Appendix: *Method of identifying potential designs*

**Definitions**

Let *G* be a graph with *n* vertices. A *sub-graph* of *G* is a graph consisting of a sub-set of the vertices of *G* together with the edges of *G* that connect elements of the chosen sub-set. A graph is *connected* if any two of its vertices can be connected by a sequence of edges. Two sub-graphs of *G* are *connected* if at least one vertex of one is connected to at least one vertex of the other. A *cluster* is a connected subgraph of *G*. A *minimum separator set* of a cluster is the smallest set of vertices whose removal disconnects the cluster.

A *solution graph*, *S(G),* is a sub-graph of *G* that itself contains the maximum possible number of disconnected subgraphs. A *clique* is a subgraph of G with an edge between every pair of vertices in the subgraph. We denote by *k(G)* the maximum number of separate cliques. The *independence number*, *a(G)*, of *G*, is the size of the biggest possible set of vertices such that no two vertices in the set are connected.

**Lemma**

In a solution graph *S(G)*, all clusters are cliques.

**Proof**

Suppose one of the clusters *C* of *S(G)* is not a clique. Then there will be two vertices *v1* and *v2* in *C* that have no edge connecting them. Removing all vertices in *C* except *v1* and *v2* increases the number of disconnected subgraphs in *S(G).* This is a contradiction, since *S(G)* is maximal by definition.

**Lemma**

The independence number of *G* is equal to the maximum number of independent cliques of *G.*

**Proof**

If we have a maximum number of independent cliques *{C1,C2,…,Cm}* ,where *m=a(G),* then choosing a vertex from each clique, *{v1,v2,…,vm}*, will give a set of independent vertices, so *a(G)≥k(G).* A set of independent vertices *{v1,v2,…,vm}* is also a set of independent cliques, so *a(G)≤k(G).* Hence *a(G)=k(G)*

**Observation**

Finding the independence number of a graph is NP-hard [1], therefore finding the maximum number of cliques is also NP-hard. This means that finding a solution is likely to involve searching all options. However, the R-package igraph implements an algorithm due to [2] for finding the independence number of a graph. We now describe an algorithm that uses the functions ivs and min\_separators in the igraph package to find a maximal set of independent cliques of *G.*

**Algorithm**

Step 1

Decompose *G* into separate clusters and define an empty set, *L*, to receive cliques that will in due course form the final design.

Step2

For each cluster *D* of G:

1. If *D* is a clique then add *D* to *L* and move to the next cluster.
2. If *D* is not a clique:

(i) use the function ivs to find the independence number of *D*;

(ii) find a minimum separator set *M(D)* of *D* using the function min\_separators (this set may not be unique);

(iii) consider the resulting graph *D\M(D).* For each cluster of *D\M(D),* repeat step 2(a) or 2(b) as appropriate.

Step3

All remaining clusters now form a maximal set of cliques, *L*. Compare the size of *L* with the independence number *a(G)* of *G*.

1. If the size of *L* is equal to *a(G)* then *L* is a maximum set of cliques.
2. If the size of *L* is less than *a(G)* then repeat the above process, choosing different minimum separator sets until the maximum is found.

**Observation**

At each stage of the algorithm, where there is a choice of minimum separator sets for cluster under consideration different choices may lead to different maximal sets of cliques. Amongst these multiple solutions, we can choose the one that meets the secondary criterion of maximising the number of included sampling units.

References

1. Garey MR, Johnson DS. Computers and intractability. A guide to the theory of NP-completeness. Freeman; 1979.

2. Tsukiyama S, Ide M, Ariyoshi H, Shirakawa I. A new algorithm for generating all the maximal independent sets. SIAM Journal on Computing. SIAM; 1977;6:505–17.