**Finding an optimised design**

Information matrix =I = XCX

Screening methods

MS-optimal criterion

maximised tr(I)

minimised tr(I2)

**Three methods:**

a) Random sampling the rows of the treatment design matrix. Find the designs that:

1) Maximised the trace of the information matrix.

2) If the trace of in information matrix is the same, then find the design that maximised the average efficiency factor, i.e. harmonic mean of the non-zero canonical efficiency factors.

3) If the average efficiency factor is also the same, the find the design that has the minimal spread of the non-zero canonical efficiency factors, i.e. standard deviation of the non-zero canonical efficiency factors.

This method will affected by the number of the iteration is performed. The more iterations will likely to produce a more accurate result with more computational time required.

b) Random swapping two rows of the treatment design matrix. This method utilized the updating method on the information matrix without inverting the matrix.

This method is more sequential than the previous method.

The initial design plays a big part in the method, as swapping two rows or design points can only reach the local max-/minimum instead of the global max-/minimum.

An alternative is the repeat the procedure many times and using the design from method one as the initial design.

c) Building the treatment design matrix from matrix with all element equals to zero. Then study one row, i.e. one observation at a time.

1) Maximised the trace of the information matrix.

2) minimised the trace of the square of the information matrix.

Problem with this method is that it likely to produce many designs which past those two tests. Hence, the main issue with this method is to store all these designs. The current solution is that when the number of the design become more than 5000, we randomly choose 2500 out of them and continue with the iteration.

**Algorithms in general or Working Rule of an Algorithm**

The most common steps involved in any typical Optimisation Algorithm are the following:

1. Introduce a naïve design as a starting design and call it OLD.

2. Evaluate the criterion of optimisation on the OLD design.

3. Introduce a NEW design.

4. Evaluate the criterion of optimisation on the NEW design.

5. Compare the two evaluated values.

6. Accept the design that favours Optimality Criterion.

7. Call the design accepted at step 6 as OLD.

8. Go to step 3 and continue until no improvement is achieved on the criterion of evaluation or all possible configurations have been tested.

9. Accept the design that favours the optimality criterion the most.

**The Proposed Algorithm**

 Step1. We introduce a naïve design as a starting design randomly. The starting design should be connected and binary. The starting design can be obtained by any of the following two approaches:

Full Randomisation: Allocate test treatments and control treatments randomly to blocks with randomly selected replication numbers of both the test treatments as well as the control treatments.

Partial Randomisation: First allocate control treatments to each block once deliberately. Then allocate test treatments randomly in blocks with their replication numbers as close as possible.

 Step2. Delete the weakest observation and replace it by the strongest observation. This step is called exchange procedure.

 Step3. Continue with Step2 until no improvement is achieved.

 Step4. When no further improvement is achieved then start a new procedure named as *Interchange Procedure.*

 Step5. John and Eccleston (1980) gave an Interchange Procedure, which selects and interchanges a pair of treatments whose interchange yields an improvement with respect to optimality criterion, whereas we used *Strongest Interchange* for individual observation. In this algorithm we find the strongest interchange for each observation.

Another idea: set a upper bound average efficiency factor. When the during the search of the optimal design, if the average efficiency factor of the given design has reach close enough to the upper bound, then the search can be terminated.

What if there is no BIBD?

Kiefer's result suggests that we should look for optimal designs among the nearly

balanced incomplete block designs dened as follows.

Denition An incomplete block design is said to be nearly balanced if its

replication numbers ri dier by at most one and, for xed i, its concurrences ij

dier by at most one.

Denition An equireplicate nearly balanced incomplete block design is called a

regular graph design. Such designs have two dierent ij

's of the form and

+ 1. The concurrence graph consists of the -fold complete graph together

with a simple regular graph.

Such designs maximize tr(C) and minimize tr(C)

2

among those maximizing

tr(C)