply the $\mathbf{C}_2$-criterion which lay behind the double-line algorithm. Of course, it will always be possible to refer to the $\mathbf{A}$-criterion ultimately (the graphical criteria being, in principle, less sensitive than the $\mathbf{A}$-criterion). Nevertheless, the point of the graphical criteria is that they can serve the function of, as it were, a preliminary sieve: the only designs for which it is necessary to go through the laborious business of calculating the value of $\mathbf{A}$ are those which pass this preliminary test, and are therefore reasonably efficient according to certain of the graphical criteria.

In this Section 5.3, therefore, we describe an algorithm - the triangle algorithm - which generates three-replicate α designs by referring to both the $\mathbf{C}_2$-criterion and the $\mathbf{C}_3$-criterion. But before we enter into its description, there is one small caveat which should be borne in mind. Although this algorithm can be seen as a refinement of the double-line algorithm for three-replicate designs, it should not be taken as entirely superseding it: the double-line algorithm does remain of interest in its own right. Searching for designs with a minimal number of circuits of length two has been a recurrent theme in the literature (usually expressed in terms of searching for designs with a small range of concurrences, and reflecting the widespread faith in the $\underline{\underline{S}}$-criterion - see Section 1.2.2). The double-line algorithm can be interpreted as a contribution to this area of work, and the validity of its assumptions about the desirability of avoiding double lines will therefore be generally acceptable to experimenters. On the other hand, the triangle algorithm requires, besides, the acceptance of a certain conjecture based on the graphical criterion $\underline{\underline{C}}_3$: namely, that the best three-replicate $\alpha(0,1)$ designs will be found among those which have a minimal number of triangles in their variety concurrence graphs. This conjecture, having been first advanced only in this thesis (Section 2.5), has not been as fully investigated and discussed as the one concerning double-lines; in consequence, and until this new conjecture has been more thoroughly tested, the triangle algorithm should perhaps be treated with rather more circumspection than the double-line algorithm.

One way of using the $\underline{\underline{C}}_3$-criterion would be to impose it on the double-line algorithm at the points where we proposed earlier to calculate $\underline{\underline{A}}$. That is, for each of the $\alpha(0,1)$ designs produced by the algorithm, we would calculate the number of triangles in the variety concurrence graph by using the theory in 4.2.1. Then we would reject those designs for which this number was greater than a certain threshold. Only in order to distinguish among the designs which remained would we calculate the value of the harmonic mean efficiency factor $\underline{\underline{A}}$.[†]

This, in itself, does, of course, enable some savings in

[†] We will refer to this as the modified double-line algorithm.

computer time. We could, for example, set the threshold initially at the number of triangles in the first design to be found by the double-line algorithm, and subsequently at the minimum number of triangles in the designs listed so far. Adopting this practice can reduce by a useful amount the number of designs for which the value of $\underline{\underline{A}}$ would have to be calculated: for instance, in looking for extensions of the array:

$$\alpha = \begin{matrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 5 \end{matrix}$$

(with s = 8) to three replicates, this approach reduces the number of designs listed from 76 to 11.

However, this is not the best method of invoking the $\underline{\underline{C}}_3$-criterion, for it makes savings only at the very end of the algorithm: that is, only in the calculation of the values of the harmonic mean efficiency factor after the new rows have been selected. It does not modify that part of the algorithm which is actually concerned with this selection: it merely rejects those choices which have, in some defined sense, too many triangles. Thus, this method combines the $\underline{\underline{C}}_3$-criterion and the double-line algorithm in a way that is closely analogous to the initial proposal we made (page 214) for combining the $\underline{\underline{C}}_2$-criterion and a complete tree search. So the drawbacks of that earlier proposal are present here also: both suggestions fail to incorporate the criteria they invoke into the basic mechanism of choice in the underlying tree searches.

Just as we went on to show in Section 5.2 that more subtle use could be made of the $\underline{\underline{C}}_2$-criterion, so, here, we now explain how more subtle use can be made of the $\underline{\underline{C}}_3$-criterion. That is, we show how the theory we developed earlier (Section 4.2.1) for enumerating triangles can be used as the essential mechanism of an algorithm. Instead of invoking the $\underline{\underline{C}}_3$-criterion only after a whole new row has been selected, we refer to it each time we choose an element of the row - in much the same way as we referred to the $\underline{\underline{C}}_2$-criterion in the double-line algorithm.

Here, the specific piece of theory which we use is the formula (4.17) for enumerating triangles in a three-replicate $\alpha(0,1)$ design. The only variable part of that formula is the number of three-replicate triangles. So, in order to find designs with a minimal number of triangles, it will be sufficient to look for those with a minimal number T_3 of triangles of this type. In other words, we will look for designs which have a minimal value of the quantity

$$T_3 \;=\; 6s \sum_{j=0}^{k-1} \sum_{x=1}^{s-1} h_{1j}(x)\, h_{2j}(x) \ .$$

Now, that it is possible to incorporate this aim into an algorithm will turn out to be a consequence of the theory which underlay our derivation of this formula. It arose from the arrays G_1 and G_2 which we defined on page 129; and the sum

$$\sum_{j=0}^{k-1} \sum_{x=1}^{s-1} h_{1j}(x)\, h_{2j}(x) \tag{5.2}$$

(which we called $t_3(0,\ 1,\ 2)$) is the number of equalities of the form (4.14) between off-diagonal elements in the same rows of G_1 and G_2. That is, it is the number of equalities of the form:

$$\alpha_{2i} \mathbin{\dot{-}} \alpha_{2j} \;=\; \alpha_{1i} \mathbin{\dot{-}} \alpha_{1h} \tag{5.3}$$

(where, if the design has no double lines, it is sufficient, as we noted on page 130, to stipulate that the suffices i, j, and h are not all the same). The algorithm we develop here will add a third row to a two-replicate $\alpha(0,1)$ design in such a way as to give rise to as few such equalities as possible. It will be the forms of the two arrays G_1 and G_2 themselves which will enable us to achieve this in a straightforward and useful fashion.

Thus far, then, the theory for this algorithm parallels closely that for the double-line algorithm in 5.2. In practice, however, there is, here, an important new complication. Previously, we were able to eliminate double lines altogether in most designs of interest; so the double-line algorithm was constructed so as to yield designs with no double lines at all. Here, on the other hand, it is by no means always possible to eliminate completely three-replicate triangles: for many values of s and k which might well arise in real experiments, the minimum attainable number of such triangles is greater than 0. (For example, for $s = 8$ and $k = 4$, the minimum is 96; that is, $t_3(0, 1, 2)$ cannot be made smaller than 2.)

To solve this complication on the level of theory would be very difficult: indeed, it seems unlikely that combinatorial arguments alone could, in general, predict just how many three-replicate triangles there must be. So our approach will be rather more heuristic. The details of how we deal with the complication cannot be thoroughly explained until we have described the algorithm more fully . However, it will be useful if we indicate here at the outset the general approach which we will adopt to getting round it.

Essentially, what we do is extend the amount of searching which the algorithm has to conduct. However, we do not do this indiscriminately: we use some theory as well. To start out with, the algorithm constructs an initial design by some determinate rule (such as, keeping to a minimum the number of new equalities of the type (5.3) which are created in turn by each new element of the third row). The value of $t_3(0, 1, 2)$ for this design (that is, the total number of equalities of the type (5.3)) then becomes a threshold in the search for the next design: a new third row is built up element by element with the restriction that the number of equalities of the type (5.3) should never exceed this threshold. Then, in searching for a third new row, the threshold is the value of the quantity $t_3(0, 1, 2)$ for this second one, and so on.

In principle, this plan might appear rather clumsy. However, it turns out, in practice, to permit some considerable savings over the search based on the double-line algorithm. We return to this point at the end of this Section (page 248). Before that, we will now describe in detail how the theory from Section 4.2.1, along with the general points we have just outlined, can be used to develop an algorithm.

The function of the algorithm will be to add a third replicate to a two-replicate $\alpha(0,1)$ design. It will do this by adding a third row to the two rows of the generating array α of the two-replicate design, and it will form this third row by adding elements from the left (starting always with $\alpha_{20} = 0$). The central point of the discussion which follows is to devise rules for choosing these elements. These rules will ensure that two conditions are satisfied: first, that each design constructed in this fashion has no double lines, and second that it has no more equalities of the type (5.3) than the threshold that has been set by the previous designs listed.

Now, the first of these conditions is satisfied if there are no equations of the form (5.1) among elements of the extended generating array. For this to be the case, it will be sufficient for the following to hold, since the two-replicate design which we start out with is postulated to have no double lines:

$$\left.\begin{array}{ll} \text{(i)} & \alpha_{2i} \dot{-} \alpha_{2j} \neq 0 \\ \text{(ii)} & \alpha_{2i} \dot{-} \alpha_{2j} \neq \alpha_{1i} \dot{-} \alpha_{1j} \end{array}\right\} \text{ for } i \neq j \, .$$

The second of the conditions requires the calculation of the number of equalities like (5.3). We will concentrate initially on devising a straightforward way of counting these equalities which will be based on the array G_1 and a series of modified forms of it. Subsequently, we will show how the first condition, in its two parts, can be incorporated into this process in a very easy fashion.

As with the exposition in 5.2 for the double-line algorithm, it will clarify matters here if we continually refer the theory back to a particular example. The one we choose has block size 4 and has 8 blocks in each replicate, and the initial two-replicate design is generated by the array:

$$\alpha = \begin{matrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 4 \end{matrix} \, .$$

Then the array G_1 is in this case:

	0	1	2	4
0	0	7	6	4
1	1	0	7	5
2	2	1	0	6
4	4	3	2	0

.

First of all, then, how to define rules for choosing the elements of the third row which are related to the number of equalities of the type (5.3) which are created? Choosing an element α_{2u} can give rise to equalities of this type along with previous elements of the third row in two different ways: with i = u or with j = u. The number of such equalities with i = u is the number of times α_{2u} occurs in the set:

$$S_{1u} = \{\alpha_{2j} \dotplus \alpha_{1u} \dot{-} \alpha_{1h} : 0 \leq h \leq k-1, \text{ and } 0 \leq j \leq u-1\}$$

($j \leq u-1$ ensuring that i, j, and h are not all the same); and the number of such equalities with j = u is the number of times α_{2u} occurs in the sets:

$$S_{2ui} = \{\alpha_{2i} \dot{-} \alpha_{1i} \dotplus \alpha_{1h} : 0 \leq h \leq k-1\} ,$$

for $0 \leq i \leq u-1$ ($i \leq u-1$ ensuring that i, u, and h are not all the same).

The calculation of these numbers will be facilitated by certain modifications to the array G_1. After having chosen the new elements $\alpha_{21}, \ldots\ldots, \alpha_{2,u-1}$ of the extended generating array α, we construct an array which we will call $G_1^{(u)}$: it has u rows; row i ($0 \leq i \leq u-1$) is formed by subtracting the row i of the array G_1 from the quantity α_{2i}; that is, it is the set S_{2ui}; then row u of $G_1^{(u)}$ consists of the uk elements of the set S_{1u}.

To illustrate this construction with reference to our example, suppose we have chosen $\alpha_{21} = 3$. So $u = 2$. Then the first two rows of $G_1^{(2)}$ are:

$$\begin{array}{cccc} \alpha_{20} \dot{-} 0 & \alpha_{20} \dot{-} 7 & \alpha_{20} \dot{-} 6 & \alpha_{20} \dot{-} 4 \\ \alpha_{21} \dot{-} 1 & \alpha_{21} \dot{-} 0 & \alpha_{21} \dot{-} 7 & \alpha_{21} \dot{-} 5 \end{array} .$$

That is, these first two rows are:

$$\begin{array}{cccc} 0 & 1 & 2 & 4 \\ 2 & 3 & 4 & 6 \end{array} .$$

The the third row of $G_1^{(2)}$ consists of the numbers:

$$\begin{array}{cccc} \alpha_{20} \dot{+} 2 & \alpha_{20} \dot{+} 1 & \alpha_{20} \dot{+} 0 & \alpha_{20} \dot{+} 6 \\ \alpha_{21} \dot{+} 2 & \alpha_{21} \dot{+} 1 & \alpha_{21} \dot{+} 0 & \alpha_{21} \dot{+} 6 \end{array} .$$

That is, this row is:

$$2 \quad 1 \quad 0 \quad 6 \quad 5 \quad 4 \quad 3 \quad 1 .$$

So the array $G_1^{(2)}$ is:

$$\begin{array}{cccccccc} 0 & 1 & 2 & 4 & & & & \\ 2 & 3 & 4 & 6 & & & & \\ 2 & 1 & 0 & 6 & 5 & 4 & 3 & 1 \end{array} .$$

It follows immediately that the number of new equalities of the type (5.3) which are created by choosing α_{2u} equal to some x is precisely the number of times x occurs in this array $G_1^{(u)}$. In our example, for instance, if we chose $\alpha_{22} = 5$, then one new such equality is created, since 5 occurs once in the array $G_1^{(2)}$. This equality has $i = 2$, $j = 1$, and $h = 0$: that is, it is:

$$\alpha_{22} \dot{-} \alpha_{21} = \alpha_{12} \dot{-} \alpha_{10} .$$

There are two further points to be made about this process. The first is that the arrays $G_1^{(u)}$ can be formed sequentially, since rows $0, 1, \ldots\ldots, u-1$ in the array $G_1^{(u+1)}$ are the same as these rows in the array $G_1^{(u)}$. The second point concerns the first condition which we temporarily left aside on page 236: that is, we show, finally, how double lines can be avoided by means of certain simple rules based on the arrays $G_1^{(u)}$. Double lines will be avoided if the requirements (i) and (ii) on page 236 are adhered to. The first of these stipulates, simply, that the numbers α_{2i} be distinct; this is easy to ensure. The second will hold if we choose each element α_{2u} in such a way that for each $i \leq u$:

$$\alpha_{2i} \dot{-} \alpha_{2u} \neq \alpha_{1i} \dot{-} \alpha_{1u}.$$

That is, if:

$$\alpha_{2u} \neq \alpha_{2i} \dot{-} (\alpha_{1i} \dot{-} \alpha_{1u}).$$

In other words, α_{2u} must be chosen so that it does not equal any of the elements in the column u of the array $G_1^{(u)}$.

For example, in our illustration, we must choose α_{22} so that, first, it does not equal α_{20} or α_{21}; that is, so that it does not equal 0 or 3; and, second, so that it does not equal any of the elements in column two of the array $G_1^{(2)}$; that is, so that it does not equal 2, 4, or 0. Thus, to ensure that there will be no double lines, we have to choose α_{22} to be in the set {1, 5, 6, 7} .

Bringing together all these points, we can now describe the Steps in the triangle algorithm. We will subsequently illustrate them in detail with reference to our example.

That the theory we have presented does provide a reasonable basis for an algorithm is due to the fact that it defines conditions on the element α_{2u} which depend only on the given array that is to be extended and on the elements α_{2j} $(0 \leq j \leq u-1)$ which have already been selected: it is this which allows the theory to be used to define Steps for selecting these elements sequentially.

Step (O)

Initially, we set $\alpha_{20} = 0$. We also have a threshold number of equalities of the type (5.3), which we denote by η. (As we explained earlier, the threshold is the minimum number of such equalities in the designs listed so far; at the beginning of the algorithm, when no designs have yet been listed, we set $\eta = \infty$.)

Step (1):

(a) Form the array $G_1^{(1)}$.

(b) Choose α_{21} according to the rules:

(1) α_{21} must not equal α_{20};

(2) α_{21} must not lie in the column 1 of $G_1^{(1)}$;

(3) the number of times (n_1) which α_{21} occurs in $G_1^{(1)}$ must not exceed the threshold η.

Step (2):

(a) Form the array $G_1^{(2)}$.

(b) Choose α_{22} according to the rules:

(1) α_{22} must not equal α_{20} or α_{21};

(2) α_{22} must not lie in the column 2 of $G_1^{(2)}$;

(3) the number of times (n_2) which α_{22} occurs in $G_1^{(2)}$ must not exceed $\eta - n_1$.

In general, for $1 \leq u \leq k-1$, Step (u):

(a) Form the array $G_1^{(u)}$.

(b) Choose α_{2u} according to the rules:

(1) α_{2u} must not equal $\alpha_{20}, \alpha_{21}, \ldots\ldots, \alpha_{2,u-1}$;

(2) α_{2u} must not lie in the column u of $G_1^{(u)}$;

(3) the number of times (n_u) which α_{2u} occurs in $G_1^{(u)}$ must not exceed:

$$\eta - \sum_{i=1}^{u-1} n_i .$$

Incorporating these rules for choosing the numbers α_{2u} into a tree search will, broadly, follow the practice we adopted for the double-line algorithm: that is, each branch of the tree will correspond to a set of k - 1 elements α_{21}, α_{22},....., $\alpha_{2,k-1}$, and a branch will be terminated either when a complete set is constructed (in which case the algorithm has found another new row), or else when the constraints (b) prevent any further elements being chosen. The difference with this algorithm, compared to the double-line algorithm, concerns the threshold η which arises in constraint (3). We propose that the order in which the algorithm goes through the various choices available for α_{2u} after the three constraints have been imposed will be according to increasing values of the number n_u of new equalities like (5.3) which are created. The advantage of choosing in this way is that it can often lead to more rapid revision of the threshold η, and hence to further restriction of the amount of searching that needs to be carried out. (We will illustrate this point below, page 245.)

This general method will be clarified if we work through its application to the example we have been referring to throughout this Section 5.3. That is, we want to extend to a three-replicate α(0,1) design the two-replicate α(0,1) design which is generated by the array:

$$\alpha = \begin{matrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 4 \end{matrix},$$

with s = 8.

First of all, we fix $\alpha_{20} = 0$. Initially, the threshold η is set at ∞. We have also that the array G_1 is:

	0	1	2	4
0	0	7	6	4
1	1	0	7	5
2	2	1	0	6
4	4	3	2	0

In searching for the first extension, the threshold does not really operate, being larger than any possible number of equalities that could arise. Below (page 244), we illustrate the operation of the threshold in looking for subsequent extensions.

Step (1)

(a) Form the array $G_1^{(1)}$ (as described on page 237):

$$\begin{matrix} 0 & 1 & 2 & 4 \\ 1 & 0 & 7 & 5 \end{matrix}.$$

(b) Choose α_{21} according to the rules:

(1) α_{21} must not equal α_{20}; that is, it must not equal 0.

(2) α_{21} must not lie in the column 1 of $G_1^{(1)}$; that is, it must not equal 1 or 0.

So we have to choose α_{21} in the set {2, 3, 4, 5, 6, 7} . Letting α_{21} be 3 or 6 creates no equalities of the form (5.3), whereas letting it equal 2, 4, 5, or 7 creates one such equality. Adopting the principle that each α_{2u} should be chosen (in the set defined by the constraints (b)) so as to minimise the number of new equalities of the form (5.3) that are created, we will let $\alpha_{21} = 3$.

Step (2)

(a) Form the array $G_1^{(2)}$:

$$\begin{array}{cccccccc} 0 & 1 & 2 & 4 & & & & \\ 2 & 3 & 4 & 6 & & & & \\ 2 & 1 & 0 & 6 & 5 & 4 & 3 & 1 \; . \end{array}$$

(b) Choose α_{22} so that:

(1) it does not equal α_{20} or α_{21}; that is, so that it does not equal 0 or 3;

(2) it does not lie in the column 2 of $G_1^{(2)}$; that is, so that it does not equal 2, 4, or 0.

So we have to choose α_{22} to lie in the set $\{1, 5, 6, 7\}$. The number 1 occurs three times in $G_1^{(2)}$, 6 occurs twice, 5 occurs once, and 7 does not occur at all. So we choose $\alpha_{22} = 7$.

Step (3)

(a) Form the array $G_1^{(3)}$:

$$\begin{array}{cccccccccccc} 0 & 1 & 2 & 4 & & & & & & & & \\ 2 & 3 & 4 & 6 & & & & & & & & \\ 5 & 6 & 7 & 1 & & & & & & & & \\ 4 & 3 & 2 & 0 & 7 & 6 & 5 & 3 & 3 & 2 & 1 & 7 \; . \end{array}$$

(b) Choose α_{23} so that:

(1) it does not equal α_{20}, α_{21}, or α_{22}; that is, so that it does not equal 0, 3, or 7.

(2) it does not lie in the column 3 of $G_1^{(3)}$; that is, so that it does not equal 4, 6, 1, or 0.

So we have to choose α_{23} to lie in the set $\{2, 5\}$. The choice $\alpha_{23} = 2$ creates four equalities of the form (5.3) (since 2 occurs four times in the array $G_1^{(3)}$). The choice 5, on the other hand, creates only two such equalities. So we let $\alpha_{23} = 5$.

So we have the first extension 0 3 7 5, which gives rise to $t_3(0, 1, 2) = 2$ equalities of type (5.3). So we now set the threshold η to be 2; henceforth in the algorithm, we will be interested only in those choices of the numbers α_{2u} which create no more than two equalities.

Having listed this extension, the algorithm then returns (or "backtracks" in computing terminology) to Step (2), and takes the next available choice for α_{22}. The remaining set at this Step is {1, 5, 6}. However, now that we have re-set the threshold to be 2, the choice 1 (which would create three equalities of the type (5.3)) is prohibited. Of the other two elements in this set, 5 creates the smaller number of equalities, and so we next let $\alpha_{22} = 5$. (This creates one equality, which, being less than the threshold, 2, is acceptable; that is, $n_2 = 1$.)

Then we go on to a new Step (3):

(a) Form the array $G_1^{(3)}$:

$$\begin{array}{cccccccccccc}
0 & 1 & 2 & 4 & & & & & & & & \\
2 & 3 & 4 & 6 & & & & & & & & \\
3 & 4 & 5 & 7 & & & & & & & & \\
4 & 3 & 2 & 0 & 7 & 6 & 5 & 3 & 1 & 0 & 7 & 5 \ .
\end{array}$$

(b) Choose α_{23} so that:

(1) it does not equal α_{20}, α_{21}, or α_{22}; that is, so that it does not equal 0, 3, or 5.

(2) it does not lie in the column 3 of $G_1^{(3)}$; that is, so that it does not equal 4, 6, 7, or 0.

So we have to choose α_{23} to lie in the set {1, 2}. The number

1 occurs twice in the array $G_1^{(3)}$, and 2 occurs three times. So, to choose α_{23} equal to 1 would create 3 equalities of type (5.3) altogether (including the $n_2 = 1$ equality which arose at Step (2)); and to choose α_{23} equal to 2 would create 4 equalities altogether. Since both of these choices therefore create more equalities than the threshold 2, we have to terminate this branch of the tree without having found any admissible third rows.

(Notice how this example illustrates the usefulness of going through the choices for α_{23} according to increasing numbers of new equalities that are created. If, for example, we had chosen $\alpha_{22} = 5$ the first time round - instead of $\alpha_{22} = 7$ - then, working through the Step (3) that would have ensued, the best third row we could have found would have been 0 3 5 1, which creates altogether $t_3(0, 1, 2) = 3$ equalities of type (5.3). So the threshold would have been set after this first extension at $\eta = 3$, which, being less restrictive than $\eta = 2$, would have left us with more searching to go through subsequently. Eventually, of course (as we show below, page 247), the algorithm would have set the threshold at $\eta = 2$; but the point is that it would have taken longer to get there, and would have listed a number of designs that were not $\underline{\underline{C}}_3$-optimal in the meantime.)

Pursuing this argument for each of the possibilities, we would list the following third rows:

0 3 7 5; 0 6 3 7; 0 2 6 7; 0 7 4 5.

Each of these has $t_3(0, 1, 2) = 2$ equalities of the type (5.3). In a complete listing of all possible extensions of the initial two-replicate array to a three-replicate $\alpha(0,1)$ design (for which we can use the double-line algorithm), these four are the only ones with values of $t_3(0, 1, 2)$ equal to 2, and there are none with values less than two. Moreover, these four include (as our conjecture in 2.5 would lead us to expect) the best extensions according to the $\underline{\underline{A}}$-criterion as well: the first two have $\underline{\underline{A}} = .70954931$, and the other two have $\underline{\underline{A}} = .70931253$.

That each design listed has no double lines is guaranteed by the theory we presented earlier (specifically, constraints (1) and (2) at each Step (u)(b)). We must show also, finally, that the algorithm will list every third row which generates a design which has no double lines and has a minimal number of triangles. (That is, a minimal number of equalities of the type (5.3).) So, suppose the numbers $\alpha_{20}, \alpha_{21}, \ldots\ldots, \alpha_{2,k-1}$ constitute such a third row. To show that this design will be found by the algorithm, it is sufficient to show that this set of numbers occurs as a completed branch of the tree. And to demonstrate this, we have to show that for each $u \geq 1$, once the numbers $\alpha_{20}, \ldots\ldots, \alpha_{2,u-1}$ have been chosen, the number α_{2u} will be among those available at the next Step. So we have to show that this number α_{2u} is not excluded by any of the restrictions (b) at Step (u).

It is not excluded by (b)(1), since the premise that this row generates a design with no double lines guarantees that α_{2u} does not equal any of $\alpha_{20}, \ldots\ldots, \alpha_{2,u-1}$.

If α_{2u} were excluded by constraint (b)(2), then it would lie in column u of array $G_1^{(u)}$, which consists of the numbers

$$\alpha_{2i} \dot{-} (\alpha_{1i} \dot{-} \alpha_{1u})$$

for $0 \leq i \leq u-1$, and 0 (in row u). Again because the third row yields a design with no double lines, α_{2u} cannot lie in this set. So α_{2u} cannot be excluded by (b)(2).[†]

If α_{2u} were excluded by (b)(3), then the number of times, n_u, that α_{2u} occurs in the array $G_1^{(u)}$ would exceed $\eta - \sum_{i=1}^{u-1} n_i$. But this would imply that $\eta < \sum_{i=1}^{k-1} n_i$, which sum equals the value of $t_3(0, 1, 2)$ for this design. In other words, this design would have strictly more than the threshold number of equalities of the form (5.3). But this is impossible, since the design is postulated to have a minimal number $t_3(0, 1, 2)$ of such equalities, and each value which η assumes (apart from its initial value ∞) actually occurs as the number of equalities in some design that has been listed. So α_{2u} cannot be excluded by constraint (b)(3).

This completes the proof that the algorithm is exhaustive; that is, that it will list every extension which generates a design with no double lines and a minimal number of triangles.

Of course, the algorithm might list other new rows as well: for example, changing slightly the example we have discussed, if we use the algorithm to list extensions of the array

$$\begin{matrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 5 \end{matrix}$$

(with s = 8), then it gives six rows, the first two of which have

[†] This point shows how constraint (b)(2) operates in essentially the same way as the mechanism of the double-line algorithm, and so makes it possible to describe this triangle algorithm as a refinement of that earlier one - as we did on page 211.

3 equalities like (5.3), and the remaining four have 2. That this sort of thing happens is one of the weaknesses of our theory, and is due to the rather heuristic approach which we have adopted to finding the minimal number of triangles that must occur. If, in contrast, we could devise a way of predicting exactly what this minimum will be, then we could eliminate superfluous listing such as this: we would, simply, set the threshold η to this minimum at the beginning and leave it at that value throughout.

However, in the absence of being able to predict this, we have been forced to adopt the practice we have outlined: that is, of revising the threshold η after each new design is found. The mechanism of the rest of the algorithm will, we hope, lead fairly rapidly, in general, to η being reduced to the minimum.

In any case, the algorithm as it stands does seem to offer useful savings in time over all the other methods which we have proposed. Table 5 (page 249) gives some examples of timings (in seconds) on the EMAS 2980 computer at Edinburgh University. The comparison we make here is between the time taken by the triangle algorithm (column 4), and the time taken by the modified double-line algorithm (column 5). (See footnote on page 231.) Columns one and two give the values of s and k respectively, and column 3 gives the second row of the array which is to be extended to three rows. (The first row of this array is, as always, taken to consist entirely of zeroes.)

That most of the savings shown by this Table are due to the more efficient mechanism of the triangle algorithm can be illustrated by comparing also the time taken by the full, un-modified, double-line algorithm, as expounded in 5.2. For instance, in the case of the last example in the Table, the full double-line algorithm takes 12.60 seconds, and so the simple modification we proposed on page 231 saves only 1.18 seconds, in contrast to the triangle algorithm which saves 11.17 seconds.

TABLE 5

Comparison of computer timings (in seconds) for triangle algorithm and modified double-line algorithm

s	k	Array to be extended	Triangle algorithm	Modified double-line algorithm
30	4	0 1 2 5	10.26	158.12
20	7	0 1 3 7 11 19 13	87.65	> 7200
20	6	0 1 2 5 7 11	3.16	6904.58
12	6	0 1 2 5 7 11	11.94	109.36
12	6	0 1 3 5 7 9	18.52	109.21
11	5	0 1 3 5 2	1.43	11.42

Of course, the speed of operation of the triangle algorithm is of more than purely formal interest. The whole purpose of the algorithms we have described in this Chapter is to help in the search for efficient α designs by reducing the amount of fruitless searching that has to be carried out. Because it makes significant gains in this respect, the triangle algorithm allows, for example, a complete search of all reasonably good three-replicate designs for the values of s and k in the ARCUS catalogue.

We have used the algorithm to search for extensions of pairs of rows 0 and 1 as given in the catalogue: that is, in each case to search for a third row which generates a design that has a higher value of the harmonic mean efficiency factor than the one listed. In twenty cases (out of the forty-nine in the catalogue), we have been able to find a better third row by this method: the absolute improvements range from .00004463 to .00168974, and the relative improvements from .00559061% to .21527135%. For full details of this, see Appendix 2.

5.4 Algorithm using the graphical criterion $\underline{\underline{C}}_4$: the square algorithm

The triangle algorithm which we presented in the last Section is restricted to designs in three replicates. In principle, it could be extended - at the expense of much greater complexity - to constructing designs with more than three replicates, using the theory we expounded in 4.2.2 for enumerating triangles in general $\alpha(0,1)$ designs. However, the one case where the graphical criterion $\underline{\underline{C}}_3$ cannot be put to use is in helping to construct $\alpha(0,1)$ designs that have only two replicates: as we observed earlier, the number of triangles in these designs is simply a function of s and k, and so the $\underline{\underline{C}}_3$-criterion is of little use in assessing their efficiency. So, to develop a refinement of the double-line algorithm that can be used for constructing efficient two-replicate designs, we have to turn to the next graphical criterion, namely $\underline{\underline{C}}_4$.

In this Section 5.4, we describe an algorithm which invokes that criterion: it constructs two-replicate $\alpha(0,1)$ designs which have a minimal number of squares in their variety concurrence graphs. This algorithm will be called the square algorithm.

The theory which we use comes, again, from Chapter 4: specifically, the formulas for calculating the number of squares in two-replicate $\alpha(0,1)$ designs (which we summarised on pages 170 and 171). In this sense, the square algorithm is analogous to the triangle algorithm: it builds up second rows of the generating array α by using these formulas as a guide in the choice of the individual elements of that row. (The first row is, as always, postulated to consist entirely of zeroes.) There are, however, two important differences in our approach here which distinguish it from the one we adopted in Section 5.3. In the first place, we were not able to predict in advance for the triangle algorithm the minimal number of triangles that would occur. Here, in contrast, we can be more successful in calculating the number of squares which the best designs will contain - by using the material in Section 4.4.3. There, we derived a lower bound L_4 for the number of squares in two-replicate $\alpha(0,1)$ designs, and we observed that in most cases we have looked at, the lower bound is attained by the best designs. On the strength of these observations, it would seem likely to be worthwhile to develop an algorithm that will list only those designs which have this minimum number L_4 of squares - and this is what we do in the square algorithm. In other words, the approach we adopt for this algorithm is less heuristic than the one we adopted for the triangle algorithm.

There is, also, a further important difference between the square algorithm and the triangle algorithm. This difference concerns the assumptions on which they are based. It will be recalled that the triangle algorithm is founded on a conjecture about the desirability of minimising the number of triangles: namely, that the best $\alpha(0,1)$ design will always be amongst those which have a minimal number of triangles. Valid as we would argue this conjecture to be, it is related to the assumptions of the double-line algorithm only through the theory of Chapter 2, and, in consequence, requires (as we mentioned on page 231) to be treated with care until it has been more thoroughly investigated.

The square algorithm, on the other hand, does not require any significant extension of the assumptions which underlay the double-line algorithm. The conjecture on which the square algorithm is based concerns the desirability of minimising the number of squares in a two-replicate $\alpha(0,1)$ design: to be precise, it asserts that the best two-replicate $\alpha(0,1)$ design will always be found amongst those which have a minimal number of squares. Now, this conjecture is (as we observed in 4.4.3) related, through the contraction of the two-replicate design, to a conjecture closely analogous to that which underlay the double-line algorithm: namely, that the best cyclic designs have a minimal number of circuits of length two. The double-line algorithm, it will be recalled, is based on the conjecture that the best α designs have a minimal number of circuits of length two, and both these conjectures are special cases of the widely accepted general conjecture that all $\underline{\underline{A}}$-optimal designs have a minimal number of circuits of length two (- widely accepted in the equivalent form, that is, which asserts that all **A**-optimal designs are **S**-optimal). In this sense, therefore, the square algorithm involves assumptions that are no wider than those underlying the double-line algorithm. Consequently, we would recommend for the square algorithm and two-replicate designs what we were as yet reluctant (page 231) to recommend for the triangle algorithm and three-replicate designs: namely, that the square algorithm should supersede the double-line algorithm (and all other algorithms based on the **S**-criterion) in the search for **A**-optimal two-replicate designs.

Of course, as before with the $\mathbf{C}_3$-criterion (page 231), one way of using the $\mathbf{C}_4$-criterion would be to calculate the number of squares in each two-replicate design listed by the double-line algorithm, and then to discard those designs in which that number was greater than L_4. However, again as before, that is not the best way of using the criterion. Just as, earlier, we employed the formulas for numbers of triangles to devise rules for choosing the elements of the new row one by one, so, here, we do likewise with the formulas for the number of squares. First of all, we describe how the formulas can be used as the basis of the algorithm, and then we illustrate with an example how this works out in practice.

In Section 4.4.3, we found that the total number of squares in a two-replicate $\alpha(0,1)$ design is:

$$\mathbf{C}_4 \quad = \quad 2k^2 s(k-1)^2 + 2s \sum_{x=1}^{s-1} H(x)^2 \ ,$$

where $H(x)$ is the number of times the residue x occurs in the array G_{01} (whose $(j_0, j_1)^{th}$ entry is $\alpha_{1j_0} \dot{-} \alpha_{1j_1}$). We subsequently showed that a design would have the minimum number L_4 of squares whenever these quantities $H(x)$ differed by at most one, and that this would happen if ρ of them had the value $\delta + 1$, and $s - 1 - \rho$ of them had the value δ (ρ and δ being the unique integers satisfying:

$$k(k - 1) \;=\; \delta(s - 1) + \rho, \quad 0 \leq \rho \leq s - 2).$$

That is, the minimum would be attained if ρ of the residues occurred $\delta + 1$ times among the $k(k - 1)$ off-diagonal entries of the array G_{01}, and $s - 1 - \rho$ of them occurred δ times.

These observations form the theoretical background of the square algorithm. Postulating that the first of the two rows of the generating array α is to consist of zeroes, and that the entry α_{10} is also to be zero, this algorithm adds, one by one, the elements α_{1j} $(1 \leq j \leq k{-}1)$ in such a way that the completed row both generates a design with no double lines, and also yields quantities $H(x)$ that differ by at most one.

The first of these requirements is incorporated easily: to avoid double lines, it is necessary and sufficient that the entries α_{1j} $(1 \leq j \leq k{-}1)$ be distinct and non-zero. The second requirement, also, gives rise to constraints operating on the choices of the individual entries α_{1j}. If we denote by $G_{01}^{(j)}$ the $(j + 1)\times(j + 1)$ sub-array of G_{01} determined by the first $j + 1$ entries $\alpha_{10}, \alpha_{11}, \ldots, \alpha_{1j}$ of the new row[†], then necessary and sufficient conditions for the $H(x)$ to differ by at most one are on the one hand that no

[†] In this notation, G_{01} is, of course, the same as $G_{01}^{(k-1)}$.

residue x should appear more than $\delta + 1$ times in any of the subarrays $G_{01}^{(j)}$, and on the other hand that no more than ρ of the residues x should appear as many as $\delta + 1$ times in any of the subarrays.

There is, moreover, one further constraint which can be introduced, in order to reduce the amount of searching that the algorithm must carry out: we can stipulate that the entries of the added row be in ascending order of magnitude from the left. This constraint leads to no loss of thoroughness in the search, since every two-replicate α design is isomorphic to one in which this is satisfied. (See Patterson and Williams (1976b), page 86: permutation of the columns of the generating array yields an isomorphic design.)

In summary, therefore, the square algorithm is essentially a tree search (as were the double-line algorithm and the triangle algorithm), choosing, sequentially, entries α_{1j} of the new row so as to satisfy the following four constraints:

(1) the α_{1j} are distinct and non-zero;

(2) the α_{1j} are in ascending order of magnitude from the left;

(3) no residue x appears more than $\delta + 1$ times in $G_{01}^{(j)}$;

(4) no more than ρ of the residues x appear as many as $\delta + 1$ times in $G_{01}^{(j)}$.

(It will be useful in practice to notice that constraints (1) and (2) together have the consequence that α_{1j} must not exceed $s - k + j$ (for $1 \leq j \leq k-1$), since the $k - 1 - j$ entries to the right of α_{1j}

must be distinct (condition (1)), and greater than α_{1j} (condition (2)).)

Finally in this Section 5.4 we illustrate the operation of the square algorithm by means of an example. Consider the problem of listing efficient two-replicate $\alpha(0,1)$ designs for block size 4 and for 8 blocks in each replicate. First of all, the algorithm calculates the values of δ and ρ: $k(k - 1)$ is 12 and $s - 1$ is 7. So δ is the integer part of $\frac{12}{7}$, which is 1, and ρ is the remainder, which is 5. So the minimum value of the sum $\sum_{x=1}^{s-1} H(x)^2$ is $\rho(\delta + 1)^2 + (s - 1 - \rho)\delta^2 = 22$, and hence the minimum number of squares is $L_4 = 2k^2s(k - 1)^2 + 2s\times 22 = 2656$.

So for a design to have this minimum number of squares, it is necessary that five of the quantities $H(x)$ have the value 2, and that two of them have the value 1. That is, it is necessary that five of the residues occur among the off-diagonal entries of G_{01} twice, and that two of them occur once.

Step (1)

(a) Only constraints (1) and (2) operate at this Step. Choose α_{11} not equal to zero, and, as we noted above, not greater than $s \dot{-} k \dot{+} 1 = 5$. So α_{11} must lie in the set $\{1, 2, 3, 4, 5\}$. As an example, consider what happens when we choose α_{11} to be 1.

(b) Form the array $G_{01}^{(1)}$: it is:

$$\begin{array}{c|cc} & 0 & 1 \\ \hline 0 & 0 & 7 \\ 1 & 1 & 0 \end{array} \quad .$$

Step (2)

(a) To satisfy constraints (1) and (2), α_{12} must be chosen greater than 1, but not greater than $s \dot{-} k \dot{+} 2 = 6$. None of these choices violates constraints (3) and (4). Continuing with the example, we choose $\alpha_{12} = 2$.

(b) Form the array $G_{01}^{(2)}$:

	0	1	2
0	0	7	6
1	1	0	7
2	2	1	0 .

Step (3)

To satisfy the constraints (1) and (2), we must choose α_{13} greater than 2, and not greater than $s \dot{-} k \dot{+} 3 = 7$; that is, in the set $\{3, 4, 5, 6, 7\}$. Not all of these choices satisfy constraints (3) and (4). For example, if we were to choose α_{13} equal to 3, then the residue 1 would appear in the array G_{01} three times. However, the choice $\alpha_{13} = 4$ does satisfy the constraints: G_{01} is then:

	0	1	2	4
0	0	7	6	4
1	1	0	7	5
2	2	1	0	6
4	4	3	2	0 .

The numbers of times the various residues occur among the off-diagonal entries of this array are as follows:

x	1	2	3	4	5	6	7
H(x)	2	2	1	2	1	2	2 .

So, as anticipated, the sum $\sum_{x=1}^{s-1} H(x)^2$ equals 22, and so the design has the minimal number 2656 of squares. (As a matter of fact, this design is the best on the $\underline{\underline{A}}$-criterion as well.)

CHAPTER 6

THE ESTIMATION OF DIFFERENCES BETWEEN VARIETY EFFECTS

6.1 Introduction

In this final Chapter, we return to an aspect of the background of the graphical criteria which we left uninvestigated in Chapter 2. It will be recalled that the introductory, intuitive, motivation for these criteria (which we presented in 2.2) rested on the link between paths in the variety concurrence graph and the precision with which the design estimates the difference between variety effects: we explained how each path joining the varieties i and j gives rise to an unbiased estimator of the difference between their effects, and we suggested on the strength of this that a design with a large number of such paths would tend to provide a lot of information about this difference. This suggestion led us to the graphical criteria. We then went on to establish them on a firmer theoretical basis which was entirely independent, in fact, of their heuristic origins in that suggestion. It is to the validity and implications of the suggestion that we now turn. Of course, precisely because the criteria were subsequently established on a basis that was independent of the suggestion, the material in this Chapter is not directly relevant to what we have been discussing in Chapters 2, 3, 4 and 5. However, the link between paths and the estimation of variety differences is interesting in its own right, providing, as it does, yet another illustration of the use of graph theory in elucidating important features of the design of experiments. Furthermore, it will emerge (at the end of 6.2.2) that our investigations here yield an extra argument in favour of the graphical criteria.

6.2 Paths and variances

6.2.1 The importance of differences

That the capacity of a design to estimate differences between variety effects is important is a consequence, quite simply, of the frequency with which this is one of the principal purposes of the experiment. In trials for the National and Recommended Lists of cereal varieties[†], for example, the experimenter will want to know whether the new varieties are giving yields that are significantly better than the yields of varieties already on the Lists. Testing this will require the separate comparisons of each of the new varieties with each of the old ones; that is, the estimation of differences between the effects of pairs of varieties, one new and one old.

There are two properties required of a design in order for it to provide useful information on differences. Although we will only be concerned with one of them here, they both have interpretations in terms of graph theory.

The first requirement is that each difference should be estimable. Now, as has often been pointed out elsewhere, a necessary and sufficient condition for this is that the design should be connected, in the sense defined by Bose (1947). But, as

[†] See, for example, Patterson and Silvey (1980), page 220.

we have already observed in 1.4.3(a), this is equivalent to requiring that the design graph should be connected in the graph theory sense. We do not pursue this further here; but, as we also mentioned in 1.4.3(a), it seems likely that graph theory techniques could prove useful at this most basic level of ensuring that designs are connected.

What will concern us is the second requirement: namely, that the estimators of the differences between variety effects should have low variance. Our exposition will be largely of a theoretical and explanatory nature, elucidating the way in which the magnitudes of these variances are related to the numbers of paths - and also circuits - in the variety concurrence graph. Nevertheless, out of this will emerge what we will recommend as a practically useful result: we will show, in 6.3.2, that the theory we develop in this Chapter can be used to devise rules for the efficient allocation of control varieties. These rules will be seen to be at least as good as those proposed hitherto, and we will describe an example in which they are actually better.

6.2.2 The variance of differences between variety effects

It is intuitively reasonable, and, moreover, in conformity with accepted statistical practice, to measure the amount of information about a variety difference by the inverse of the variance of its best estimator. What we will show is that these variances are determined by numbers of paths and circuits in the variety concurrence graph. As will become clear, the development is similar to that of Section 2.3, and, indeed, a number of results established there will be of use here.

First, we will outline briefly some standard results on the variances of differences. Recall that the statistical model is:

$$y_{ij} = \mu + \tau_i + \beta_j + e_{ij} , \qquad (6.1)$$

where μ is the overall mean, τ_i is the effect of variety i, β_j is the effect of block j, and the e_{ij} are independent normal variables with expected value 0 and variance σ^2. Then the normal equations for estimating the variety effects $\underline{\tau}$, subject to the side condition $\underline{1}^T\underline{\tau} = 0$, are:

$$C\underline{\hat{\tau}} = \underline{q} , \qquad (6.2)$$

where:

$$C = rI - \frac{1}{k}NN^T.$$

It follows from this that if $\underline{\gamma}$ is the vector of a contrast - that is, if $\underline{1}^T\underline{\gamma} = 0$ - then the quantity $\psi = \underline{\gamma}^T\underline{\tau}$ is estimable, with estimator $\hat{\psi} = \underline{\gamma}^T\underline{\hat{\tau}}$ which has variance $V(\hat{\psi}) = \underline{\gamma}^T C^- \underline{\gamma}\,\sigma^2$. ($C^-$ could be any reflexive generalised inverse of C, but we will continue to assume it is the one we defined in 2.3.1 (page 56): that is, $C^- = (C + rJ)^{-1}$.)†

Now, the difference between the effects of the varieties i and j is $\tau_i - \tau_j$, which we will represent by $\underline{\gamma}_{ij}^T\underline{\tau}$, where $\underline{\gamma}_{ij}$ is the v×1 vector with entry i equal to 1, entry j equal to -1, and all other entries equal to 0. So this is a contrast, is therefore estimable, and in consequence has variance $\underline{\gamma}_{ij}^T\, C^-\, \underline{\gamma}_{ij}\sigma^2$. Expanding

†The theory behind all this can be found in, for example, John (1971).

this quantity, we find that it equals σ^2 multiplied by:

$$V_{ij} = (C^-)_{ii} + (C^-)_{jj} - 2(C^-)_{ij} . \qquad (6.3)$$

Our next step is to express this equation in terms of the variety adjacency matrix A. It was shown in 2.3 that if $C^- = (C + rJ)^{-1}$, then we get the following expression for rC^- in terms of A:

$$rC^- = I + \sum_{n=1}^{\infty} \sum_{h=0}^{n} \frac{1}{k^n} \left(\binom{n}{h} \frac{1}{r^h} A^h - \frac{k^n}{n+1} J \right) .$$

Inserting this in equation (6.3) gives:

$$V_{ij} = \frac{2}{r} + \sum_{n=1}^{\infty} \sum_{h=0}^{n} \frac{1}{k^n} \frac{1}{r^{h+1}} \binom{n}{h} \left((A^h)_{ii} + (A^h)_{jj} - 2(A^h)_{ij} \right) , \qquad (6.4)$$

and so we have indeed derived a relationship between the variance of the difference and numbers of paths and circuits in the variety concurrence graph. Notice, in particular, that the variance $V_{ij}\,\sigma^2$ is smallest when the quantities $(A^h)_{ii} + (A^h)_{jj} - 2(A^h)_{ij}$ - which we will call $q_h(i, j)$ - are smallest. And minimising this quantity is associated with, on the one hand, increasing the number of i - j paths of length h (as we anticipated in the suggestion we made in Section 2.2, and which we referred to again on page 259), and, on the other hand, decreasing the numbers of circuits which start at i and j. This observation indicates a further important aspect of the graphical criteria. Not only are they concerned with some overall, average, quality of the design (in the way the A-criterion is, for example); they also are linked to the precision with which the individual variety differences are estimated. That is, the graphical criteria will tend both to increase the number of i - j paths of length h (as pointed out in 2.2), and also to decrease the numbers of circuits of length h starting at i and j (insofar as they decrease the total number of circuits). In other words, we might expect that minimising the quantity $\underline{\underline{C}}_h$ will tend to minimise all the quantities $q_h(i, j)$, and so to maximise the precision with which the design estimates each of the individual variety differences.

This, then, is a further argument in favour of the graphical criteria, since the criteria currently in use are not linked thus to the variances with which individual variety differences are estimated; they are linked only to some overall average of these variances.

6.3 Applications

We can now use the expression (6.4) to examine some of the questions which arise from the general connection between paths and variances which we outlined in Section 2.2. To be precise, we will investigate the following points:

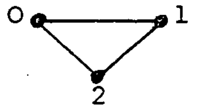
6.3.1 Any i - j path - of whatever length - yields an unbiased estimator of $\tau_i - \tau_j$, and the least squares estimate provided by the design as a whole represents, in a sense, a weighted combination of these paths. What is the relative significance in this least squares estimate of i - j paths of different lengths, and what is the precise role of circuits?

6.3.2 Suppose that the experiment is to incorporate control varieties. The differences between these varieties and all the rest will, therefore, be of especial interest.

How should we allocate the controls in order to maximise the information on these differences?

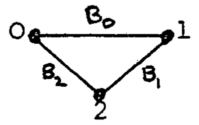
6.3.1 Relative importance of paths of different lengths

First, how relatively important are paths of different lengths? Now, in a general sense, shorter paths give rise to estimators with lower variances. To explain this, we will look at an example. Suppose that the variety concurrence graph contains the following configuration as a subgraph:



,

where the line joining the points 0 and 1 corresponds to concurrence in block B_0, the line joining 1 and 2 to concurrence in block B_1, and the line joining 2 and 0 to concurrence in B_2. Thus the configuration can be depicted as:



.

So, amongst the plots of the design there is one containing the variety 0 in block B_0, having in consequence a yield y_{00}. Similarly there are yields y_{02}, y_{10}, y_{11}, y_{21} and y_{22}. The triangle gives two unbiased estimators, ϕ_0 and ϕ_1, of the difference $\psi = \tau_0 - \tau_1$, since it provides two paths joining the varieties 0 and 1. One of these paths is simply the line 0 - 1, and so, following the development in 2.2, the corresponding estimator is:

$$\phi_0 = y_{10} - y_{00} \quad . \qquad (6.5)$$

The other path consists of the two lines 0 - 2 and 2 - 1, and so the estimator is:

$$\Phi_1 = y_{22} - y_{02} + y_{11} - y_{21} \; . \tag{6.6}$$

The path which yielded the estimator Φ_0 is shorter than the path which yielded Φ_1, and this is directly reflected in the fact that Φ_0 has the smaller variance: since the yields y_{ij} are assumed under the model (6.1) to be uncorrelated, the variances of the two estimators are:

$$\mathrm{Var}(\Phi_0) = 2\sigma^2; \tag{6.7}$$

$$\mathrm{Var}(\Phi_1) = 4\sigma^2. \tag{6.8}$$

In fact, it is straightforward to show in general that an i - j path of length m gives rise to an estimator of $\tau_i - \tau_j$ with variance $2m\sigma^2$(with one important proviso: if any y_{ij} appears more than once with the same sign, then the variance is obviously greater than this)[†].

Of course, this rather simplistic approach to estimating the difference $\tau_i - \tau_j$ obscures some of the complexities of the least squares estimates provided by the design taken as a whole. Most obviously, it ignores the contribution from circuits in equation (6.4). Furthermore, by concentrating on comparing single plots containing the two varieties, it fails to allow for the impact of replication; and it is precisely the purpose of replication to overcome some of the variability of single observations by comparing instead variety means.

[†] It is interesting to note that this result holds even if the model incorporates recovery of interblock information, for, in calculating the variance of the estimators, the appearances of the intrablock correlation σ_b^2 cancel each other out in the same way as do the block effects.

Nevertheless, the considerations exemplified above do suggest why we might expect that it would be desirable for varieties to be joined by paths that are as short as possible.

Now, these heuristic arguments can be given a firmer mathematical foundation by means of equation (6.4). For that equation enables us to evaluate exactly the importance of individual paths and circuits. Thus, the contribution of an i - j path of length h $\geq$ 1 to the variance $V_{ij}\sigma^2$ of the difference $\tau_i - \tau_j$ is the coefficient which multiplies $(A^h)_{ij}$ in (6.4), and this is:

$$\frac{-2k}{r^{h+1}(k-1)^{h+1}}$$

(arguing as in 2.4.1(a)). Similarly, the contribution of a circuit of length h starting at i, or of one starting at j, is:

$$\frac{k}{r^{h+1}(k-1)^{h+1}}{}^{\dagger}.$$

Hence, paths of length h - 1 are r(k - 1) times more important in determining the pattern of the variances than are paths of length h. Likewise for circuits. Of course, this does not mean that paths and circuits of length h cannot be outweighed by paths and circuits of lengths greater than h. Indeed, we will describe later an example in which this happens. (See Example (2b).) Nevertheless,

[†] One important point about this should be emphasised, analogous to a point we made earlier - page 66 - at a similar stage in the development of the graphical criteria. Although we have isolated the contribution to V_{ij} made by paths and circuits of each length h, it is not in general true that the infinite sum on the right-hand side of equation (6.4) can be separated into three components, one each for $(A^h)_{ii}$, $(A^h)_{jj}$, and $(A^h)_{ij}$. The convergence of this sum depends on the way its terms are arranged, as the following example for v = b = r = k = 2 shows:

$$\begin{matrix} 0 & 1 \\ 0 & 1 \end{matrix}$$

the above discussion does suggest that, in a general sense, what matter in determining the pattern of the variances are paths and circuits of shorter lengths. Again, therefore, we have found that important statistical features of the design are reflected in certain easily conceptualised aspects of the variety concurrence graph.

As a preliminary clarification of these observations, we will now consider the first few values of h on the right-hand side of the equation (6.4): h = 0, 1, and 2. Subsequently, we will examine some specific examples.

For h equal to 0, the total contribution is

$$V_{ij}^{(0)} = \sum_{n=1}^{\infty} \frac{1}{k^n} \frac{1}{r} \binom{n}{0} (1 + 1)$$

$$= \frac{2}{r(k - 1)} ,$$

which is invariant over all i and j, and therefore is not particularly informative. However, it does already indicate the importance of the replication number r: $V_{ij}^{(0)}$ decreases as r increases.

For h equal to 1, we get a rather more interesting result. The contribution here is:

$$V_{ij}^{(1)} = \sum_{n=1}^{\infty} \frac{1}{k^n} \frac{1}{r^2} \binom{n}{1} (-2a_{ij})$$

$$= \frac{-2k}{r^2(k - 1)^2} a_{ij} .$$

Thus, $V_{ij}^{(1)}$ varies in a reverse relationship with the number of times that i and j concur. This is exactly what is usually assumed in practice for the variance $V_{ij}\sigma^2$.[†]

The case of h equal to 2 is the first one where circuits start to play a role. Here, the contribution is:

$$V_{ij}^{(2)} = \frac{k}{r^3(k-1)^3}\,((A^2)_{ii} + (A^2)_{jj} - 2(A^2)_{ij}) .$$

If all the diagonal elements $(A^2)_{uu}$ are the same, then, again, the only quantity that matters is the number of i - j paths of length two. This will happen in, for example, a design with no double lines (when all the diagonal elements are r(k - 1)), or in any cyclic design. To illustrate these points more concretely, we will now discuss a number of specific examples.

Example (1)

The first example concerns cyclic designs. One of the interesting properties of cyclic designs is that the number of circuits which start at each of the points is the same. (We have previously noticed this in Chapter 3: page 100. The number of these circuits of length h is given by the quantity d_h on page 94.) So, for a cyclic design, the variance of the variety difference $\tau_i - \tau_j$ is, by equation (6.4),

$$V_{ij} = \frac{2}{r} + 2\sum_{n=1}^{\infty}\sum_{h=0}^{n}\frac{1}{k^n}\,\frac{1}{r^{h+1}}\binom{n}{h}(d_h - (A^h)_{ij}) . \qquad (6.9)$$

For a given cyclic design, the only variable quantities on the right-hand side of the equation (6.9) are the numbers of i - j paths of the different lengths h: circuits, therefore, do not enter

[†] See, for example, Patterson, Williams and Hunter (1978), page 398. Henceforth in this Chapter we will suppress the constant σ^2.

into comparisons between variances of these differences.

The example we will look at is for 10 varieties in 10 blocks of size 4, with initial block {0, 1, 3, 8} . We will be interested in the pattern of variances of differences with the variety 0; that is, the variances of quantities $\tau_j - \tau_0$. The Table 6 below lists in its first column the distinct variances which occur for estimating these differences. In column two are the varieties j which give rise to these variances. In columns three and four are the numbers of j - 0 paths of lenghts, respectively, 1 and 2.

Table 6

Cyclic design for 10 varieties in 10 blocks generated by {0,1,3,8}.

Comparison of variances of differences $\tau_j - \tau_0$ with numbers of j - 0 paths of lengths 1 and 2.

Variance of	Variety	Numbers of j - 0 paths of:	
$\tau_j - \tau_0$	j	length 1	length 2
.5773	5	2	16
.5907	2, 8	2	13
.5935	3, 7	2	12
.6542	1, 9	1	12
.6956	4, 6	0	16

This Table reveals how the pattern of the variances is reflected by the pattern of paths in what can be called a reverse lexicographic fashion. Thus, the pattern of paths of length 1 is the reverse of the pattern of the variances. Then, among cases where there are the same number of j - 0 paths of length 1, the pattern of variances is reflected by the pattern of paths of length 2 (again in reverse order). That the pattern of paths is important in a lexicographic way is the consequence of the fact that paths of length h - 1 count for more than paths of length h. Thus, for example, there are sixteen 4 - 0 paths of length 2, and only twelve 1 - 0 paths of length 2, yet the variance V_{10} is smaller than the variance V_{40}: the four extra 4 - 0 paths of length 2 are not enough to compensate for the fact that there are no 4 - 0 paths of length 1 while there is one 1 - 0 path of length 1; such compensation would require (by the theory on page 267) 1×r(k - 1) = 9 extra paths of length 2.

Examples (2)

However, in general, the number of circuits starting at variety j is not the same for each j. So, if we are interested in comparing the variances $\tau_j - \tau_0$ for a given design, then the equation (6.4) looks like:

$$V_{0j} = \frac{2}{r} + \sum_{n=1}^{\infty} \sum_{h=0}^{n} \frac{1}{k^n} \frac{1}{r^{h+1}} \binom{n}{h} ((A^h)_{00} + (A^h)_{jj} - 2(A^h)_{0j}) \ . \qquad (6.10)$$

For a given design, the quantity which varies in this is $q_h(0, j)$ (which we defined on page 263). Now, since $q_h(0, j)$ is

$$(A^h)_{00} + (A^h)_{jj} - 2(A^h)_{ij} \ ,$$

the numbers of j - 0 paths alone do not fully account for the pattern of the variances of the differences $\tau_j - \tau_0$; circuits of length h starting at j are relevant as well. Nevertheless, for differences involving a fixed variety (here 0), paths are twice as important as circuits. Further, in designs with no double lines, $(A^2)_{jj} = r(k - 1)$ for each j, and so the pattern of the numbers q_2 is the same as the pattern of the numbers of j - 0 paths of length 2 (but in reverse order).

We take these remarks into account in the next two examples, both of which are α designs.

Example (2a)

This is an α(0,1) design for 30 varieties in blocks of size 5, each variety being replicated 4 times. The generating array is:

$$\alpha = \begin{array}{ccccc} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 3 & 2 & 4 \\ 0 & 5 & 2 & 3 & 1 \\ 0 & 4 & 5 & 1 & 2 \end{array} \quad .$$

Then the Table 7 on page 273 records information about paths of lengths 1 and 2; since the design has no double lines, this is adequate to represent the patterns of the quantities q_1 and q_2 (though in reverse order). (In fact, $(A^2)_{ii} = r(k - 1) = 16$ for each i.)

Similar remarks can be made about this Table as we made about the Table 6. Thus, for example, a great deal of the pattern of the variances is reflected lexicographically in the pattern of paths of lengths 1 and 2. However, the patterns of paths are not as subtle as the variances: for example,

$$\mathrm{Var}(\tau_{15} - \tau_0) > \mathrm{Var}(\tau_{24} - \tau_0) ,$$

but 15 and 24 are both joined to 0 by one path of length 1 and

Table 7

α design for 30 varieties generated by

$$\begin{array}{ccccc} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 3 & 2 & 4 \\ 0 & 5 & 2 & 3 & 1 \\ 0 & 4 & 5 & 1 & 2 \end{array}$$

Comparison of variances of differences $\tau_j - \tau_0$ with numbers of j - 0 paths of lengths 1 and 2.

Variance of	Variety	Numbers of j - 0 paths of:	
$\tau_j - \tau_0$	j	length 1	length 2
.60075645	6, 20, 26	1	9
.60336062	11, 19	1	8
.60342262	10, 18	1	8
.60373264	12, 14	1	8
.60435268	25, 28	1	8
.60596478	24	1	7
.60633681	15, 17	1	7
.60863095	7, 21	1	6
.63392857	13	0	12
.63944692	1, 5	0	10
.64062500	3	0	10
.64205109	2, 4	0	9
.64242312	8, 22, 27	0	9
.64502728	9, 23	0	8
.64763145	29	0	7
.64955357	16	0	6

seven paths of length 2. In fact, to match faithfully the pattern of the variances, it is necessary to proceed to the next two values of h. The quantities q_3, it turns out, almost fully reflect the variances. (For example, $q_3(15, 0) = -34$, while $q_3(24, 0) = -40$.) The only exception is the comparison between the pairs {10, 18} and {11, 19} , all of which have q_3 equal to -40. However, the first of these pairs have q_4 equal to 240, and the second have q_4 equal to 228, which serves to distinguish them in a way that reflects the difference in the respective variances lexicographically.

Again, as in Example (1), extra paths of length 2 are, here, never able to compensate for a deficiency of paths of length 1. Thus, for example, there are twelve 13 - 0 paths of length 2, and only six 21 - 0 paths of length 2, yet $\tau_{13} - \tau_0$ has higher variance than $\tau_{21} - \tau_0$: the one 21 - 0 path of length 1 is not outweighed by the six extra 13 - 0 paths of length 2.

In point of fact, this observation is no accident for designs with no double lines: a deficiency of paths of length 1 in such a design can never be made good by paths of length 2 alone. We can establish this by showing that in any design which has no double lines, each pair of varieties are joined by at most $r(k - 1)$ paths of length 2. Suppose that this were not the case for some such design: that is, suppose that varieties i and j were joined by more than $r(k - 1)$ paths of length 2. Now, the valency of the point i is exactly $r(k - 1)$. So some pair of these paths would have to start with the same line. But, then, since these two paths are distinct, they would have to finish with different lines (since each contains only two lines). This would imply that there would be a pair of points joined by two lines, which would violate the condition that the design has no double lines. This contradiction establishes the result.[†]

Although this result does provide yet further evidence of the importance of paths of shorter lengths, it does not establish that a deficiency of paths of length 1 cannot be compensated for collectively by paths of lengths greater than 1. Nevertheless, we have not been able to find an example of a design in which this occurs: that is, in all the examples we have analysed, if there are more i - j lines than there are i' - j' lines, then $\mathrm{Var}(\tau_i - \tau_j)$ is strictly less than $\mathrm{Var}(\tau_{i'} - \tau_{j'})$. Indeed, in nearly every design we have looked at (which includes many (0,1) designs and all (0,1,2) designs in the ARCUS catalogue of α designs), the first value of h for which there is a difference in the quantities q_h reflects the pattern of the variances: that is, if $q_u(i, j)$ equals $q_u(i', j')$ for each $u < h$, and if $q_h(i, j) < q_h(i', j')$, then $\mathrm{Var}(\tau_i - \tau_j) < \mathrm{Var}(\tau_{i'} - \tau_{j'})$, regardless of the relative size of $q_u(i, j)$ and $q_u(i', j')$ for values of u greater than h. We would speculate that this will hold in all "reasonable" cases. (Clearly this speculation is analogous to, though weaker than, the speculation in 2.5 concerning the graphical criteria.)

[†] This result can be extended to show that if for any two points u and v there are not more than e lines joining u and v, then for any two points i and j, there are at most $er^{h-1}(k - 1)^{h-1}$ i - j paths of length h. However, this extension is of limited value because of the importance of circuits in the quantities q_h. Nevertheless, if for each h the number of circuits of length h starting at each i is the same (as is the case in, for example, cyclic designs), then this extension enables us to show that a deficiency of paths of length h cannot be made good by paths of length n alone for any n greater than h. (The proof is by mathematical induction on h.)

However, although we have found this to hold in most cases, we have been able to construct an example of an $\alpha(0,1,2)$ design in which it does not hold. This is the example we describe next.

Example (2b)

The example is of an $\alpha(0,1,2)$ design for 24 varieties in blocks of size 4, each variety being replicated 3 times. The generating array is:

$$\alpha = \begin{array}{cccc} 0 & 0 & 0 & 0 \\ 0 & 5 & 1 & 2 \\ 0 & 0 & 1 & 2 \end{array},$$

and $s = 6$. Obviously this example would not be used in practice: its interest is purely formal, in that it indicates that the speculation on the previous page does not hold for all designs. (The best listed $\alpha(0,1)$ design for these values of v, r, and k does satisfy the speculation.) The variety differences we will be interested in here are $\tau_{21} - \tau_0$ and $\tau_{17} - \tau_0$. Now, $\mathrm{Var}(\tau_{21} - \tau_0) = 1.1334$ and $\mathrm{Var}(\tau_{17} - \tau_0) = 1.1468$, and so $\mathrm{Var}(\tau_{21} - \tau_0) < \mathrm{Var}(\tau_{17} - \tau_0)$. Neither variety 21 nor variety 17 is joined to variety 0 by a line. However, $q_2(21, 0) = 24$ and $q_2(17, 0) = 22$. So $q_2(21, 0) > q_2(17, 0)$, and so this is contrary to the speculation (with $h = 2$).

What in fact happens here is that higher values of h compensate for this excess in $q_2(21, 0)$. The next three values of the quantities q_h are in the following table:

	q_3	q_4	q_5
21	48	478	2016
17	54	524	2590

For each of these values of h, $q_h(21, 0) < q_h(17, 0)$, and indeed the total contribution to the variances of these three quantities is enough to make good the effect of the opposite difference in the values of q_2: to show this, it is (by equation 6.10)) sufficient to show that:

$$\sum_{h=2}^{5} \frac{kq_h(21, 0)}{r^{h+1}(k-1)^{h+1}} < \sum_{h=2}^{5} \frac{kq_h(17, 0)}{r^{h+1}(k-1)^{h+1}} ;$$

that is, that:

$$\sum_{h=2}^{5} \frac{q_h(21, 0)}{9^{h-2}} < \sum_{h=2}^{5} \frac{q_h(17, 0)}{9^{h-2}} ,$$

and that this does hold can be shown by simply calculating each of these sums. (The left-hand side is 38 and the right-hand side is 38.022.)

6.3.2 The incorporation of control varieties

If the experiment involves control varieties, then the differences between them and each of the others will be of particular interest.[†] So it will be desirable that these differences should have low variance. We propose here a set of rules for allocating the controls in such a way as to achieve this aim. These rules will be seen to be in a sense analogous to the graphical criteria of efficiency which we discussed in Chapter 2. They will be based (as were the criteria) on certain features of the variety concurrence graph, and we will indicate how this basis can be used

[†] For discussion of controls in trials for National and Recommended Lists of cereal varieties, see Patterson and Silvey (1980), page 225.

to simplify the application of the rules to α designs: specifically, we will show that for α designs the rules lead to an allocation of controls that is at least as efficient as the allocation recommended in the literature, and we will describe an example in which the new allocation is more efficient than the conventional one.

First of all, we introduce some notation in order to make the argument clearer. Suppose there are to be c controls, which we will refer to collectively as the set K. Then the problem is to find a subset K of the varieties which is such that the variances of differences between a variety in K and a variety outside it are as small as possible. Now, the average of these variances is a scalar multiple of the quantity:

$$\sum_{i\varepsilon K} \sum_{j\notin K} V_{ij} ,$$

which we will for convenience call V(K). So our aim is to determine how to choose the subset K so as to minimise this quantity V(K). By equation (6.4), V(K) equals

$$\frac{2c(v - c)}{r} + \sum_{n=1}^{\infty} \sum_{h=0}^{n} \frac{1}{k^n} \frac{1}{r^{h+1}} \binom{n}{h} \Big(\sum_{i\varepsilon K} \sum_{j\notin K} ((A^h)_{ii} + (A^h)_{jj} - 2(A^h)_{ij})\Big) \tag{6.11}$$

To simplify this, we replace the innermost summation by the symbol y_h; that is, we let

$$y_h = \sum_{i\varepsilon K} \sum_{j\notin K} ((A^h)_{ii} + (A^h)_{jj} - 2(A^h)_{ij}) \tag{6.12}$$

So:

$$V(K) = \frac{2c(v - c)}{r} + \sum_{n=1}^{\infty} \sum_{h=0}^{n} \frac{1}{k^n} \frac{1}{r^{h+1}} \binom{n}{h} y_h .$$

Now, arguing as we have done several times before, the contribution of y_h to this sum is:

$$\frac{k}{r^{h+1}(k - 1)^{h+1}} y_h .$$

In consequence, the terms y_h are of steadily decreasing significance (by a factor of $r(k - 1)$ each time), in the same way as were the numbers of circuits - $\underline{\underline{C}}_h$ - in Chapter 2. So, following the practice we adopted in Chapter 2, we propose here that a reasonable method of minimising the quantity $V(K)$ would be to minimise the quantities y_h sequentially. In other words: we will find first those subsets K which minimise y_1; then, amongst these, we will find those subsets which minimise y_2; and so on.

To make this clearer, we will examine in more detail what minimising the first few quantities q_h entails.

<u>h = 1</u>

It turns out that minimising the quantity y_1 is equivalent to the conventional and widely accepted rules for allocating controls in block designs. (See, for example, Patterson, Williams and Hunter (1978), page 397.) We have:

$$\begin{aligned} y_1 &= \sum_{i\varepsilon K} \sum_{j\notin K} (a_{ii} + a_{jj} - 2a_{ij}) \\ &= -2 \sum_{i\varepsilon K} \sum_{j\notin K} a_{ij} . \end{aligned}$$

Now,

$$\begin{aligned} \sum_{i\varepsilon K} \sum_{j\notin K} a_{ij} &= \sum_{i\varepsilon K} \sum_{j=0}^{v-1} a_{ij} - \sum_{i\varepsilon K} \sum_{j\varepsilon K} a_{ij} \\ &= cr(k - 1) - \sum_{i\varepsilon K} \sum_{j\varepsilon K} a_{ij} . \end{aligned}$$

So,

$$y_1 = -2cr(k-1) + 2 \sum_{i\varepsilon K} \sum_{j\varepsilon K} a_{ij} .$$

Hence, in order to minimise y_1, we would want to minimise the quantity

$$\sum_{i\varepsilon K} \sum_{j\varepsilon K} a_{ij} ,$$

which is twice the number of times which controls concur together in blocks. In other words, we would want to allocate the controls to variety labels in such a way that the controls concur with each other in blocks as seldom as possible. Thus, again (as with the $\underline{\underline{S}}$-criterion in 2.4.3), we have been able to exploit a graph theory approach to give a firmer mathematical foundation to a standard practice (namely, the method of allocating controls advocated by Patterson, Williams and Hunter (1978)). Furthermore, this approach indicates how this practice could be generalised, for it suggests that, having minimised y_1, we should subsequently attempt to minimise y_2.

<u>h = 2</u>

The quantity y_2 can be simplified as follows.

$$y_2 = \sum_{i\varepsilon K} \sum_{j\not\varepsilon K} ((A^2)_{ii} + (A^2)_{jj} - 2(A^2)_{ij})$$

$$= (v-c) \sum_{i\varepsilon K} (A^2)_{ii} + c \sum_{j\not\varepsilon K} (A^2)_{jj} - 2 \sum_{i\varepsilon K} \sum_{j\not\varepsilon K} (A^2)_{ij}$$

$$= (v-c) \sum_{i\varepsilon K} (A^2)_{ii} + c(\mathrm{Tr}(A^2) - \sum_{i\varepsilon K} (A^2)_{ii}) - 2cr^2(k-1)^2$$

$$+ 2 \sum_{i\varepsilon K} \sum_{j\varepsilon K} (A^2)_{ij}$$

$$= cTr(A^2) - 2cr^2(k-1)^2 + (v-2c) \sum_{i\varepsilon K} (A^2)_{ii} + 2 \sum_{i\varepsilon K} \sum_{j\varepsilon K} (A^2)_{ij}$$

$$= cTr(A^2) - 2cr^2(k-1)^2 + (v-2c+2) \sum_{i\varepsilon K} (A^2)_{ii} + 2 \sum_{\substack{i,j\varepsilon K \\ i\neq j}} (A^2)_{ij}$$

So, for a given design, minimising y_2 is equivalent to minimising the final two terms in this expression. In particular, if the diagonal elements $(A^2)_{ii}$ are all equal (for example, if the design is cyclic, or if it has no double lines), then to minimise y_2 we must minimise:

$$\sum_{\substack{i\varepsilon K \\ i\neq j}} \sum_{j\varepsilon K} (A^2)_{ij}$$

That is, we must minimise the number of paths of length two which join controls to each other.

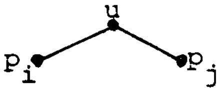
In other words, having first listed the ways of allocating the controls so that they are joined to each other by as few paths of length 1 as possible, we now suggest that the second, subsequent, aim should be to allocate them so that they are joined to each other by few paths of length two.

The calculation of the number of such paths for the various choices of the set K can, as we might expect, be reduced to fairly straightforward algorithms in the case of certain specific series of designs. As an illustration, we will consider the application to $\alpha(0,1)$ designs in which there are to be fewer than s control varieties. (That is, where $c < s$, s being the number of blocks in a replicate.)

Patterson, Williams and Hunter (1978) recommend that in a situation like this the controls should be allocated to the first c variety labels 0, 1,....., c-1. This allocation would certainly achieve our aim of minimising the quantity y_1, since these first c varieties do not concur with each other in an α design. However, allocating the controls in this way would not necessarily achieve our second aim: namely, minimising y_2. What we show here is how that second aim might be fulfilled. We will show that it leads to a straightforward set of rules for allocating the controls which is based on the entries of the generating array α (which we will assume to be in standard form).

To simplify matters, we will follow Patterson, Williams and Hunter (1978) as far as choosing to allocate the controls to variety labels in the first column of the design: that is, to labels in the set Z = {0, 1,....., s-1}. So our problem is to find that subset K of c varieties in Z which minimises y_2. Since the design has no double lines, this means that we should find the subset whose members are joined to each other by as few paths of length two as possible.

Consider any subset K = {p_0,....., p_{c-1}} consisting of c varieties from Z. Then paths of length 2 between two elements of K are of the form:



where u is some other variety not necessarily in Z. Suppose that u lies in column j_1 of the design, that the line p_i - u corresponds to concurrence in block x_0 of replicate h_0, and that the line p_j - u corresponds to concurrence in block x_1 of replicate h_1. Then, since varieties p_i and p_j both occur in column 0, and since the generating array α is in standard form, we get the following four equations (arguing as in Chapter 4 - for example, pages 127 - 129):

$$\begin{aligned} p_i &= x_0 \ ; \\ u &= j_1 s + (x_0 \dotplus \alpha_{h_0 j_1}) \ ; \\ u &= j_1 s + (x_1 \dotplus \alpha_{h_1 j_1}) \ ; \\ p_j &= x_1 \ . \end{aligned}$$

It follows from this that:

$$p_j \dot{-} p_i = \alpha_{h_0 j_1} \dot{-} \alpha_{h_1 j_1} \ . \qquad (6.13)$$

Conversely, for each equation (6.13) there is exactly one p_i - p_j path of length 2, since with $x_0 = p_i$ and $x_1 = p_j$, the value of variety u is fully determined.

Hence the number of p_i - p_j paths of length 2 is the same as the number of equations (6.13). So the total number of paths of length 2 which join two controls to each other is exactly half the number of equations like (6.13) (with p_i and p_j ranging over the set K, $0 \leq h_0, h_1 \leq r-1$ $(h_0 \neq h_1)$, and $0 \leq j_1 \leq k-1$). The factor of half is present because the enumeration of equations (6.13) counts each path twice, once for each of its endpoints.

We can translate this into a simple algorithm as follows. Given any $\alpha(0,1)$ design, we first form the $r(r - 1)\times k$ array H of elements $\alpha_{h_0 j_1} \dot{-} \alpha_{h_1 j_1}$ for $0 \leq h_0, h_1 \leq r-1$ $(h_0 \neq h_1)$, and $0 \leq j_1 \leq k-1$. We then choose the c control varieties p_u out of the set Z so that the values which appear among the resulting differences $p_i \dot{-} p_j$ occur as seldom as possible in the array H.

For instance, consider the Example (2a) which we discussed earlier (page 272): an $\alpha(0,1)$ design for 30 varieties in blocks of size 5, each variety being replicated 4 times. The generating array is:

$$\alpha = \begin{array}{ccccc} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 3 & 2 & 4 \\ 0 & 5 & 2 & 3 & 1 \\ 0 & 4 & 5 & 1 & 2 \end{array},$$

and s = 6. Then the array H is contained in the final five columns of the following table:

h_0	h_1					
0	1	0	5	3	4	2
0	2	0	1	4	3	5
0	3	0	2	1	5	4
1	0	0	1	3	2	4
1	2	0	2	1	5	3
1	3	0	3	4	1	2
2	0	0	5	2	3	1
2	1	0	4	5	1	3
2	3	0	1	3	2	5
3	0	0	4	5	1	2
3	1	0	3	2	5	4
3	2	0	5	3	4	1 .

The occurrences of the residues 1, 2, 3, 4, and 5 in this array are:

1	2	3	4	5
10	9	10	9	10 .

Suppose now that there are to be 2 control varieties, p_0 and p_1. There are $\binom{6}{2} = 15$ possible pairs, but among these only three sets of differences $p_0 \dot{-} p_1$, $p_1 \dot{-} p_0$ occur, namely {5, 1} , {4, 2} , and {3, 3} . The pair of differences {1, 5} gives 10 + 10 = 20 equations (6.13) since 1 occurs 10 times in H, as does 5. Similarly, the pair {2, 4} gives 9 + 9 = 18 equations (6.13), and the pair {3, 3} gives 10 + 10 = 20. So the best choice of two controls under our proposed rules is a set which gives rise to differences 2 and 4. For example, we could allocate the controls to the labels 0 and 2.

Now, this conflicts with the choice that would have been made if we were to use the rule suggested by Patterson, Williams, and Hunter (1978): they would allocate the controls to the labels 0 and 1. But, in fact, in this case our recommendation is slightly better than this conventional recommendation. If the controls are given to labels 0 and 1, then the average variance of a difference between a control variety and the rest is .62073280, whereas if our recommendation is followed and the cotrols given to labels 0 and 2, then the average is the slightly smaller .62063979.

If we apply these proposed rules sequentially, then they will, clearly, be at least as good as the rules conventionally in use. Minimising the quantity y_1 will, as we explained, lead to our allocating the controls in such a way as to minimise the concurrences between them. That is the same as spreading them throughout the blocks as evenly as possible, and this is, of course, the advice given by, for example, Patterson, Williams and Hunter (1978). Moreover, subsequently attempting to minimise the quantity y_2 can lead, as we have just seen in the Example, to some improvement over the allocation that would have been arrived at by the conventional recommendations. Of course, the theory on pages 277 - 281 above offers no conclusive guarantee that the allocation which minimises y_1 and y_2 will also minimise the average variance of differences between control varieties and the rest. Nevertheless on the strength of that theory, we would speculate that this will usually happen in practice. (This speculation can be seen to be analogous to the speculation on page 275, and to the conjecture in Section 2.5.) So far, we have not found a counterexample to this speculation; indeed, the theory on which it is based makes clear that it is as reasonable to accept its truth as it is to accept the rules conventionally recommended. In the light of these considerations, we would suggest that it might be of some advantage to experimenters if our rules for allocating controls were adopted in practice.

CONCLUSION

A conclusion is, of course, invidious, suggesting, as it does, a finality which is quite contrary to the spirit in which we would like to end: the whole tenor of this thesis is that it is merely a preliminary essay on the areas we have covered. So we will confine ourselves to mentioning very briefly the ways in which our material could be extended.

It could be extended, most obviously, by further research into the theory and applications of the graphical criteria of efficiency: into, for instance, the conjecture in 2.5, and into their use in generating efficient cyclic designs. The basis of the criteria in the variety concurrence graph will, we would expect, continue to permit considerable simplifications.

But the more important extension would be in the further and broader use of graph theory to illuminate the problems of the design of experiments. We have been concerned in this thesis mainly with the capacity of graph theory to simplify combinatorial complexities; and our material in Chapters 3 and 4 especially would indicate that there is, indeed, a great deal of scope for this to be done. There is, as well, something which we have not looked at very much here: the prospect of using some of the deeper theorems of abstract graph theory itself to clarify the structure of designs.

So it does appear likely, we would suggest, that graph theory has the potential to become a highly productive methodology in the general investigation of experiment design.

APPENDIX 1

THE CHOICE OF SOLUTION TO THE NORMAL EQUATIONS

Underlying the theory in Chapter 2 was the solution (2.1) to the normal equations for estimating the variety effects subject to the side condition $\underline{1}^T\underline{\tau} = 0$:

$$\hat{\underline{\tau}} = (C + rJ)^{-1}\underline{q} .$$

It was by expanding the expression $(C + rJ)^{-1}$ as an infinite sum of matrices that we established the link between the harmonic mean efficiency factor $\underline{\underline{A}}$ and the numbers of circuits in the variety concurrence graph. The form of this solution was, therefore, crucial to the development of our argument; it was the pivot which held together the several aspects of our exposition: the mathematical structure of the design, the conceptual framework of graph theory, and the statistical purpose of the experiment. It also had the incidental advantage (as we noted on page 55) of being the solution used by other researchers, such as Pearce and Tocher.

Beyond these advantages, moreover, lies one further reason for choosing this particular solution. This reason concerns the rate of convergence of the infinite series (2.4) and (2.5) (the series which, of course, also played a central role in our argument). Any solution of the normal equations subject to the side condition $\underline{1}^T\underline{\tau} = 0$ must be of the form:

$$\hat{\underline{\tau}} = (C + \theta J)^{-1}\underline{q} ,$$

where θ is some non-zero real number. We show in this Appendix that choosing $\theta = r$ is sensible if we want the series associated with the infinite expansion of the solution to converge rapidly.

Now,

$$(C + \theta J) = rI - \frac{1}{k}NN^T + \theta J$$
$$= r(I - \frac{1}{rk}NN^T + \frac{\theta}{r} J).$$

So:

$$(C + \theta J)^{-1} = \frac{1}{r}(I - \frac{1}{rk}NN^T + \frac{\theta}{r} J)^{-1}.$$

Arguing as in 2.3.1(a), we can establish that the eigenvalues of the matrix $\frac{1}{rk}NN^T - \frac{\theta}{r} J$ are $1 - \frac{\theta}{r}$ and $1 - e_i$ $(1 \leq i \leq v-1)$. The quantities $1 - e_i$ all lie in the interval $[0,1)$. So, provided $0 < \theta < 2r$, all these eigenavalues lie in the interval $(-1,1)$, and hence:

$$(I - \frac{1}{rk}NN^T + \frac{\theta}{r} J)^{-1} = I + \sum_{n=1}^{\infty} (\frac{1}{rk}NN^T - \frac{\theta}{r} J)^n.$$

That is,

$$r(C + \theta J)^{-1} = I + \sum_{n=1}^{\infty} (\frac{1}{rk}NN^T - \frac{\theta}{r} J)^n ,$$

which is analogous to the series numbered (2.4). We will therefore call this series here (2.4a). Now,

$$\mathrm{Tr}(r(C + \theta J)^{-1}) = r\theta^{-1} + \sum_{i=1}^{v-1} e_i^{-1} ,$$

which we will label $\beta(\theta)$. Hence:

$$\underline{\underline{A}} = \frac{v - 1}{\beta(\theta) - r\theta^{-1}} .$$

Also,

$$\mathrm{Tr}(r(C + \theta J)^{-1}) = v + \sum_{n=1}^{\infty} \mathrm{Tr}((\frac{1}{rk}NN^T - \frac{\theta}{r} J)^n) ,$$

which we will call (2.5a) (since it is analogous to the series (2.5) in the main text).

Define $\beta_m(\theta) = v + \sum_{n=1}^{m} \mathrm{Tr}((\frac{1}{rk}NN^T - \frac{\theta}{r} J)^n)$, so that

$$\beta_m(\theta) \rightarrow \beta(\theta) \text{ as } m \rightarrow \infty .$$

So it makes sense to choose θ so as to minimise, if possible, all the quantities:

$$|\beta_m(\theta) - \beta(\theta)| .$$

Now,

$$\beta_m(\theta) = v + \sum_{n=1}^{m} ((1 - \frac{\theta}{r})^n + \sum_{i=1}^{v-1} (1 - e_i)^n) ,$$

and

$$\beta(\theta) = v + \sum_{n=1}^{\infty} ((1 - \frac{\theta}{r})^n + \sum_{i=1}^{v-1} (1 - e_i)^n) .$$

So:

$$\begin{aligned}\beta(\theta) - \beta_m(\theta) &= \sum_{n=m+1}^{\infty} ((1 - \frac{\theta}{r})^n + \sum_{i=1}^{v-1} (1 - e_i)^n) \\ &= \frac{r}{\theta}(1 - \frac{\theta}{r})^{m+1} + \sum_{i=1}^{v-1} \frac{(1 - e_i)^{m+1}}{e_i} .\end{aligned}$$

It will be convenient to divide the range of θ into two, so that (a) $0 \leq 1 - \frac{\theta}{r} < 1$, and (b) $-1 < 1 - \frac{\theta}{r} \leq 0$.

(a) $\underline{0 \leq 1 - \frac{\theta}{r} < 1}$

That is, $0 < \theta \leq r$. Then $\beta(\theta) - \beta_m(\theta) \geq 0$, and so to minimise this difference, we want to minimise the quantity $\frac{r}{\theta}(1 - \frac{\theta}{r})^{m+1}$. By considering the first derivative of this expression, we can find that this minimum is attained in the range $(0,r]$ at $\theta = r$.

(b) $\underline{-1 < 1 - \frac{\theta}{r} \leq 0}$

That is, $r \leq \theta < 2r$. Here, we have a slight problem, since the expression $\beta(\theta) - \beta_m(\theta)$ might be either positive or negative. (To be precise, it is positive for m odd, and also for m even if $\theta \leq 2r - re_{min}$.) However, we still have:

$$|\beta(\theta) - \beta_m(\theta)| \leq \frac{r}{\theta}(\frac{\theta}{r} - 1)^{m+1} + \sum_{i=1}^{v-1} \frac{(1 - e_i)^{m+1}}{e_i} .$$

Since $\sum_{i=1}^{v-1} \frac{(1 - e_i)^{m+1}}{e_i} \to 0$ as $m \to \infty$, it still makes sense to minimise $\frac{r}{\theta}(\frac{\theta}{r} - 1)^{m+1}$. Again by considering the first derivative, the minimum of this expression for $r \leq \theta < 2r$ is attained at $\theta = r$.

So the choice of $\theta = r$ is, in this sense, optimal, in that it will tend to maximise the rate of convergence of the series (2.4a) and (2.5a).

APPENDIX 2

SOME BETTER THREE-REPLICATE $\alpha(0,1)$ DESIGNS

In this Appendix we give one small illustration of the results that can be achieved when the graphical criteria of efficiency are used to help search for efficient designs.

We mentioned at the end of Section 5.3 that we had used the triangle algorithm to improve on certain of the three-replicate $\alpha(0,1)$ designs that are listed in the ARCUS catalogue. It is these improvements that we detail here. We have concentrated on improving the third rows of the generating arrays: that is, in each case we took the first two rows as given in the catalogue, and then used the triangle algorithm to search for a third row that was better than the third row listed. That we could do this was, we reiterate, a consequence of the savings made possible by the graph theory techniques which underlie the algorithm: these techniques permit a considerable curtailment of the amount of searching that has to be carried out. It seems probable that graph theory could help in a similar way in achieving greater improvements by altering the second row as well. And it could, more generally, be used to devise algorithms for constructing efficient α designs with more than three replicates, and efficient designs of other types.

The table which follows lists the 20 improved designs which we have found. (In the remaining 29 cases in the catalogue we were not able to find any better designs than the ones already known.) They are arranged as in the catalogue, according to increasing values of s and k with k changing more rapidly than s. In each case, the generating array is given first, at the left-hand margin: in fact, only the last two rows and the last k - 1 columns, the first row and the first column being assumed to consist of zeroes. Then, in column 2 of the table, is the harmonic mean efficiency factor of this design, in column 3 is the absolute improvement this represents over the best design in the catalogue; and in column 4 this improvement is expressed as a percentage of the harmonic mean efficiency factor of the design in the catalogue. For example, the first design is for s = 10 and k = 5. The generating array is:

$$\begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 3 & 4 & 8 \\ 0 & 5 & 9 & 2 & 1 \end{matrix} ,$$

and this has harmonic mean efficiency factor .75731454. The catalogued generating array has harmonic mean efficiency factor .75716650. The absolute improvement is, therefore,

$$.75731454 - .75716650 = .00014804 ,$$

and this is, as a percentage of the harmonic mean efficiency factor in the catalogue,

$$\frac{.00014804}{.75716650} \times 100$$

$$= .01955184 .$$

Generating array	Harmonic mean efficiency factor	Absolute improvement	Percentage improvement
(1)	(2)	(3)	(4)
s = 10 k = 5			
1 3 4 8 5 9 2 1	.75731454	.00014804	.01955184
s = 10 k = 6			
1 2 3 5 6 4 9 5 1 7	.79834761	.00004463	.00559061
s = 10 k = 7			
7 6 9 2 4 1 2 3 8 4 7 5	.82726784	.00027245	.03294456
s = 11 k = 5			
1 4 9 7 8 2 6 10	.75409153	.00089929	.11939714
s = 11 k = 6			
1 2 4 8 9 4 6 9 7 10	.79481871	.00012560	.01580484
s = 11 k = 7			
4 5 2 1 6 3 3 6 9 10 8 7	.82417192	.00015464	.01876660
s = 12 k = 5			
3 2 10 7 7 5 9 8	.75135796	.00051776	.06895742
s = 12 k = 6			
3 2 10 7 1 4 8 9 10 5	.79200849	.00008914	.01125620
s = 12 k = 7			
1 7 9 8 3 10 5 6 3 10 4 8	.82169780	.00033593	.04089915

Generating array (1)	Harmonic mean efficiency factor (2)	Absolute improvement (3)	Percentage improvement (4)
s = 12 k = 8			
2 6 7 10 3 5 4 4 5 11 3 10 8 1	.84388275	.00014691	.01741185
s = 13 k = 4			
1 3 9 2 6 5	.68331835	.00024538	.03592296
s = 13 k = 5			
1 2 4 9 6 10 11 12	.74811319	.00100363	.13433505
s = 13 k = 6			
1 3 9 4 7 5 10 12 6 2	.78981944	.00016945	.02145887
s = 13 k = 7			
1 3 9 12 7 4 4 11 8 1 5 10	.81954858	.00024924	.03042112
s = 14 k = 4			
1 9 11 13 10 5	.67957796	.00077505	.11417894
s = 14 k = 5			
1 9 11 2 7 4 5 13	.74589440	.00093852	.12598330
s = 14 k = 6			
1 9 11 2 5 3 6 4 10 1	.78812133	.00123013	.15632784
s = 14 k = 7			
1 9 11 4 3 5 6 7 8 10 4 13	.81763004	.00028606	.03499873

Generating array	Harmonic mean efficiency factor	Absolute improvement	Percentage improvement
(1)	(2)	(3)	(4)
s = 15 k = 5			
1 3 7 13 2 9 12 11	.74354255	.00132177	.17808313
s = 15 k = 6			
1 3 7 13 14 8 9 10 2 12	.78662467	.00168974	.21527135

REFERENCES

ARCUS catalogue of α designs. See Patterson and Silvey (1980), page 226.

BERGE, C. (1973). Graphs and hypergraphs. Translated Minieka, E. North-Holland Publishing Company, London.

BIGGS, N. (1974). Algebraic graph theory. Cambridge University Press.

BIRKHOFF, G. and MacLANE, S. (1965). A survey of modern algebra. The Macmillan Company, New York.

BOSE, R.C. (1947). Presidential address. Proceedings of the Thirty-fourth Indian Science Congress. Quoted in Eccleston and Hedayat (1974).

BOSE, R.C. and NAIR, K.R. (1962). Resolvable incomplete block designs with two replications. Sankhyā, A24, 9-24.

CONNIFFE, D. and STONE, J. (1974). The efficiency factor of a class of incomplete block designs. Biometrika, 61, 633-6.

CONNIFFE, D. and STONE, J. (1975). Some incomplete block designs of maximum efficiency. Biometrika, 62, 685-6.

DAVID, H.A. (1963). The structure of cyclic paired-comparison designs. J. Aust. Math. Soc., 3, 117-27.

DAVID, H.A. and WOLOCK, F.W. (1965). Cyclic designs. Ann. Math. Statist., 36, 1526-34.

DEO, N. (1974). Graph theory with applications to engineering and computer science. Prentice-Hall, Englewood Cliffs, New Jersey.

di PAOLA, J.W. (1966). Block designs and graph theory. J. Combin. Theory, 1, 132-48.

ECCLESTON, J.A. and HEDAYAT, A. (1974). On the theory of connected designs: characterisation and optimality. Ann. Statist., 2, 1238-55.

EHRENFELD, A. (1955). On the efficiency of experimental designs. Ann. Math. Statist., 26, 247-55.

GILCHRIST, F. (1975). Block designs for large numbers of varieties. MSc dissertation, University of Edinburgh.

HARARY, F. (1969). Graph theory. Addison-Wesley, Reading, Massachusetts.

JACROUX, M. (1978). On the properties of proper (M,S)-optimal block designs. Ann. Statist., 6, 1302-9.

JACROUX, M. (1980). On the E-optimality of Regular Graph Designs. J. R. Statist. Soc., B42, 205-9.

JOHN, J.A. (1966). Cyclic incomplete block designs. J. R. Statist. Soc., B28, 345-60.

JOHN, J.A. (1981). Efficient cyclic designs. J. R. Statist. Soc., B43. (In the press at time of writing.)

JOHN, J.A. and MITCHELL, T.J. (1977). Optimal incomplete block designs. J. R. Statist. Soc., B39, 39-43.

JOHN, J.A., WOLOCK, F.W. and DAVID, H.A. (1972). Cyclic designs. United States National Bureau of Standards, Applied Mathematics Series, Number 62, Washington.

JOHN, P.W.M. (1971). Statistical design and analysis of experiments. The Macmillan Company, New York.

KEMPTHORNE, O. (1956). The efficiency factor of an incomplete block design. Ann. Math. Statist., 27, 846-9.

KIEFER, J. (1958). On the non-randomized optimality and randomized non-optimality of symmetric designs. Ann. Math. Statist., 29, 675-99.

KSHIRSAGAR, A.M. (1958). A note on incomplete block designs. Ann. Math. Statist., 29, 907-10.

MITCHELL, T.J. and JOHN, J.A. (1976). Optimal incomplete block designs. Oak Ridge National Laboratory/CSD-8.

PATTERSON, H.D. (1973). Quenouille's changeover designs. Biometrika, 60, 33-45.

PATTERSON, H.D. and WILLIAMS, E.R. (1976a). Some theoretical results on general block designs. Proceedings of the fifth British Combinatorial Conference, Congressum Numerantium XV, 489-96, Utilitas Mathematica, Winnipeg.

PATTERSON, H.D. and WILLIAMS, E.R. (1976b). A new class of resolvable incomplete block designs. Biometrika, 63, 83-92.

PATTERSON, H.D., WILLIAMS, E.R. and HUNTER, E.A. (1978). Block designs for variety trials. J. Agric. Sci., 90, 395-400.

PATTERSON, H.D., and SILVEY, V. (1980). Statutory and recommended list trials of crop varietes in the United Kingdom. J. R. Statist. Soc., A143, 219-52.

PEARCE, S.C. (1963). The use and classification of non-orthogonal designs. J. R. Statist. Soc., A126, 353-77.

RAGHAVARAO, D. (1971). Constructions and combinatorial problems in design of experiments. John Wiley and Son, New York.

RAY CHAUDHURI, D.K. (1970). On some connections between graph theory and experimental designs and some recent existence results. In Graph theory and its applications, edited Harris, B. Academic Press, New York.

RAY CHAUDHURI, D.K. and WILSON, R.M. (1970). On the existence of resolvable BIBDs. In Combinatorial structures and their applications. Calgary International Conference, University of Calgary. Gordon and Breach, New York.

RYSER, H.J. (1963). Combinatorial mathematics. John Wiley and Sons, New York.

SHAH, K.R. (1960). Optimality criteria for incomplete block designs. Ann. Math. Statist., 31, 791-4.

SHAH, K.R., RAGHAVARAO, D. and KHATRI, C.G. (1976). Optimality of two and three factor designs. Ann. Statist., 4, 419-22.

TAKEUCHI, K. (1961). On the optimality of certain types of PBIB designs. Report on Statistical and Applied Research, J.U.S.E., 8, 29-33.

TOCHER, K.D. (1952). The design and analysis of block experiments. J. R. Statist. Soc., B14, 45-100.

WALD, A. (1943). On the efficient design of statistical investigations. Ann. Math. Statist., 14, 134-40.

WHITE, A.J. (1968). Real analysis: an introduction. Addison-Wesley publishing company, London.

WILLIAMS, E.R. (1975). A new class of resolvable block designs. PhD thesis, University of Edinburgh.

WILLIAMS, E.R. (1976). Resolvable paired-comparison designs. J. R. Statist. Soc., B38, 171-4.

WILLIAMS, E.R. and PATTERSON, H.D. (1977). Upper bounds for efficiency factors in block designs. Aust. J. Statist., 19, 194-201.

WILLIAMS, E.R., PATTERSON, H.D. and JOHN, J.A. (1976). Resolvable designs with two replications. J. R. Statist. Soc., B38, 296-301.

WILLIAMS, E.R., PATTERSON, H.D. and JOHN, J.A. (1977). Efficient two-replicate resolvable designs. Biometrics, 33, 713-7.

WILLIAMS, R.M. (1952). Experimental designs for serially correlated observations. Biometrika, 39, 151-67.