DiGGer, a Design Generator

N.E. Coombes

September 27, 2006

Contents

1 Introduction **2**

2 Getting started with DiGGer **3**
2.1 DiGGer job description 5
2.2 The DiGGer `.in` input file 8
2.2.1 Design initialisation 8
2.2.2 Search Phase Settings 8
2.2.3 Objective Specifications 10
2.2.4 DiGGer control 11
2.3 The `.trt` treatment file 11
2.4 Screen output 11
2.5 The `.out` file 12
2.6 The `.log` file 13
2.7 The `.lst` file 13
2.8 Analysis for this design 13

3 Maximum design sizes **14**

4 Design Efficiency in the Presence of Correlation **14**

5 The DiGGer search algorithm **16**

6 DiGGer search strategies **17**
6.1 Blocking and control of nested sub-blocks 17
6.2 Treatment Groups 19
6.2.1 Higher replication of control treatments 19
6.2.2 Treatments of interest: test lines in a breeding trial 19
6.2.3 Treatments from different sources 20
6.3 Reading an initial design 20
6.4 Adjusting the number of interchanges 20
6.5 Terminating a search 21

7 Detailed description of the DiGGer `.in` input file **21**
7.1 Initialisation 21
7.2 Search Phase Settings 23
7.3 Objective specifications 25
7.4 DiGGer job control 26

8 Other input files **26**
8.1 The `.trt` treatment information file 26
8.2 The `.din` initial design file 27
8.3 The `.swp` swap code file 27

9 DiGGer output files **27**
9.1 The DiGGer `.out` output file 28
9.2 The DiGGer `.log` design file 28
9.3 The DiGGer `.lst` field book listing file 28

10 Examples **28**
10.1 Variety trial for 26 entries 29
10.1.1 Standard DiGGer correlation with no additional blocks . . 29
10.1.2 Standard DiGGer correlation with blocks in two directions 33
10.1.3 Relative efficiency of Example 10.1 designs 33
10.2 Animal house trial 36
10.3 Early generation $p-$rep trial 38
10.4 Unreplicated variety trial with grid controls 42
10.5 Sequential testing in a laboratory trial 42
10.6 Regular check varieties in a disease screening nursery 46
10.6.1 Step 1: Unreplicated check design 46
10.6.2 Step 2: Replicated test lines with fixed Checks 47
10.7 Missing plots in a design 48
10.8 Restricted randomisation for unruly pumpkins 49

1 Introduction

DiGGer is a design search program that finds efficient designs for experiments under specified correlation structures. DiGGer is for non-factorial experiments with experimental units that can be specified as a rectangular array.

Input files tell DiGGer about

- the layout of the experiment
- the correlation pattern between experimental units in the experiment
- how to control the search.
- experimental treatment properties.

DiGGer may be run from

- a command window by
  - interactively answering questions about the search to create an input file
  - providing a valid input file

DiGGer is provided as an executable file `DiGGer.exe`. It should be in a directory with no spaces in the directory path and should be used in directories with no spaces in the path names.

> WARNING:
>
> DiGGer aims to be a flexible tool for designing experiments in the presence of correlation. The designs produced may be efficient for the specified correlation but may have some features that are not optimal in particular layout and correlation combinations.
>
> **Designs must be checked** for odd features before they are used.
>
> The block factors used in generating designs should be included in analyses. It is recommended a dummy analysis be prepared for non-standard designs to ensure that experiments have valid analyses.

In the example code that follows `typewriter font` is used to display output from the computer and user responses are given in **`bold`**.

2 Getting started with DiGGer

The easiest way to get started with DiGGer is to run an interactive session for a simple design. This will create typical input and output files and demonstrate DiGGer output. The input file can be used as a template and can be edited for more complicated examples.

Here is an example design for an experiment with 24 treatments and 3 replicates and the experimental units arranged as 24 rows by 3 columns. Each replicate is 24 rows by 1 column. Interactive DiGGer can provide default correlation settings which are appropriate for spatially correlated field trials, where correlation often decreases with row and column separation of experimental units. The file base name used for the design files in this example is `d24r3`.

At a command prompt call DiGGer with the file base name as its argument. If the file `d24r3.in` does not exist in the directory DiGGer will prompt for responses as in Figure 1. The information requested is the number of treatments, the design layout and the replicate layout.

```
C:\test>  DiGGer d24r3

 Job start time: Thu Jul 20 14:10:09 2006
 DiGGer 1.0.0

 Inputs from the keyboard will be used to create
 the search input file:
 d24r3.in

 Simple DiGGer is for equally replicated designs.
 Do you want to run the simplified version?[y]

<Enter> or y

 How many varieties (treatments) in the trial?

24

 What are the dimensions of this experiment?
 Enter Rows then Columns(Ranges) in this experiment

24  3

 Enter Rows then Columns(Ranges) in each replicate

24  1

 Do you want replicates or blocks
 in 2 directions? [y]

<Enter> or y.
```

Figure 1: Interactive DiGGer: simple case

The DiGGer program now runs a search. During the search output is produced in the command window showing the progress of the search. There are 5 files produced by the program:

- `d24r3.in` the DiGGer input file controls the search, Figure 4.
- `d24r3.trt` the DiGGer treatment information file (see Section 2.3) holds treatment name, number, replication and group details.
- `d24r3.out` the DiGGer output file records the details of the search and its progress.
- `d24r3.log` is created and updated after each 10% of a search phase and holds a matrix representation of the design.
- `d24r3.lst` is also created and updated after each 10% of each search phase and holds a field-book listing of the design, unit by unit, as rows nested within columns.

DiGGer recognises 2 other file type extensions: `.din` for a file that holds an initial design in matrix form, and `.swp` for a file holding search swap codes, in matrix form, to restrict allowable interchanges between plots within the design.

2.1 DiGGer job description

Before examining the input file `d24r3.in` in detail it is useful to discuss the strategy behind the DiGGer job and give a description of the search.

A DiGGer job has four components

- initialisation
- search phase control
- search phase objective
- DiGGer job control.

There is one initialisation section and at least one search phase. Each search phase has at least one objective function to optimise. At the end of a search phase a job control statement tells DiGGer whether the search continues in another phase. The Simple DiGGer strategy is

- Phase 1: to optimise blocks in 2 directions across the design under no correlation
- Phase 2: to randomise the treatment allocation within these blocks to optimise the design under spatial correlation across the whole design.

Initialise:

Treatments	= 24
Treatment type	= fixed
Design Layout	= 24 rows by 3 columns
Rep. Layout	= 24 rows by 1 column
Initial Design	= RCB generated in DiGGer

Search:

Phase 1:

Swap block layout	= 24 rows by 1 column
Corr block layout	= 8 rows by 3 columns
Objective measure	= A-value
Target value	= 0.0
Max. interchanges	= 10000

Objective 1:

Weight	= 0.8
Blocks 1	= 8 rows by 3 columns, $\gamma = 1.0$

Objective 2:

Weight	= 0.2
Blocks 1	= 8 rows by 1 column, $\gamma = 1.0$

Phase 2:

Swap block layout	= 8 rows by 1 column
Corr block layout	= 24 rows by 3 columns
Objective measure	= A-value
Target value	= 0.0
Max. interchanges	= 90000

Objective 1:

Weight	= 1.0
Blocks 1	= 1 row by 3 columns, $\gamma = 1.0$
Blocks 2	= 24 rows by 1 column,$\gamma = 0.5$
Blocks 3	= 1 row by 1 columns, $\gamma = 0.1$
Between rows	= AR1(0.5)
Between cols	= AR1(0.5)

Figure 2: d24r3 job description.

Figure 2 gives a job description for this search.

The Initialise section establishes the basic properties of the design. There are 24 fixed treatments. The design is arranged as 24 rows by 3 columns. A replicate occupies 24 rows by 1 column. DiGGer will create an initial Randomised Complete Block (RCB) design meeting these specifications.

The first phase of the search will allow treatments to be swapped within replicates (24×1) to try to improve on the initial design. Correlation blocks are 8 rows by 3 columns, the smallest size that contains the blocking factors specified in the objectives that follow. Correlation blocks must tile evenly within the design. The design objective measure to be minimised is the A-value. The target A-value which would trigger the end to this search phase (0.0) is unattainable so the other stopping criterion, the maximum number of treatment interchanges to test, will be used. The first objective to optimise is the set of 8 by 3 blocks which cut across the replicates. The secondary objective is the set of 8 by 1 blocks formed by the intersection of replicates with the Objective 1 blocks (Figure 3). The A-value of each objective is multiplied by the objective weight and summed to give an A-value for the design.

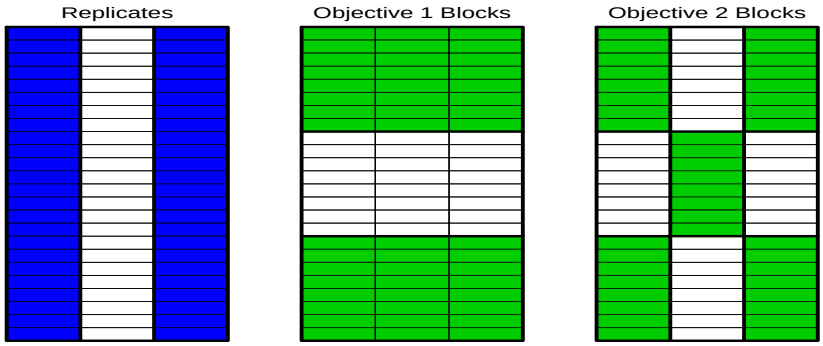


Figure 3: d24r3: Phase 1 block structures.

The optimised design from the first search phase is carried forward to the second search phase. This phase restricts swaps to within the intersection blocks (8×1), defines the correlation block as the whole design and will test 90000 interchanges. The objective to optimise is the standard spatial objective in DiGGer of random rows, columns and error with AutoRegressive errors in rows and columns.

Simple DiGGer asks whether to block in 2 directions when a replicate spans either the full set of rows, or the full set of columns in the design. Here there are 3 replicates which each span the full set of rows. Simple DiGGer tests whether the number of replicates divides the number of rows evenly and in this case can divide the columns into 3 sets of 8 rows. The set of blocks of 8 rows by

3 columns can each hold a replicate of the treatments. The objective weights control whether the main blocks hold replicates. The default weights $(0.8, 0.2)$ were chosen to give more emphasis to the efficiency of the larger blocks and in this case they will be replicates. For weights that sum to 1 the weight for Objective 1 must be at least 0.565 to get replicates in these blocks in this design.

2.2 The DiGGer `.in` input file

Figure 4 gives the input file that was created by DiGGer.

2.2.1 Design initialisation

Lines beginning with `#` are comments and are ignored. The design initialisation section shown in blue in Figure 4 always has 4 lines of data.

`SEEDS:` with no integers supplied causes DiGGer to set seeds for the Random Number Generator (RNG). Two seeds, $0 <$ seed1 $<$ 31328 and $0 <$ seed2 $<$ 30081, are reported in the `.out` file. These seeds control the randomisation of the initial design and the path of the search when no additional seeds are supplied. See Section 7.1 for a more complete discussion of the RNG seeds.

`24 FIXED:` gives the number of treatments and the treatment type. If treatments are given as `RANDOM` a treatment variance ratio ($\gamma > 0$) must also be supplied.

`24 3 24 1:` give the dimensions of the experiment, rows and columns, the dimensions of each replicate, rows and columns. Replicates must tile evenly within the design.

`GENERATE:` causes DiGGer to generate a Randomised Complete Block initial design

2.2.2 Search Phase Settings

The search phase settings sections, shown in red in Figure 4, always have 5 lines of data. In particular, the Phase 1 settings are:

`24 1 8 3:` giving the dimensions of swap blocks and correlation blocks, as rows and columns, in the first search phase. Swap blocks control allowable treatment interchanges in the search for an improved design. Initially swaps are allowed within replicates but are reduced to `8 1` blocks in the second search phase. This ensures that blocks established in the first search phase are maintained.

d24r3.in generated by Simple DiGGer

```
# Initialise Design ############
SEEDS                          # RNG seeds [-1 -1 [-1 -1]]
24   FIXED                     # No. treatments, [F-Fixed/R-Random [gamma]]
24 3   24 1                    # Layout: Design r c, Replicate r c
GENERATE                       # G-GENERATE/R-READ 1st design [S-SWAP file]

# Search Phase Settings #######
  24 1   8 3                   # Swap r c, Correlation r c
  A++                          # A-type
  0.0 10000                    # Target A-measure, Max. interchanges to test
  100                          # Search intensity, (low) 0 - 100 (high)
  2                            # Number of objectives
#   Objective Specifications ##
    0.8                        # Objective weight
      1                        #  No. of block factors
          8    3   1.00        #  Block r c gamma
      NONE                     #  None/Row/Column/Both - Linear trend code
      ID                       #  Correlation between rows: Model [par.]
      ID                       #  Correlation between columns: Model [par.]
      1.0                      #  Spatial gamma
#   Objective Specifications ##
    0.2                        # Objective weight
      1                        #  No. of block factors
          8    1   1.00        #  Block r c gamma
      NONE                     #  None/Row/Column/Both - Linear trend code
      ID                       #  Correlation between rows: Model [par.]
      ID                       #  Correlation between columns: Model [par.]
      1.0                      #  Spatial gamma

# DiGGer control ###############
NEW PHASE                      # N - NEW PHASE or E - END search here

# Search Phase Settings #######
  8 1   24 3                   # Swap r c, Correlation r c
  A++                          # A-type
  0.0 90000                    # Target A-measure, Max. interchanges to test
  100                          # Search intensity, (low) 0 - 100 (high)
  1                            # Number of objectives
#   Objective Specifications ##
    1.0                        # Objective weight
      3                        #  No. of block factors
          1    3   1.00        #  Block r c gamma
         24    1   0.50        #  Block r c gamma
          1    1   0.10        #  Block r c gamma
      NONE                     #  None/Row/Column/Both - Linear trend code
      AR  0.5                  #  Correlation between rows: Model [par.]
      AR  0.5                  #  Correlation between columns: Model [par.]
      1.0                      #  Spatial gamma

# DiGGer control ###############
END                            # N - NEW PHASE or E - END search here
```

Figure 4: Basic variety trial input file specifications.

The correlation blocks must tile evenly within the design. These define blocks of correlated plots in the design. Initially they are set to `8 3` to contain the blocks (8×3 and 8×1) defined in the Phase 1 search objectives. The design is split into three independent correlation blocks in this phase. The correlation block size is increased to a single `24 3` block in Phase 2 when the search objective has correlation across the whole design.

`A++:` is the code for the standard A-value type that optimises comparisons between all treatments in the design.

`0.0 10000:` gives the target A-value and the maximum number of treatment interchanges to test in the search for an improved design. They are alternative stopping rules for the search. Note that A-values are always greater than 0 so the target A-value does not affect this search. 10000 treatment interchanges will always be tested in the first search phase. Similarly 90000 interchanges will always be tested in the second search phase.

`100:` is the search intensity parameter to control the search when there are no treatment interchanges which improve the current design. It is a percentage [0-100]. Low values restrict acceptable interchanges resulting in a jagged, less intensive search. The smoothest, most intensive search occurs with the standard value of 100.

`2:` the final line in the search phase settings gives the number of objectives to be optimised in the search. Up to nine objectives may be given. The A-value for each objective will be calculated and may be used in a weighted objective A-value.

2.2.3 Objective Specifications

The objective specifications are shown in shades of green in Figure 4. Objective 1 of Search Phase 1 has:

`0.8:` which gives the weight associated with the first objective in the first search phase. The A-value for this objective will be multiplied by 0.8.

`1:` indicates there is one blocking factor in this objective.

`8 3 1.00:` gives the block dimensions, as rows and columns, and the variance ratio ($\gamma = 1.00$) associated with this block factor.

`NONE:` indicates linear trends across rows or columns will be assumed to be absent from the design.

`ID:` gives the model for correlation between rows of the design. `ID` assumes no correlation. Note that in the second search phase the correlation is `AR` indicating AutoRegressive errors and a parameter ($-1 < \rho < 1$), here `0.5`, must be supplied.

`ID:` is the model for correlation between columns with details equivalent to the previous item.

`1.0:` gives the variance ratio (γ) associated with the spatial model across rows and columns.

The objective in the second search phase is the standard correlation pattern used in DiGGer. It was was chosen to be appropriate for a typical cereal variety trial where sets of columns combine to form replicates and autoregressive correlations in rows and columns are the norm.

2.2.4 DiGGer control

The final component in DiGGer input files is a single line job control.

`NEW PHASE:` flags another phase in the search. The optimised design will be carried forward and a new search phase begins. `END` after the second phase of `d24r3.in` indicates the end of the DiGGer input file. Lines after this statement are ignored.

2.3 The `.trt` treatment file

Simple DiGGer creates a `.trt` file, if it does not exist, with 4 columns of data:

- treatment names which may be up to 30 characters long
- treatment number codes
- repeats of treatments within each replicate, 1 for equally replicated treatments
- treatment group. By default all treatments are in Group 1. Group information can be used to modify which treatment comparisons contribute to the A-measure.

Lines in the `.trt` file beginning with `#` are ignored.

2.4 Screen output

The progress of the search is output to the screen. Each time a design is accepted by DiGGer, when it becomes the current working design, the following information is printed to the screen:

- `Phs` the current search phase

d24r3.trt generated by Simple DiGGer

# TREATMENT NAME	NUMBER	REP	GRP
V1	1	1	1
V2	2	1	1
V3	3	1	1
...	..	.	.
V23	23	1	1
V24	24	1	1

Figure 5: Part of a basic treatment file.

- `Swaps` the number of interchanges tested in this phase
- `Best` the A-measure of the best design found
- `Current` the A-measure of the current working design
- `Desc` the descent number. A descent is completed when a randomisation of all acceptable interchanges has been tested
- `D` a design improvement code
  - `#` the current working design is the best design
  - `+` the current design improves on the previous working design
  - `-` the current design is worse that the previous working design
- `SCode` the swap code is a number associated with a particular plot pair eligible for interchange
- `Pl1` the plot identifier of the first plot in the `SCode` swap pair
- `Tr1` the treatment at plot `Pl1`
- `Pl2` the plot identifier of the second plot in the `SCode` swap pair
- `Tr2` the treatment at plot `Pl2`
- `B` ban length. The RTS algorithm bans the last `B` interchanges from being repeated in the search
- `Rpt` shows an integer n indicating the current working design is a repeat of the n^{th} working design in the search (0 = new design).

A message is also printed each time the `.log` and `.lst` files are updated.

2.5 The `.out` file

The `.out` file records the details of the search process from the design setup and the initial design to the best design found during the search. The best design at the end of each search phase is also shown. The search parameters, objectives and the A-measure of each improvement to the best design in the search path are recorded for each phase.

2.6 The .log file

The .log file provides a design in matrix form together with details of the A-measure, the interchanges used and the current search phase. This file is overwritten with the best design after each tenth of each search phase is completed. The numbers shown in the .log file are DiGGer design numbers corresponding to the position of the treatment in the .trt file.

2.7 The .lst file

The .lst file gives a field-book listing of the design as rows nested within columns corresponding to the design currently in the .log file. Treatment names are given in the ID column, treatment numbers in the ENTRY column and ROW, RANGE (column) and REP (replicate) details. The TRT column holds design numbers used by DiGGer and these are the numbers that appear in the .log file.

d24r3.lst generated by DiGGer

UNIT	ID	ENTRY	ROW	RANGE	REP	TRT
1	V4	4	1	1	1	4
2	V5	5	2	1	1	5
3	V23	23	3	1	1	23
4	V24	24	4	1	1	24
5	V9	9	5	1	1	9
6	V15	15	6	1	1	15
7	V12	12	7	1	1	12
..	..	..	..	..	..	..
..	..	..	..	..	..	..
68	V12	12	20	3	3	12
69	V23	23	21	3	3	23
70	V8	8	22	3	3	8
71	V6	6	23	3	3	6
72	V20	20	24	3	3	20

Figure 6: Part of the list file.

2.8 Analysis for this design

Blocking structures used in the construction of designs represent restrictions on randomisation which should be included in the analysis of the experiment. DiGGer reports the blocking structures used in a phase at the beginning of each phase in the .out file.

The blocking structures which control the randomisation of this design are:

- $\texttt{Rep}_{24\times1} \equiv \texttt{Column}$
- $\texttt{RBlock}_{8\times3}$
- $\texttt{Rep.RBlock}_{8\times1}$

- $\texttt{Row}_{1\times 3}$

The blockstructure which should be included in the analysis is:

$$\texttt{Rep} * \texttt{RBlock} + \texttt{Row} = \texttt{Rep} + \texttt{RBlock} + \texttt{Rep.RBlock} + \texttt{Row}$$

3 Maximum design sizes

A design with

- v treatments replicated r_i times, $i = 1, 2, \ldots v$
- a search phase with
  - B objectives
  - correlation blocks of C_r rows by C_c columns

is subject to the following restrictions:

- the maximum total number of experimental units is 1500 ($\sum_{i=1}^{v} r_i \leq 1500$)
- the number of treatments $v < 1500$
- the number of treatment groups $G \leq 100$
- the number of objectives $B < 10$
- the treatment information matrix requires $B \times v \times (v+1)/2 \leq 2250000$.
- the correlation matrix requires $C_r \times C_c \times B \leq 1500$.

The maximum number of plot interchanges in the swap list is 600000. For large designs the size of the swap list may be reduced by reading an initial design with some blocking established. Partially replicated and unreplicated designs can be set up with 'replicate' blocks already established.

4 Design Efficiency in the Presence of Correlation

The spatial mixed linear model from Gilmour *et al.* [4] may be written to emphasise the fixed effects of interest $\boldsymbol{\tau}_1$,

$$\mathbf{y} = \mathbf{X}_1\boldsymbol{\tau}_1 + \mathbf{X}_2\boldsymbol{\tau}_2 + \mathbf{Z}\mathbf{u} + \boldsymbol{\eta} + \boldsymbol{\xi} \tag{1}$$

In a variety trial this equation could be describing yield as the sum of:

- Variety effect, the fixed effects of interest
- Trial mean, fixed block, linear row or linear column trend, nuisance fixed effects

- Row effect, column effect and plot error effect, nuisance random block effects
- Spatial effect due to correlation between plots in the trial.

The distribution of $\mathbf{y}$ can be written

$$\mathbf{y} \sim \mathrm{N}(\mathbf{X}_1\boldsymbol{\tau}_1 + \mathbf{X}_2\boldsymbol{\tau}_2, \sigma^2\boldsymbol{\Lambda}) \tag{2}$$

where the variance matrix

$$\boldsymbol{\Lambda} = \sum_{i=1}^{m} \mathbf{Z}_i\mathbf{Z}_i'\gamma_i + \boldsymbol{\Sigma}_c \otimes \boldsymbol{\Sigma}_r\gamma_s \tag{3}$$

The $\mathbf{Z}_i$ are design matrices for random block effects. The standard DiGGer correlation model for a design with r rows and c columns has:

- random rows $\mathbf{Z}_r\mathbf{Z}_r'\gamma_r$, for the $1 \times c$ blocks with variance ratio $\gamma_r = 1.0$
- random columns $\mathbf{Z}_c\mathbf{Z}_c'\gamma_c$, for the $r \times 1$ blocks with variance ratio $\gamma_c = 0.5$
- random plot error $\boldsymbol{\eta} = \mathbf{Z}_e\mathbf{Z}_e'\gamma_e$, for the 1×1 blocks with variance ratio $\gamma_e = 0.1$
- $\boldsymbol{\xi} = \boldsymbol{\Sigma}_c \otimes \boldsymbol{\Sigma}_r\gamma_s$
  - $\boldsymbol{\Sigma}_r$ models correlation between rows as an AR1 (0.5) process
  - $\boldsymbol{\Sigma}_c$ models correlation between columns as an AR1 (0.5) process
  - the spatial variance ratio $\gamma_s = 1.0$

The mixed model equations lead to the equations for $\boldsymbol{\tau}_1$ as

$$\mathbf{X}_1'\boldsymbol{\Lambda}^*\mathbf{X}_1\boldsymbol{\tau}_1 = \mathbf{X}_1'\boldsymbol{\Lambda}^*\mathbf{y} \tag{4}$$

with the variance $\boldsymbol{\Lambda}$ adjusted for the nuisance fixed effects in the model

$$\boldsymbol{\Lambda}^* = \boldsymbol{\Lambda}^{-1} - \boldsymbol{\Lambda}^{-1}\mathbf{X}_2\left(\mathbf{X}_2'\boldsymbol{\Lambda}^{-1}\mathbf{X}_2\right)^{-1}\mathbf{X}_2'\boldsymbol{\Lambda}^{-1} \tag{5}$$

The matrix

$$\mathbf{C} = \mathbf{X}_1'\boldsymbol{\Lambda}^*\mathbf{X}_1 \tag{6}$$

is the information matrix for $\boldsymbol{\tau}_1$. If the variance components and spatial parameters in Equation 3 are assumed known then

$$\hat{\boldsymbol{\tau}}_1 \sim \mathrm{N}\left(\boldsymbol{\tau}_1, \sigma^2\mathbf{C}^-\right) \tag{7}$$

where $\mathbf{C}^-$ is the generalised inverse of $\mathbf{C}$.

The A-measures used in DiGGer are calculated from $\mathbf{C}^-$. In particular the standard A-measure, `A++`, for comparisons between all treatments in the design is $2\left(\mathrm{trace}(\mathbf{C}^-) - \mathrm{sum}(\mathbf{C}^-)/v\right)/(v-1)$. This value is proportional to the average variance of pairwise treatment contrasts for the design under the specified correlation structure. The usual A-value in the literature is just the $\mathrm{trace}(\mathbf{C}^-)$ which is $\sum 1/\psi_i$, where $\psi_i, i = 1, \ldots, v-1$ are the non-zero eigenvalues of $\mathbf{C}$ [5].

5 The DiGGer search algorithm

DiGGer uses a form of Reactive Tabu Search (RTS) (Battiti and Tecchiolli [1], Coombes [2]) to find improvements on an initial design. This is essentially a descent algorithm with rules for escaping from local minima when they are encountered in the search.

The swap list is the list of the pairs of plots on which treatment interchanges are allowed. The RTS search is divided into a series of descent steps. Each descent step applies each interchange from a new randomisation of the swap list to the current working design. After each interchange the A-measure on the new candidate design is calculated.

1. If the interchange improves on the best design it made immediately and the candidate design becomes the current working design.
2. If the interchange improves on the current working design it is accepted if the interchange is not taboo.
3. If no improvement is found to the current working design in the descent step, RTS uses the search history to choose an interchange from eligible non-improving interchanges to obtain a new design that avoids returning to designs in the search path.

The search path is the set of current working designs accepted during the search. An interchange is taboo if it repeats a plot interchange has been made in the previous b interchanges or if it would repeat a design in the recent search path. The ban length b is initially set to 1 and is altered as repeat designs enter the search path.

The search intensity parameter in DiGGer defines the percentage of non-improving moves to be considered when making the choice in item 3. above. If the search intensity parameter is 100.0 all possible interchanges are considered for non-improving moves and the new design is likely to be close in A-measure to the current design. At the opposite extreme, a search intensity parameter of 0.0 will cause an eligible interchange to be selected at random and the new A-measure may be distant from the current A-measure. The search intensity parameter thus affects the smoothness of the search. In most cases the DiGGer default value of 100.0 is the best value to use. Designs at local minima in block designs, however, can have many equivalent designs nearby. A low search intensity may be needed to find a global minimum in these cases.

The DiGGer program has two stopping rules for the search. The first is a target A-measure. The search stops when the current design has an A-measure lower than the target A-measure. Unless they are given negative weights A-measures are always greater than zero so the default DiGGer target A-measure of 0.0 will never be reached. The second stopping rule is the maximum number of interchanges to test in the search. The search stops once this number of interchanges has been reached unless the target A-measure is reached beforehand.

The list of eligible interchanges in RTS is controlled by the swap blocks defined in DiGGer and by restrictions specified in the `.swp` file if it exists. All

treatment interchanges between plots in a swap block are eligible. Only plots with the same swap code in the `.swp` file are allowed treatment swaps.

The A-measure calculation is computationally intensive. It requires the inversion of the correlation matrix and the inversion of the information matrix at the start of each search phase. Updating methods are used to calculate the new information matrix following a treatment interchange. The speed of the DiGGer search is primarily affected by the number of treatments in the design and to a lesser extent on the size of the correlation matrix.

6 DiGGer search strategies

6.1 Blocking and control of nested sub-blocks

Spatial designs are often built in DiGGer by searching to establish an intermediate block design. The resulting design is carried forward to a new search which restricts changes (treatment swaps) to within established blocks and has a spatial correlation model across the whole design. Such a design could be called a spatially optimised block design. Each component search in the DiGGer job is called a search phase. Spatial correlations should only be present in the final search phase.

By default Simple DiGGer (Figure 1) blocks a design in two directions. A number of rows combine to form the block structure `Rblock`. A number of columns combine to form `Cblock`. The initial search phase optimises blocks described by the model statement:

$$\texttt{Rblock} * \texttt{Cblock} = \texttt{Rblock} + \texttt{Cblock} + \texttt{Rblock.Cblock}$$

In Simple DiGGer one of these blocking structures will correspond to replicates in the design. The standard DiGGer correlation model fitted in the final search phase has random row and column blocks.

Writing these terms in full, the restrictions on randomisation in the design are:

$$\texttt{Rep} + \texttt{Rblock} + \texttt{Cblock} + \texttt{Rblock.Cblock} + \texttt{Row} + \texttt{Col}$$

where some of these terms may be redundant.

The blocks `Rblock.Cblock` formed by the intersection of the block structures `Rblock` and `Cblock` are an example of a nested blocking structure. The efficiency of nested sub-blocks dominate the efficiency of the blocks within which they are nested. The balance between the efficiency of nested blocking factors can be controlled by using separate objectives in the search and giving more weight to the objective with the larger block size. For example, the first search phase in Figure 2 gives the 8×3 blocks a weight of 0.8 while the nested 8×1 sub-blocks are given a weight of 0.2. This balance is adequate to ensure that the larger blocks contain a replicate.

An alternative method using one objective and the variance ratios of the nested blocking factors to control this emphasis is shown in Figure 12. The

9×3 blocks have $\gamma = 1.0$ while the nested sub-blocks 9×1 have $\gamma = 0.1$. The aim here is to find the most efficient sub-blocks within fully efficient main blocks.

Blocking in two directions ensures that particular treatments do not cluster together in favourable spatial configurations: self-diagonals or knight-move separations (Figure 11).

The importance of including intersection blocks is illustrated in Figure 7. Consider an example with 48 treatments arranged 24×4 with replicates 24×2. In the figure, treatments within blocks in the first replicate are plotted with the same colour. Simple DiGGer with blocks in two directions would by default optimise row blocks, 12×4 and intersection blocks 12×2 (Figure 7(c)). The design is connected. There are 7 treatments common across intersection blocks within row blocks. If intersection blocks are omitted from the search, Figure 7(a), there is no connectivity between the sub-blocks within row-blocks. There are no treatments in common between blocks 1 and 4 and blocks 2 and 3, effectively splitting the design into two experiments. If three row blocks are used, it is still

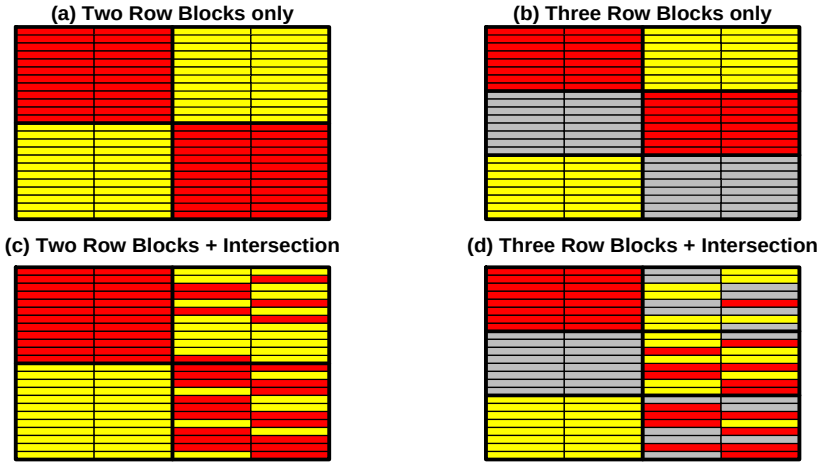


Figure 7: Connectedness of some block designs.

possible to reach a disconnected design (Figure 7(b)) if intersection blocks are not included in the search. Figure 7(d) shows direct connection between blocks within row blocks. There is one treatment repeat within row blocks. This is not necessary, however, and there are many indirect connections between blocks, yellow/grey, red/yellow and red/grey, in the second replicate blocks. Simple DiGGer always includes intersection blocks in the search. Designs produced out of Simple DiGGer need to include intersection blocks where appropriate to avoid disconnected designs.

6.2 Treatment Groups

Unequal replication and structure in the treatments give scope for adjusting the A-measure using treatment groups defined in the `.trt` file.

6.2.1 Higher replication of control treatments

Control treatments at higher replication in replicated trials are a problem when the standard `A++` A-measure is used with AR1 correlations. The resulting designs will have the higher replicated treatments placed in the perimeter of the design. This is not desirable when comparisons of treatments with the controls is a primary objective. Group codes may be used to distinguish treatments with different levels of replication. Specifying the `Agg` A-measure will ensure treatments from different groups are distributed more evenly to non-correlated blocks or spread more evenly in designs with AR1 correlations. Intermediate block designs are an important step to distribute treatments with higher replication across the design.

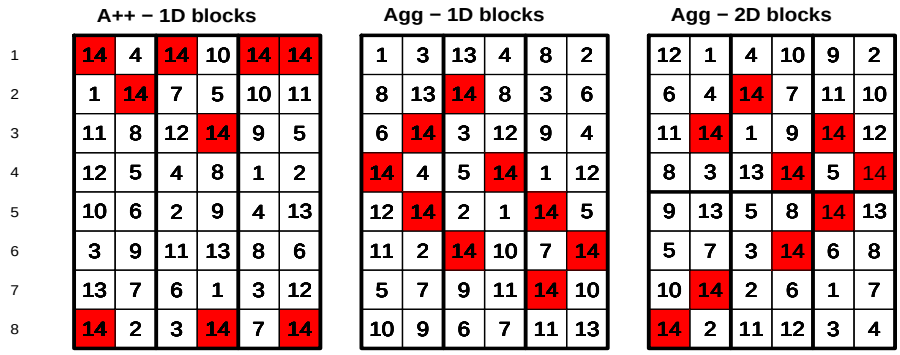


Figure 8: Designs with additional controls.

Figure 8 shows the results for different A-measures and blocking strategies for an example with Treatments 1 to 13 included in each replicate once and Treatment 14 three times in each replicate. The `Agg` A-measure distributes Treatment 14 more widely across the design. The self-diagonals in Treatment 14 are the effect of AR1 correlations in rows and columns. An alternative strategy for regularly spacing control treatments in a design is given in Example 10.6.

6.2.2 Treatments of interest: test lines in a breeding trial

Treatment groups may be used to distinguish between the types of treatments in a trial. A breeding trial will often contain control varieties at higher replication and test material at lower replication. The experimenter wishes the test line

comparisons to be most efficient, in a trial with control varieties spread evenly across the trial.

Suppose that control varieties are coded as Treatment Group 2 and test lines are coded as Treatment Group 1 in the `.trt` file. The first phase in designing such a trial would be to form blocks in two directions and distribute treatments from the two treatment groups evenly to those blocks. The `Agg` A-measure will achieve this. The second phase of the design would optimise comparisons between test lines using the `A11` A-measure (Example 10.3). The positions of control varieties in the design contribute indirectly to the efficiency of the comparisons between the treatments of interest, the test lines.

6.2.3 Treatments from different sources

There may be additional information about treatments which could be used to produce a better randomisation. In breeding trials there may be family information. Families should be distributed as evenly as possible across the trial to avoid variations in field conditions favouring particular families. In these trials family groups should be given unique codes in the `.trt` file.

In variety quality testing, where samples from a field trial are taken to the laboratory, samples have both genotype information and field location information which may be used in the laboratory design. Example 10.5 uses group codes to identify samples of the same genotype from different field locations. Samples from unreplicated entries are given a group code based on a field location property, column. The `Agg` A-measure for intermediate block designs ensures that genotypes are processed on different days and that there are a range of field columns represented in the samples on particular days.

6.3 Reading an initial design

The initial blocking of unreplicated trials into 'replicate' blocks can be slow in DiGGer. For large unreplicated trials there is a definite advantage in creating and reading an initial design. Section 10.6.1 gives an example of R code to create a `.din` file. The list of allowed swaps is large when using a search to divide a trial into replicates. Bypassing this step reduces the number of swaps that need to be considered in the search.

An initial design *must* be read when a design contains missing plots. Missing plots are indicated by a treatment code of 0 in the `.din` file.

Designs can be created with non-rectangular replicate shapes by reading an initial design from a `.din` file and using swap codes equivalent to replicate number in a `.swp` file.

6.4 Adjusting the number of interchanges

Blocks without correlation require many fewer interchanges to find optimal designs because there are multiple search paths that lead to many equivalent designs. By default Simple DiGGer with blocks in two directions use 10000

interchanges for the initial block search phase and 90000 interchanges for the search phase with correlation.

The number of interchanges required to obtain efficient block designs increases when treatment grouping information is used. The A-measure `Agg` finds many more interchanges than `A++` when unreplicated treatments have family group codes. The number of interchanges to specify in these cases is a matter of judgement. In some cases where too many interchanges have been specified for a blocking phase it is better to terminate the search and restart at the next search phase with an input design.

Designs are saved after each 10% of a search phase is completed. Specifying 100000 interchanges to test will mean that the first design is available after 10000 interchanges. For some unreplicated designs it will take too much time to output the first of these designs.

6.5 Terminating a search

A DiGGer job is killed by typing `Control-C`. If the search has progressed more than 10% through a search phase there will be a design in the `.log` and `.lst` files.

7 Detailed description of the DiGGer `.in` input file

Blank lines and lines beginning with `#` are ignored in the `.in` file. There are four components to a DiGGer job.

- The initialisation section describes features that apply to the whole of the search.
- Search phase settings define the settings which apply to a search phase.
- Search phase objectives define the block and correlation structures for a particular search phase.
- DiGGer job control statements signify the end of a DiGGer job or the beginning of a new search phase in the current DiGGer job.

The order of input is fixed in DiGGer input files and must be strictly adhered to. The files produced by Simple DiGGer have comments beside each input line describing the purpose of the input code. Simple DiGGer `.in` files may be used as templates for more complicated designs

7.1 Initialisation

The design initialisation section is shown in blue in the examples in this manual. There are 4 lines that initialise the DiGGer search.

1. `SEEDS` to initialise the Random Number Generator (RNG). The first line of the initialisation section has the key word `SEEDS`. If no numbers are given after the key word DiGGer will use the system clock to obtain two seeds to control the generation of an initial design and the order in which treatment interchanges are made in trying to find improvements to a design. The seeds, $0 < \text{seed1} < 31328$ and $0 < \text{seed2} < 30081$, are reported in the `.out` file.

   The user may give two or four seeds for the RNG. The first two seeds specifically control the generation of an initial design in DiGGer. If two additional seeds are given, they will re-initialise the RNG to control the order of interchanges tested during the search for improvements. Thus a search may be conducted generating the same starting design using the same design-initialising RNG seeds but using a different search path to find improvements by using different search seeds, seed3 and seed4, $0 < \text{seed3} < 31328$ and $0 < \text{seed4} < 30081$.

2. The second line of the initialisation section has the number of treatments in the design. This number may be followed by a key word `FIXED` or `RANDOM`. The key words may be abbreviated to `F` or `R`. If there is no key word treatments are assumed to be fixed. This is the default treatment type for DiGGer searches. If the treatments are random then a treatment variance ratio, γ_t, must be supplied. Negative or zero γ_ts will result in treatments being considered as fixed. Positive γ_ts cause treatments to be considered random and the inverse of γ_t is added to the diagonal of the treatment information matrix. Thus as γ_t gets large random treatments approach their fixed counterparts.

3. D_r D_c R_r R_c. The third line of the initialisation section has four integers defining the layout of the design. The first two integers, D_r and D_c, are the rows and columns in the design. The second two integers, R_r and R_c, are the rows and columns in each replicate. The first replicate begins at the first row and column. Subsequent replicates tile columnwise within the design. Replicates must tile evenly into the design. A completely randomised design has replicate dimensions equal to the design dimensions.

   There are many descriptors for rows and columns in plant breeding trials. Columns are often called ranges or banks. The design is listed in the `.lst` file as rows nested within columns. All of the plots in column 1 are listed followed by all of the plots in column 2, continuing to the last row in the final column.

4. The fourth line of the initialisation section has at least one key word, `GENERATE` or `READ`. If the keyword is `READ` it may be followed by a second key word `SWAP`. Again key words may be abbreviated to their initials.

   `GENERATE` causes DiGGer to generate an initial design for the layout and treatment information given.

   `READ` causes DiGGer to read a starting design from the `.din` file.

`SWAP` causes DiGGer to read swap codes from a `.swp` file after an initial design has been read. Treatment interchanges may only be made between plots with the same swap code.

7.2 Search Phase Settings

The search phase settings are shown in red in the examples in this manual. Search phase settings have five input lines.

1. S_r S_c C_r C_c. The first line has four integers giving the dimensions of swap blocks, S_r and S_c, and correlation blocks, C_r and C_c, respectively.

   - Swap blocks define the blocks within which treatment interchanges may be made. Swap blocks tile columnwise within the design. In the initial search phase swap blocks will usually have the same dimensions as replicate blocks. Exceptions may occur when initial designs are being read from files.

     The final swap block in a row or column may be truncated if it extends past the dimensions of the design. Uneven tiling is permitted.
   - Correlation blocks tile columnwise in the design but must also tile evenly within the design. In the special case where the design contains missing plots (see Section 10.7) the correlation block must have the same dimensions as the design. The size of correlation blocks affect the speed of the search. For intermediate search phases it is sometimes possible to reduce the correlation block size to speed up the search (Phase 1, Section 10.6.1).

2. A-measure type. There are 7 possible A-measures:

   - `A++` is the standard measure for making comparisons amongst all treatments. Treatment codes do not affect this measure. The standard A-measure is proportional to the trace of the inverse of the treatment information matrix ($\mathbf{C}^-$, Section 4).
   - `Agg` is the A-measure that optimises comparisons between treatments belonging to different treatment groups. Treatment group codes are given in the `.trt` file. `Agg` is the average variance of differences between treatments in different treatment groups. Let c_{jj} and c_{kk} be the diagonal elements of $\mathbf{C}^-$ and c_{jk} the off-diagonal element corresponding to treatments j and k. Let G_j and G_k be their treatment groups and N_{jk} be the number of comparisons between treatments of different groups. $\texttt{Agg} = \left(\sum_{G_j \neq G_k} c_{jj} + c_{kk} - 2c_{jk}\right) / N_{jk}$. `Agg` is often used to ensure that treatments from different treatment groups are evenly distributed across blocks in the first phases of a search, for example when check varieties and test lines in a variety trial are being allocated to blocks (Section 10.3). When control varieties are

in a replicated trial at higher replication than others, basing treatment groups on replication level will prevent control varieties moving to the perimeter of the design.

- `A22` is the average variance of treatment comparisons where both treatments being compared are in treatment group 2.
- `A11` is identical to the above but optimises comparisons between treatments in treatment group 1.
- `A1+` optimises comparisons between group 1 treatments and treatments in all other treatment groups.
- `Aa2` optimises the comparison between the average of Group 1 treatments with treatments in treatment Group 2.
- `Aaa` optimises the comparison between the average of Group1 treatments and the average of Group 2 treatments.

`A11` and `A22` are most often used in the last search phase, once blocks have been established, to optimise unreplicated or partially replicated experiments where treatment comparisons between test lines are of primary importance and the test lines are coded as Group1 and Group2 respectively. The final two A-measures may be of use where check treatments are coded Group1 and test treatments are coded Group2.

3. The third line gives two stopping rules for the search phase. The search phase ends when the A-measure is less than the target A-measure. This value defaults to 0.0 in files produced interactively and will not come in to play as A-measures are positive. The second stopping rule for the phase is the maximum number of treatment interchanges to test. The search phase stops when this number of interchanges is reached. The number of interchanges specified also affects the creation of `.log` and `.lst` files. These files are created and updated after each 10% of the interchanges specified have been tested.

4. The fourth line of the search phase gives a measure of search intensity. The default value of 100.0 givens the smoothest most intensive search. The number represents the percentage of interchanges to consider when an improvement to the current design is not possible. The RTS algorithm chooses an alternative from the available interchanges that meets the restriction imposed by the search algorithm. A low number restricts the available choices and results in a more random selection. This is useful in block designs where there are many similar designs close to local A-measure minima. It is less useful for spatial designs where improvements are likely to be found 'close' to the current design.

5. The final line in the search phase settings gives the number of objectives contributing to the A-measure. Up to 9 objectives may be defined.

7.3 Objective specifications

For each objective specified by the last line of the search phase settings there are 6 input lines plus an additional line for each block factor in the objective. Objectives are shown in shades of green in the examples in this manual. An objective gives a block/correlation structure to form the $\mathbf{\Lambda}$ correlation matrix (Equation 3) which is used to obtain the treatment information matrix $\mathbf{C}$ (Equation 6) from which an A-measure is calculated. Multiple objectives combine A-measures using weights given in the objective specifications.

1. Objective weight. The weight used to multiply the A-measure to contribute to the A-measure for the design.

2. Number of block factors. Gives the number of distinct blocking factors in the objective. There are three blocking factors in the standard DiGGer objective: rows, columns and error.

   (a) B_r B_c γ_B. Each blocking factor has a line specifying the block shape, rows and columns in each block, and a variance ratio γ_B. When $\gamma_B > 0$ the blocks are random. Any other value causes the blocks to be treated as fixed. Blocks tile through the design beginning at the first row and column. The final block in a row or a column will be truncated if it extends past the dimensions of the design. Uneven tiling is allowed.

3. Linear trend code, has a code word for the possible linear covariates that may be included in the objective.

   - `NONE`. The default. There are no linear trends across rows or columns.
   - `ROW`. A linear trend across rows is present. A covariate based on row number is included in the nuisance fixed effects.
   - `COLUMN`. A linear trend across columns is present. A covariate based on column number is included in the nuisance fixed effects.
   - `BOTH`. Linear trends acrosss both rows and columns are present in the design.

4. Correlation between rows. There are three models in DiGGer, two of which require a parameter to be specified.

   - `ID` for identity or independent errors between rows. No parameter is required.
   - `AR` for AutoRegressive errors between rows. A parameter $-1.0 < \rho < 1.0$ must be supplied. The AR1 model between r rows with parameter

ρ has the correlation pattern:

$$\begin{bmatrix} 1 & \rho & \rho^2 & \cdots & \rho^{r-1} \\ \rho & 1 & \rho & \cdots & \rho^{r-2} \\ \rho^2 & \rho & 1 & \cdots & \rho^{r-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho^{r-1} & \rho^{r-2} & \rho^{r-3} & \cdots & 1 \end{bmatrix}$$

- `MA` for Moving Average errors between rows. The parameter $-0.5 < \rho < 0.5$ must be supplied. The MA1 model has the correlation pattern:

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

5. Correlation between columns. The same models as correlation between rows apply.

6. Spatial variance ratio $\gamma > 0$, weights the spatial variance matrix when it is added in Equation 3 to form $\mathbf{\Lambda}$.

7.4 DiGGer job control

The DiGGer job control statement consists of the key `END` or `NEW PHASE`. `END` causes DiGGer to end the search. `NEW PHASE` tells DiGGer that there is another search phase and new search phase settings will follow.

8 Other input files

There are three other files that provide information to DiGGer. The `.trt` file gives treatment information. The `.din` file may provide an initial design in matrix form. The `.swp` file may give swap codes to experimental plots to control allowable treatment interchanges within a search.

8.1 The `.trt` treatment information file

The `.trt` file is created when DiGGer is invoked if it is not present in the current directory. The file has a line for each treatment in the design. Each line has a dummy treatment name, a dummy treatment number, and a replication level and group code both set to 1.

A user provided `.trt` file must have four columns either `<TAB>` or space delimited.

- The first column contains a treatment name of up to 30 characters. This information is printed in the `ID` column of the `.lst` field book list output file.
- The second column contains a treatment number of up to 7 digits. This provides a label for the `ENTRY` column of the `.lst` file. This number is not used elsewhere in the design files.
- The third column contains the number of repetitions of the treatment in a replicate. For equally replicated designs this number will be 1. The sum of these numbers must be equal to the number of occupied plots in a replicate or must divide the number of occupied plots exactly.
- The fourth column has an integer treatment group code. This is used in the calculation of the design efficiency measure in all cases except the standard A-measure, `A++` which makes comparisons amongst all treatments.

The design numbers that are used in the search and appear in the `.lst` file, in the `TRT` column, and in the `.log` file are the sequential integers indicating the treatment order in the `.trt` file. Treatment 1 is the first treatment in the `.trt` file. Treatment n is the n^{th} treatment in the `.trt` file.

8.2 The `.din` initial design file

A `.din` file must be provided when the fourth line of the initialisation section of the input file is `READ`. The `.din` file is a matrix with rows and columns matching the layout specified in the initialisation section. The matrix is expected in the same form as the matrix in the `.log` file. This is lexicological order, row by row, with all the treatments in row 1 given before all the treatments in row 2 until the treatment in the last column of the last row is reached. The `.din` file contains design numbers as defined in Section 8.1.

8.3 The `.swp` swap code file

A swap file `.swp` may be provided to control allowable swaps in the design search. A swap file may be read if the final initialise line is `READ SWAP` where both an initial design and a swap file will be read. The swap file is a matrix of integer codes in the same order as the `.din` file (Section 8.2). Only plots with the same swap code can have their current treatments interchanged. Swap codes may be used to control the location of check varieties as in Example 10.6. Another possible use is to control interchanges within non-regular shaped replicate blocks.

9 DiGGer output files

There are three output files produced by DiGGer, with extensions `.out`, `.log` and `.lst`.

9.1 The DiGGer `.out` output file

The `.out` file gives details of the search settings and records details of the progress of the search. For each search phase, the design layout, swap and correlation block dimensions and the search objective details are listed. The initial design and A-measure are printed. The number of possible treatment/plot interchanges and the number of these considered for an escape from a local minimum are given. The search phase, swaps tested and A-value are given for each improvement to the best design found. The interchanges at last improvement may be useful when considering repeats of similar searches.

At the end of a search phase the best design found is printed. This design is the initial design for the next search phase.

9.2 The DiGGer `.log` design file

The `.log` file reports the weighted A-measure and the A-measure associated with each objective in the current search phase calculated on the recorded design. The design phase and the number of interchanges tested and the interchange at last improvement to the design are shown. The printed design is a matrix in lexicological order. The rows of the matrix are output one at a time and split into groups of 16 treatments if necessary. The numbers presented are design numbers where design number n is the n^{th} treatment in the `.trt` file.

The `.log` file is updated after each tenth of the total interchanges in each search phase has been tested.

9.3 The DiGGer `.lst` field book listing file

The `.lst` file gives a listing of the design in the `.log` file, as rows nested within columns. The file has fields labelled `UNIT`, `ID`, `ENTRY`, `ROW`, `RANGE`, `REP` and `TRT`. `ID` and `ENTRY` are the treatment names and numbers from the `.trt` file. `TRT` is the design number established by the order of treatments in the `.trt` file. Missing plots in the design are given the `ID` '`BLANK`' and have a zero `ENTRY` and `TRT` number.

10 Examples

The use of DiGGer is best illustrated using examples. Examples given are:

- A typical variety trial using simple DiGGer
  - without blocks in two directions
  - with blocks in two directions.
- An animal house experiment with two objectives.
- An early generation $p-$rep variety trial, with unequal replication.

- Unreplicated variety trial with grid controls.
- A laboratory trial for sequential testing over days.
- A disease screening trial with check grid entries.
- An irregular trial using missing plots.
- Restriction on randomisation in a pumpkin trial.

10.1 Variety trial for 26 entries

In this example a variety trial is required for 26 varieties and there are 78 plots available with layout 26 rows by 3 columns. Replicates must be 26 rows by 1 column for this trial.

10.1.1 Standard DiGGer correlation with no additional blocks

The standard DiGGer correlation has random rows, columns and error with AutoRegressive correlation between plots in rows and columns. The job is easy to run interactively from a command prompt window (Figure 9).

The file generated by simple DiGGer is shown in Figure 10. The initialisation section is highlighted in blue, specifying the number of treatments and the design layout. There is one search phase. The phase settings shown in red allow swaps within replicates 26×1, with correlation across the whole design, 26×3. The search will stop after 100000 interchanges within replicates have been tested. The standard DiGGer correlation shown in green has

- a single objective with weight 1.0
- three block factors
  - random rows (1×3) with variance ratio $\gamma_r = 1.0$
  - random columns (26×1) with variance ratio $\gamma_c = 0.5$
  - random error (1×1) with variance ratio $\gamma_e = 0.1$
- no linear covariates
- AutoRegressive correlation between rows with parameter $\rho_r = 0.5$
- AutoRegressive correlation between columns with parameter $\rho_c = 0.5$
- spatial variance ratio $\gamma_s = 1.0$

Note that the column blocks are redundant here because by definition columns hold replicates and are thus fully efficient. The only block control across columns is by row blocks.

The blockstructure for an analysis of this trial has two block factors $\texttt{Rep}_{26\times1} \equiv \texttt{Col}$ and $\texttt{Row}_{1\times3}$.

```
C: \test>  DiGGer v26r3d1

 Job start time: Thu Jul 20 14:13:21 2006
 DiGGer 1.0.0

 Inputs from the keyboard will be used to create
 the search input file:
 v26r3d1.in

 Simple DiGGer is for equally replicated designs.
 Do you want to run the simplified version?[y]

<Enter> or y

 How many varieties (treatments) in the trial?

26

 What are the dimensions of this experiment?
 Enter Rows then Columns(Ranges) in this experiment

26  3

 Enter Rows then Columns(Ranges) in each replicate

26  1

 Do you want replicates or blocks
 in 2 directions? [y]

n
```

Figure 9: Example 10.1.1 interactive DiGGer.

v26r3d1.in generated by Simple DiGGer

```
# Initialise Design ############
SEEDS                          # RNG seeds [-1 -1 [-1 -1]]
26   FIXED                     # No. treatments, [F-fixed/R-random [gamma]]
26 3   26 1                    # Layout: Design r c, Replicate r c
GENERATE                       # G-GENERATE/R-READ 1st design [S-SWAP file]

# Search Phase Settings #######
  26 1   26 3                  # Swap r c, Correlation r c
  A++                          # A-type
  0.0 100000                   # Target A-measure, Max. interchanges to test
  100                          # Search intensity, (low) 0 - 100 (high)
  1                            # Number of objectives
#   Objective Specifications ##
    1.0                        # Objective weight
      3                        #  No. of block factors
          1    3   1.00        #  Block r c gamma
         26    1   0.50        #  Block r c gamma
          1    1   0.10        #  Block r c gamma
      NONE                     #  None/Row/Column/Both Linear trend code
      AR  0.5                  #  Correlation between rows: Model [par.]
      AR  0.5                  #  Correlation between columns: Model [par.]
      1.0                      #  Spatial gamma

# DiGGer control ###############
END                            # N - NEW PHASE or E - END search here
```

Figure 10: Example 10.1.1 Variety trial with replicates in 1D.

Row	Range 1	2	3
1	4	20	19
2	19	17	11
3	22	11	20
4	17	9	1
5	14	16	6
6	26	1	14
7	16	22	26
8	25	14	2
9	2	21	22
10	21	24	3
11	3	19	23
12	11	25	24
13	18	3	25
14	24	8	18
15	10	26	4
16	1	18	10
17	5	10	8
18	8	13	15
19	15	2	5
20	12	15	17
21	9	4	12
22	20	7	21
23	23	12	7
24	7	6	9
25	6	23	13
26	13	5	16

Figure 11: Design produced from Figure 10, v26r3d1.in

$$\texttt{Rep} + \texttt{Row}$$

The design produced has some features that are common in spatially optimised trials where separable AR1 correlations are assumed. Figure 11 shows treatments 7, 10 and 15 in red, green and blue respectively. Each treatment occurs as a self-diagonal and knight's move separation spanning just three rows of the design. The row blocks across replicates ensure that treatments aren't repeated in rows but the properties of AR1 correlations encourage the clustering of these treatments in the knight-move/self-diagonal positions. Further, treatments 3, 12 and 18 shown in cyan, magenta and yellow respectively span four rows and each is related by two knight-move arrangements. Such clustering of treatments can be avoided by forcing blocking in two directions as in Section 10.1.2.

10.1.2 Standard DiGGer correlation with blocks in two directions

Suppose a response **y** is given to the final question in Figure 9 and blocks in two directions are specified. The file now generated by simple DiGGer is shown in Figure 12.

In this example rows need to be combined to form blocks across replicates. DiGGer chooses blocks in the second direction by first checking whether the number of replicates divides the number of rows evenly. In this case 3 will not divide 26 evenly but DiGGer will allow the final block to have one less row. This sets the block size as 9×3 for the first two row-blocks and 8×3 for the final row-block. Additional control is provided by the intersection blocks 9×1 formed by the reps and the row-blocks. More emphasis is given to the row-blocks, using a variance ratio of 1.0, than the intersection blocks with a variance ratio of 0.1. Note that this is an alternative method for nested blocks to that in Figure 4 which used the objective weights for two objectives to provide this emphasis. Very large designs may need the variance ratio method when two objectives increase the size of the correlation matrix beyond the limit given in Section 3. No spatial correlation is specified when establishing blocks.

The second search phase restricts treatment swaps to within intersection blocks and used the same standard DiGGer correlation across the design as in the previous example.

The block factors in this design are $\texttt{Rep}_{26\times1} \equiv \texttt{Col}$, $\texttt{Rblock}_{9\times3}$, $\texttt{Rep.Rblock}_{9\times1}$ and $\texttt{Row}_{1\times3}$ and the blockstructure in an analysis:

$$\texttt{Rep} + \texttt{Rblock} + \texttt{Rep.Rblock} + \texttt{Row}$$

Figure 13 highlights the treatments occuring twice in either of the first two row-blocks. The final row-block contains 24 of the 26 treatments.

10.1.3 Relative efficiency of Example 10.1 designs

The final A-value obtained for Example 10.1.1 was 0.458988. This is lower than the A-value, 0.477438 obtained under the same correlation structure in Example

v26r3d2.in generated by Simple DiGGer

```
# Initialise Design ###########
SEEDS                          # RNG seeds [-1 -1 [-1 -1]]
26   FIXED                     # No. treatments, [F-fixed/R-random [gamma]]
26 3   26 1                    # Layout: Design r c, Replicate r c
GENERATE                       # G-GENERATE/R-READ 1st design [S-SWAP file]

# Search Phase Settings #######
  26 1   26 3                  # Swap r c, Correlation r c
  A++                          # A-type
  0.0 10000                    # Target A-measure, Max. interchanges to test
  100                          # Search intensity, (low) 0 - 100 (high)
  1                            # Number of objectives
#   Objective Specifications ##
    1.0                        # Objective weight
      2                        #  No. of block factors
          9    3   1.00        #  Block r c gamma
          9    1   0.10        #  Block r c gamma
      NONE                     #  None/Row/Column/Both Linear trend code
      ID                       #  Correlation between rows: Model [par.]
      ID                       #  Correlation between columns: Model [par.]
      1.0                      #  Spatial gamma

# DiGGer control ###############
NEW PHASE                      # N - NEW PHASE or E - END search here

# Search Phase Settings #######
  9 1   26 3                   # Swap r c, Correlation r c
  A++                          # A-type
  0.0 90000                    # Target A-measure, Max. interchanges to test
  100                          # Search intensity, (low) 0 - 100 (high)
  1                            # Number of objectives
#   Objective Specifications ##
    1.0                        # Objective weight
      3                        #  No. of block factors
          1    3   1.00        #  Block r c gamma
         26    1   0.50        #  Block r c gamma
          1    1   0.10        #  Block r c gamma
      NONE                     #  None/Row/Column/Both Linear trend code
      AR  0.5                  #  Correlation between rows: Model [par.]
      AR  0.5                  #  Correlation between columns: Model [par.]
      1.0                      #  Spatial gamma

# DiGGer control ###############
END                            # N - NEW PHASE or E - END search here
```

Figure 12: Example 10.1.2 Variety trial blocks in 2D.

Row	Range 1	2	3
1	24	21	8
2	10	8	14
3	18	23	25
4	2	5	7
5	19	4	16
6	13	15	1
7	20	3	12
8	17	22	6
9	11	9	26
10	26	19	3
11	4	25	10
12	6	11	19
13	23	12	17
14	1	24	5
15	22	14	20
16	8	13	9
17	15	16	2
18	7	18	21
19	21	20	15
20	3	1	18
21	25	26	22
22	16	10	11
23	9	17	24
24	12	7	4
25	5	6	13
26	14	2	23

Figure 13: Design produced from Figure 12, v26r3d2.in

10.1.2. The relative efficiency of the design with blocks in two directions was 96% compared to the design with only rows as blocks in the second direction. Designs with lower A-values are more efficient than designs with higher A-values under the same layout and correlation specifications.

It is possible to calculate the efficiency of the design in Example 10.1.1 as a block design under the blocking structure of the first phase of the search in Example 10.1.2. The A-value of this block design with row-blocks and intersection blocks in the absence of correlation was 0.738915 for the final design in Example 10.1.1 compared with 0.699268 for the design in Example 10.1.2. This gives a relative efficiency of 95% for the design without the blocks in two directions.

The final design efficiencies are affected by blocking constraints imposed during the construction of the design. Which design is more efficient? The A-measure gives an efficiency measure for a particular correlation structure but the true correlation structure is never known. Blocking in two directions can be considered insurance that does not significantly reduce the efficiency of the spatial design.

By default the simple interactive DiGGer option blocks in two directions. This ensures that treatments are separated to some extent in both directions.

10.2 Animal house trial

This example is an animal house trial which needed a nested blocking structure and used a combined objective as the search criterion. In the trial two groups of 8 steers would be fed 3 diets of a possible 8 over the course of the trial. Two groups of 8 pens were used to house the animals, 8 of the pens were set up to make measurements, the other 8 were holding pens. When steers from one group were in the measurement pens, steers from the other group were in the holding pens. It was decided to confound the pen effects with cattle, so each holding pen was linked to a measurement pen and a steer from each group would be restricted to each holding and measurement pen pair.

DiGGer requires designs to be described in terms of rows and columns. This design could be set up as 8 rows by 6 columns. The columns represent the six feeding periods, with the first three columns corresponding to the three feeding periods for one group of 8 steers and the last three columns corresponding to the periods for the second group of 8 steers (Figure 15). Each column must contain a replicate of diets.

There are two sets of blocks in the rows of the design. Blocks of six represent holding pen/measurement pens. Blocks of three correspond to particular steers in the trial. Figure 14 shows the input file used to produce this design.

Three objectives are defined. Two are used in the search for the design, the other gives the A-value for the trial as a block design with blocksize 3 and is calculated out of interest. The objective weight for the blocksize 3 objective is 0.0 and it does not contribute to the A-measure. The first objective defines *fixed* blocks for the rows so that treatments are allocated efficiently to pens. A block variance ratio $\gamma \leq 0$ is used to signify fixed blocks. The third objective blocks animals within rows and allows an AR1 correlation to try to impose some

Animal house trial

```
# Initialise Design ############
SEEDS 12528 224 16380 7994     # RNG seeds [-1 -1 [-1 -1]]
8   FIXED                      # No. treatments, [F-fixed/R-random [gamma]]
8 6   8 1                      # Layout: Design r c, Replicate r c
GENERATE                       # G-GENERATE/R-READ 1st design [S-SWAP file]

# Search Phase Settings #######
  8 1   1 6                    # Swap r c, Correlation r c
  A++                          # A-type
  0.0 100000                   # Target A-measure, Max. interchanges to test
  100.0                        # Search intensity, (low) 0 - 100 (high)
  3                            # Number of objectives
#   Objective Specifications ##
    0.5                        # Objective weight
      1                        #  No. of block factors
          1    6   0.0         #  Block r c gamma
      NONE                     #  None/Row/Column/Both Linear trend code
      ID                       #  Correlation between rows: Model [par.]
      ID                       #  Correlation between columns: Model [par.]
      1.0                      #  Spatial gamma
#   Objective Specifications ##
    0.0                        # Objective weight
      1                        #  No. of block factors
          1    3   0.0         #  Block r c gamma
      NONE                     #  None/Row/Column/Both Linear trend code
      ID                       #  Correlation between rows: Model [par.]
      ID                       #  Correlation between columns: Model [par.]
      1.0                      #  Spatial gamma
#   Objective Specifications ##
    0.5                        # Objective weight
      1                        #  No. of block factors
          1    3   0.0         #  Block r c gamma
      NONE                     #  None/Row/Column/Both Linear trend code
      ID                       #  Correlation between rows: Model [par.]
      AR  0.5                  #  Correlation between columns: Model [par.]
      1.0                      #  Spatial gamma

# DiGGer control ###############
END                            # N - NEW PHASE or E - END search here
```

Figure 14: Multiple objectives in the Animal House trial.

neighbour balance to treatments within rows. These two objectives are weighted by 0.5.

Fixed blocks are used here for illustrative purposes only. In general, random blocks are preferred to avoid search failures when disconnected blocks are encountered during a search.

There are four blocking factors in this experiment: $\texttt{Group}_{8\times3}$, $\texttt{Period}_{8\times1}$, $\texttt{Pen}_{1\times6}$ and $\texttt{Steer}_{1\times3} \equiv \texttt{Group.Pen}$. The blockstructure to include in the analysis is:

$$\texttt{Group/Period} + \texttt{Pen/Steer} = \texttt{Group} + \texttt{Group.Period} + \texttt{Pen} + \texttt{Pen.Steer}$$

10.3 Early generation $p-$rep trial

This trial has differences in replication levels and uses group codes to distinguish between test lines and check varieties. The group codes must be given in a user supplied `.trt` file (Figure 17). The replication scheme is given in Table 1. This is an example of a $p-$rep trial of Cullis *et al.* [3]. Grid varieties of an unreplicated trial are replaced with duplicate plots of test line entries where possible. Test lines were assigned a treatment group code of 1, check varieties were assigned a treatment group code of 2. Check varieties provide baseline yield and quality information for the trial site. The aim of the experiment is to select promising test lines to carry forward to the next stage of the crop breeding program.

Entries	Entry Number	Reps	Group
96	1:96	1	1
28	97:124	2	1
3	125:127	6	2
2	128:129	5	2

Table 1: Replication scheme for early generation trial.

The trial area was 15 rows by 12 columns (ranges). This search has two phases (Figure 16). The first phase:

- divides the trial into column-blocks of 15 rows by 6 columns, row-blocks of 5 rows by 12 columns with objective weight 0.8
- weights the intersection blocks of 5 rows by 6 columns by 0.2
- uses `Agg` as the A-type. This optimises treatment comparisons between treatments from different groups and ensures that treatments from different groups are spread as evenly as possible between blocks.

The second search phase:

- uses the standard DiGGer rows, columns and error random blocks and AR1 correlations between rows and columns.

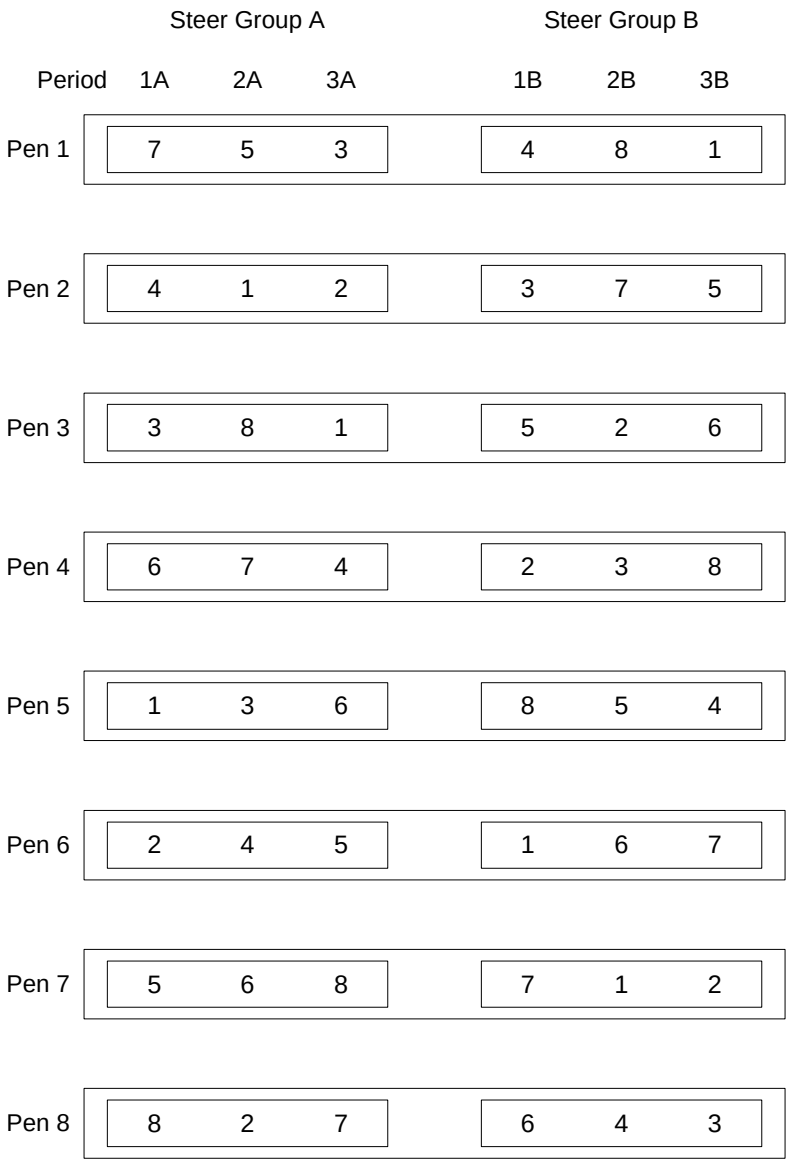


Figure 15: Design produced from Figure 14, rows are animals nested within pens, columns are feeding periods.

Early Generation trial

```
# Initialise Design ###########
SEEDS                          # RNG seeds [-1 -1 [-1 -1]]
129   FIXED                    # No. treatments, [F-fixed/R-random [gamma]]
15 12   15 12                  # Layout: Design r c, Replicate r c
GENERATE                       # G-GENERATE/R-READ 1st design [S-SWAP file]

# Search Phase Settings #######
  15 12   15 12                # Swap r c, Correlation r c
  Agg                          # A-type
  0.0 20000                    # Target A-measure, Max. interchanges to test
  100.0                        # Search intensity, (low) 0 - 100 (high)
  2                            # Number of objectives
#   Objective Specifications ##
    0.8                        # Objective weight
      2                        #  No. of block factors
          5   12   1.0         #  Block r c gamma - Rblock
         15    6   1.0         #  Block r c gamma - Cblock
      NONE                     #  None/Row/Column/Both Linear trend code
      ID                       #  Correlation between rows: Model [par.]
      ID                       #  Correlation between columns: Model [par.]
      1.0                      #  Spatial gamma
    0.2                        # Objective weight
      1                        #  No. of block factors
          5    6   1.0         #  Block r c gamma - Cblock.Rblock
      NONE                     #  None/Row/Column/Both Linear trend code
      ID                       #  Correlation between rows: Model [par.]
      ID                       #  Correlation between columns: Model [par.]
      1.0                      #  Spatial gamma

# DiGGer control ###############
NEW PHASE                      # N - NEW PHASE or E - END search here

# Search Phase Settings #######
  5 6   15 12                  # Swap r c, Correlation r c
  A11                          # A-type
  0.0 100000                   # Target A-measure, Max. interchanges to test
  100.0                        # Search intensity, (low) 0 - 100 (high)
  1                            # Number of objectives
#   Objective Specifications ##
    1.0                        # Objective weight
      3                        #  No. of block factors
          1   12   1.0         #  Block r c gamma - Row
         15    1   0.5         #  Block r c gamma - Col
          1    1   0.1         #  Block r c gamma - Error
      NONE                     #  None/Row/Column/Both Linear trend code
      AR  0.5                  #  Correlation between rows: Model [par.]
      AR  0.5                  #  Correlation between columns: Model [par.]
      1.0                      #  Spatial gamma

# DiGGer control ###############
END                            # N - NEW PHASE or E - END search here
```

Figure 16: Early generation trial using group code dependent A-measures.

p129.trt with Group codes

# TREATMENT NAME	NUMBER	REP	GRP
V1	1	1	1
...	..	.	.
V96	96	1	1
V97	97	2	1
...	..	.	.
V124	124	2	1
C1	125	6	2
C2	126	6	2
C3	127	6	2
C4	128	5	2
C5	129	5	2

Figure 17: Part of `p129.trt`.

- uses the `A11` A-type which optimises comparisons between Group 1 treatments, the test lines here, which are the treatments of interest.

The blockstructure for this design, with blocks named in Figure 16, is

$$\texttt{Cblock}_{15\times 6} + \texttt{Rblock}_{5\times 12} + \texttt{Cblock.Rblock}_{5\times 6} + \texttt{Row}_{1\times 12} + \texttt{Col}_{15\times 1}$$

The design produced is given in Figure 18. Duplicate test lines are shown in yellow, unreplicated test lines in grey. The check varieties are shown in other colours.

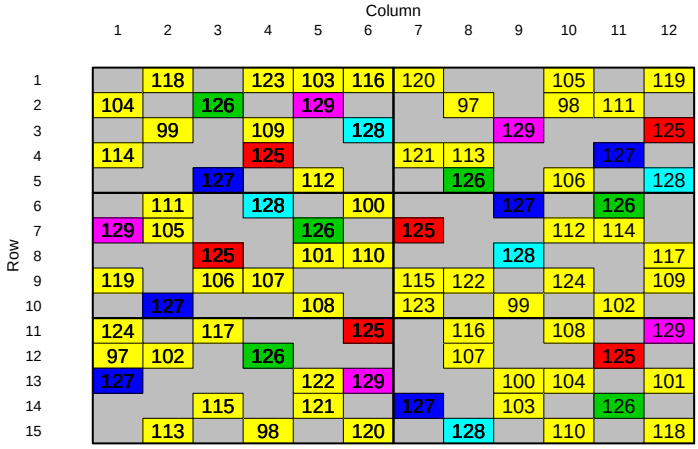


Figure 18: Design produced from Figure 16, p129.in

10.4 Unreplicated variety trial with grid controls

Sometimes it is not possible to replicate any of the test lines in a variety trial. This is a poor alternative to the $p-$rep design given in Section 10.3.

In this example 288 test lines (Group 1) were available as single plots and 5 grid varieties (Group 2) each with 18 replicates provided the control for variability in the field. The layout was 21 rows by 18 columns. The 90 grid controls can be neatly allocated to nine 7 row by 6 column blocks. The optimal allocation will have 2 replicates of each of the grid varieties in each block. The search in DiGGer requires two search phases and closely follows the previous example (Figure 16). The first phase uses the three blocking structures, $\texttt{Rblock}_{7\times18}$, $\texttt{Cblock}_{21\times6}$ and $\texttt{Rblock.Cblock}_{7\times6}$ in two objectives and the `Agg` A-measure.

Alternatively a starting design could be prepared as a 21×18 matrix with each 7×6 block holding 2 replicates of the grid varieties. The remaining spaces in the design would be filled with a randomisation of the unreplicated test lines. Using a file base name of u293, the starting design should be placed in `u293.din`. The initialise section of `u293.in` can then `READ` the initial design.

Once a suitable block design is established the search can restrict treatment interchanges to within 7×6 blocks and use the standard DiGGer correlation across the whole design, optimising comparisons between unreplicated entries.

In either case the blockstructure for the analysis is:

$$\texttt{Rblock} * \texttt{Cblock} + \texttt{Row} + \texttt{Col} = \texttt{Rblock} + \texttt{Cblock} + \texttt{Rblock.Cblock} + \texttt{Row} + \texttt{Col}$$

Figure 19 shows the resulting design with grid entries coloured and unreplicated lines in grey.

10.5 Sequential testing in a laboratory trial

Consider a field trial using a $p-$rep design (Section 10.3) where a selection of samples from the field are to be milled before further quality testing is undertaken. The field trial was 28 rows by 9 columns. There were 175 genotypes in the field trial to be taken through to the mill. Two parent varieties were at high replication, 45 test lines had 2 replicates and 128 test lines were unreplicated in the field. The mill can handle 7 samples per day with samples processed sequentially. There were 241 field plots with samples to be milled (Table 2).

The field samples of 46 test lines, numbers 47 to 128, were split to provide duplicate samples in the mill. This enables the variability within the milling process to be modelled. The resulting 287 samples were to be milled over 41 days at 7 samples per day.

Blocking factors considered for this experiment were:

- Replicate blocks, forming two groups of days each containing one of the 46 duplicate samples. A compromise was made, splitting the trial into two unequal blocks, 7×21 and 7×20.

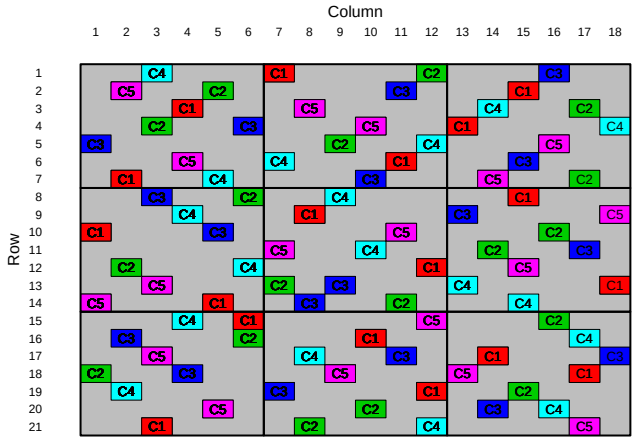


Figure 19: Unreplicated design showing the 5 replicated grid varieties

- Mill session, forming two groups across the mill days to reduce the concurrences of duplicate samples being milled early or late on mill days. The blocks were 4×41 and 3×41.
- Mill day, the most important source of block variability, 7×1.

The replicate blocks guard against a global drift in performance of the mill over time. The mill sessions help control for an observed drift in performance that occurs in a mill each day as the mill warms up. Day to day variability in the performance of the mill is captured by Mill day blocks.

The mill design has two types of replication:

- genuine replication of field samples. A sample from the field is split into two for processing in the mill.
- replication of genotypes. These have multiple field plots but each field plot provides a single sample for the mill.

The former have been selected from unreplicated genotypes in the field design. There is a one-to-one correspondence between their mill sample identifier and their genotype. The genotypes with multiple field plots do not have a one-to-one correspondence between their mill sample and genotype identifiers. Group codes are used to show the association between mill samples with the same genotype (Table 2). For example, there are 11 mill samples coded as group 56 for one of the parent genotypes, number 175.

Unreplicated genotypes from the field have field location information which may be used to separate field samples in the mill. In this case unreplicated field

Genotypes					Mill	
Type	No.	Field Plots			Samples	Group
Test	1:82	→	82	→	82	1-9 (FCol)
Test	47:128	→	46	↗	46	1-9 (FCol)
				↘	46	1-9 (FCol)
Test	129:173	↗	45	→	45	10-54 (Geno)
		↘	45	→	45	10-54 (Geno)
Parent	174	→	12	→	12	55 (Geno)
Parent	175	→	11	→	11	56 (Geno)
Total	175		241		287	

Table 2: Replication scheme and groups for the milling design.

genotypes are coded with their field column (FCol) 1 to 9 to help prevent some confounding of spatial field effects with mill effects. There are 56 groups in the design. The first 9 classify unreplicated field plots by field column. Codes 10 to 56 identify mill samples with the same genotype. DiGGer allows up to 100 groups codes in the `.trt` file.

The DiGGer search has three phases:

- Phase 1: Swap block 7×41, Correlation block 7×41, `Agg`.
  - Objective 1: Weight 0.8, Random blocks $\texttt{Rep}_{7\times 21}$ and $\texttt{Session}_{4\times 41}$
  - Objective 2: Weight 0.2, Random blocks $\texttt{Rep.Session}_{4\times 21}$
- Phase 2: Swap block 4×21, Correlation block 7×41, `Agg`.
  - Objective 1: Weight 0.5, Random blocks $\texttt{Day}_{7\times 1}$
  - Objective 2: Weight 0.5, Random blocks $\texttt{Session.Day}_{4\times 1}$
- Phase 3: Swap block 4×1, Correlation block 7×1, `A++`.
  - Objective 1: Weight 1.0, AR1 (0.5) correlation between samples within a Day.

The A-measure `Agg` in Phases 1 and 2 ensures that treatments from different treatment groups are spread as evenly as possible throughout the design. Spatial correlation is only present in the final phase. It describes correlation between samples on each day.

The blockstructure for this design is:

$$\texttt{Rep} + \texttt{Session} + \texttt{Rep.Session} + \texttt{Day} + \texttt{Session.Day}$$

Figure 20 shows the design produced. Samples are labelled with group codes. Samples in red are the 46 unreplicated genotypes with mill duplicates. Samples in yellow are field replicated genotypes with field samples unreplicated in the

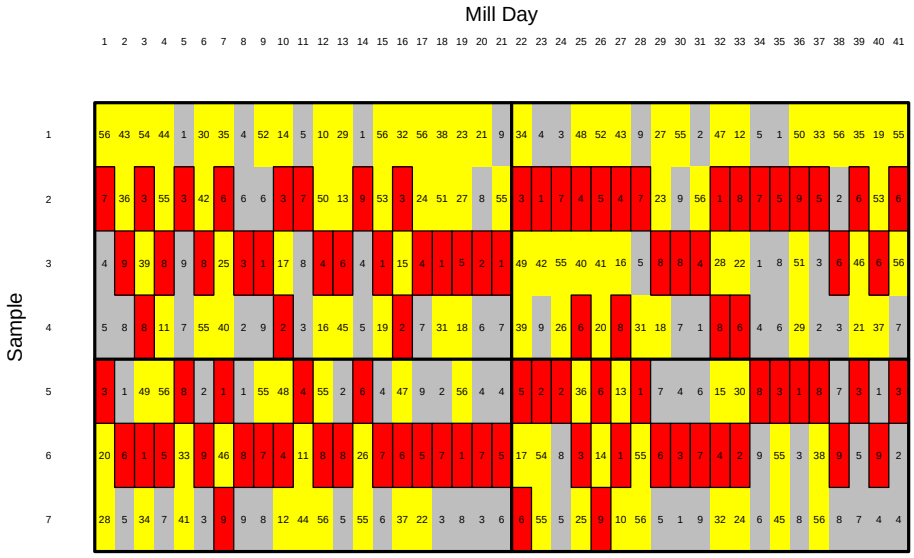


Figure 20: Mill design group codes, with duplicate mill samples in red, replicate field genotypes in yellow and unreplicated field and mill samples in grey.

mill. Grey samples are unreplicated genotypes unreplicated in the mill. The Rep.Session blocks are shown. There are two duplicate mill samples for 4 genotypes in each Rep block. There are two duplicate samples for 5 genotypes in the first Session block (4×41) and for 3 genotypes in the second Session block (3×41).

10.6 Regular check varieties in a disease screening nursery

In a disease screening nursery check varieties are required to be placed at high frequency throughout the trial. The check varieties have known susceptibility to the disease and provide an indicator of disease pressure. New lines of unknown susceptibility may be present as replicated or unreplicated lines. There is no efficiency measure in DiGGer which will ensure that check varieties at a high frequency will be evenly spread through a replicated trial. The approach to this problem is to generate a design for an unreplicated trial with the check varieties as grids as in Example 10.4. This determines optimal locations for the check varieties. The next step is to replace the unreplicated lines with replicated test lines and optimise their arrangement while preserving the locations of check varieties. A `.din` initial design file and a `.swp` swap restriction file are needed for the second DiGGer job.

Consider a trial with 118 test lines and 6 check varieties. The test lines are replicated 4 times. Trial dimensions are 40 rows by 16 columns with replicates 40 rows by 4 columns. A replicate holds 118 test lines and 7 replicates of 6 check varieties.

10.6.1 Step 1: Unreplicated check design

The best locations of check varieties are determined by optimising an unreplicated design for 472 unreplicated lines (Group 1) and 28 replicates of 6 check varieties (Group 2). The check varieties are further restricted to 7 replicates in each 40×4 block by creating a starting design and restricting treatment swaps to within these blocks. The R code to form a starting design is:

```
d0 <- matrix(0,40,16)
for (i in seq(4)){
   d0[,seq(4)+4*(i-1)] <- sample(rep(seq(7)-1,c(118,7,7,7,7,7,7)))
}
d0[d0==0] <- seq(472)+6
sink("u478.din")
write.matrix(d0)
sink()
```

A summary of the search is:

```
Design: 478 treatments; Design [40 16]; Rep [40 16] READ initial
  Phase 1: Swap [40 4]; Correlation [10 16] Agg
    Objective 1: Weight 0.8; Random block [10 16 1.0]
    Objective 2: Weight 0.2; Random block [10 4 1.0]
```

```
Phase 2: Swap [10 4] Correlation [40 18] A11
  Objective 1: Weight 1.0; Random block [40 1 0.1],
                           Random block [1 16 0.1],
                           ARrow(0.5), ARcol(0.5)
```

The first phase of the search optimises the nested block design, 10×16 and 10×4, using two objectives and the Group versus Group A-measure. The second phase of the search allows swaps within blocks 10×4 and the spatial objective is AR1 processes in rows and columns with random row and column blocks. The resulting design is shown in Figure 21.

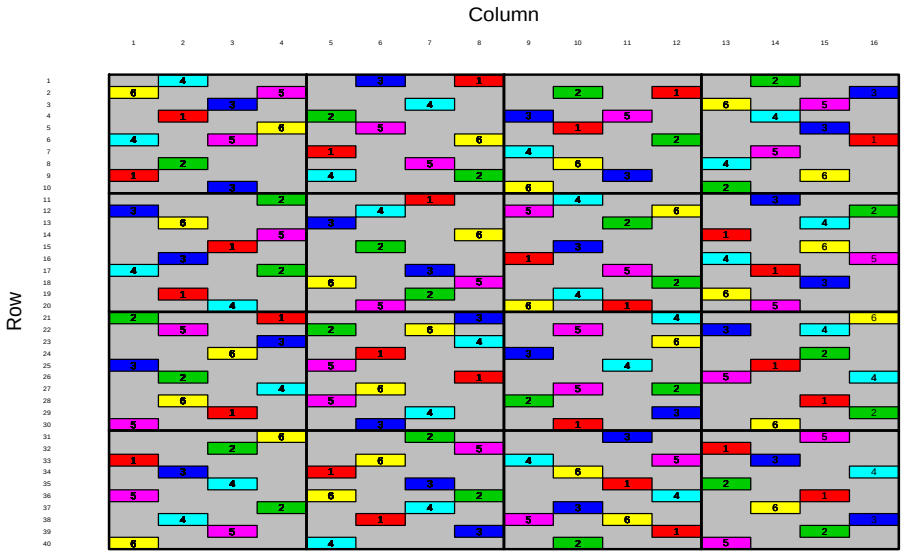


Figure 21: Check variety locations in the disease screening design.

10.6.2 Step 2: Replicated test lines with fixed Checks

Two files need to be created from the design found in the previous section. The first is a starting design, `d124.din` for the replicated trial. The unreplicated lines in the design are replaced with randomisations of the 118 test lines such that each 40×4 replicate contains a replicate of the test lines.

The second file, `d124.swp`, is a swap file to restrict swaps to between test lines while preserving the positions of the check varieties. The swap file is the design file from the previous section with unreplicated lines replaced by the swap code 7. All test lines are therefore eligible to be swapped with each other since they have the same swap code. Check lines will not be swapped because check varieties with the same swap code are the same check variety and such a swap will not change the design.

The search is essentially the same as in the previous section with these changes:

- there are 124 treatments
- `READ SWAP` tells DiGGer to read the initial design from `d124.din` and the swap file from `d124.swp`.
- the A-measure is `A++`, the standard A-measure, for both search phases
- the final objective is the standard DiGGer objective of random rows, columns and error with AR1 processes in rows and columns.

The block factors in this design are $\texttt{Rblock}_{10\times16}$, $\texttt{Cblock}_{40\times4} \equiv \texttt{Rep}$, $\texttt{Rblock.Cblock}_{10\times4}$, $\texttt{Row}_{1\times16}$ and $\texttt{Col}_{40\times1}$. The blockstructure in the analysis is:

$$\texttt{Rblock} * \texttt{Cblock} + \texttt{Row} + \texttt{Col} = \texttt{Rblock} + \texttt{Cblock} + \texttt{Rblock.Cblock} + \texttt{Row} + \texttt{Col}$$

10.7 Missing plots in a design

What happens when the trial area is made of plots in an incomplete rectangular array? Consider a two replicate trial to be run in a tub that can hold 18 strips with each strip having 16 locations for treatments. The design area is 18 rows by 16 columns, with each row being a strip of 16. In this experiment there are 2 replicates of 138 treatments. If a replicate is 9 strips there will be 6 locations within each replicate that do not have treatments. It was decided to leave the last 6 positions on strips 9 and 18 blank.

This design can be run in DiGGer by reading in a starting design with the positions corresponding to missing treatments labelled as zero. There is a restriction in DiGGer that the correlation block must be the size of the design when there are missing plots.

The search optimises the natural blocking term, 1×16 strips before optimising the arrangement of treatments within strips under the standard DiGGer spatial correlation.

A summary of the search is:

```
Design: 138 treatments; Design [18 16]; Rep [9 16] READ initial
  Phase 1: Swap [9 16]; Correlation [18 16] A++
    Objective 1: Weight 1.0; Random block [1 16 1.0]
  Phase 2: Swap [1 16] Correlation [18 16] A++
    Objective 1: Weight 1.0; Random block [1 16 1.0],
                             Random block [18 1 0.5],
                             Random block [1 1 0.1],
                             ARrow(0.5), ARcol(0.5)
```

The final design (Figure 22) shows the positions of missing plots in the design.

Strip \ Position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	138	83	66	130	120	62	5	106	34	57	86	131	43	110	59	74
2	18	35	89	67	15	56	73	46	37	135	95	7	72	127	38	47
3	97	104	91	30	79	88	25	123	14	8	48	21	36	98	58	4
4	103	109	32	118	132	105	16	112	13	80	45	108	69	92	31	26
5	53	49	82	24	115	39	137	68	102	78	111	96	116	64	107	28
6	10	22	136	63	70	3	6	133	65	85	125	61	122	12	9	81
7	100	2	71	27	20	54	126	42	134	17	19	29	60	51	128	93
8	94	23	114	40	1	117	90	52	55	113	129	84	119	44	76	75
9	99	11	33	124	41	87	121	101	77	50	0	0	0	0	0	0
10	109	47	31	25	39	65	138	111	71	101	75	134	7	59	12	1
11	77	53	40	132	22	20	56	57	90	102	104	122	128	4	72	45
12	89	5	17	3	119	32	68	38	43	2	63	48	55	41	92	97
13	79	133	49	66	87	24	34	21	35	103	51	85	16	100	129	127
14	73	114	36	42	80	6	99	117	123	86	96	19	50	118	18	115
15	8	88	67	11	116	37	84	126	10	82	26	52	110	61	120	112
16	106	95	13	58	94	74	70	30	78	9	54	121	28	69	44	29
17	60	98	46	136	130	124	135	76	108	14	137	23	125	62	93	107
18	33	81	83	113	27	15	131	105	91	64	0	0	0	0	0	0

Figure 22: Design with 12 missing (0) plots.

In this design the only blocks across replicates are the 18×1 blocks representing position. There are a number of treatments in this two replicate design where the treatments are in adjacent positions. For example treatment 109 is at strip 4 position 2 in the first replicate and strip 10 position 1 in the second replicate. Better control over the positional distribution of treatments in this design could be obtained by adding an initial phase with blocks 18×4 and sub-blocks 9×4, or blocks 18×8 and sub-blocks 9×8.

10.8 Restricted randomisation for unruly pumpkins

In this experiment, 3 varieties of pumpkin were to be grown at 4 densities on two beds of 24 plots. Pumpkins will spread into neighbouring plots and this will produce an identification problem at harvest if adjacent plots have the same variety. The beds will be separated by a buffer sown to a distinctive variety so that spreading to adjacent plots between beds will not be a problem. The experimenter requires that adjacent plots within beds are sown to different varieties.

This experiment can be designed in two steps. The first step, with file base name `PumpkinVar`, ignores density and finds the best arrangement of varieties when there is an AR1 correlation within beds. The AR1 model discourages the same treatment occurring in adjacent plots.

A summary of the search is:

```
Design: 3 treatments; Design [2 24]; Rep [1 12] GENERATE initial
  Phase 1: Swap [1 12]; Correlation [2 24] A++
    Objective 1: Weight 1.0; AR 0.5 between columns
```

There are 4 repeats of each variety in each replicate.

For the second step of the search, with file base name `Pumpkin`, allocate 4 densities to each variety within each replicate of the design obtained in the first step. Read this initial design, `Pumpkin.din` and restrict treatment swaps to within variety within replicate. The matrix from `PumpkinVar.log` has the swap codes needed to restrict swaps to plots with the same variety. Copy this matrix to `Pumpkin.swp`. Find the best design under standard correlation.

A summary of the search is:

```
Design: 12 treatments; Design [2 24]; Rep [1 12] READ SWAP
  Phase 1: Swap [1 12]; Correlation [2 24] A++
    Objective 1: Weight 1.0; Random block [1 24 1.0],
                             Random block [2  1 0.5],
                             Random block [1  1 0.1],
                             ARrow(0.5), ARcol(0.5)
```

The final design shown in Figure 23 shows varieties as different colours and treatment labels as `Variety.Density`. The design has restricted the randomisation such that varieties do not occur in adjacent plots within beds but has not restricted the position of varieties between beds.

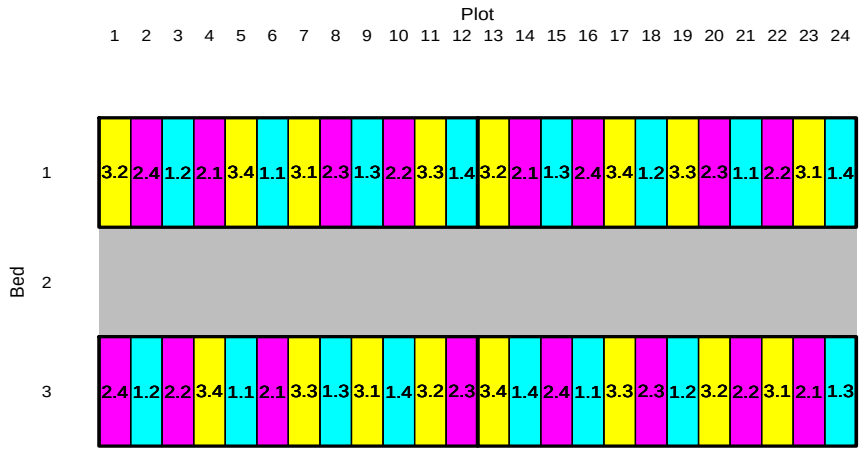


Figure 23: Pumpkin design: `Variety.Density`.

References

[1] R. Battiti and G. Tecchiolli, *The reactive tabu search*, ORSA Journal on Computing **6** (1994), no. 2, 126–140.

[2] N.E. Coombes, *The reactive tabu search for efficient correlated experimental designs*, Ph.D. thesis, Liverpool John Moores University, 2002.

[3] B.R. Cullis, A.B. Smith, and N.E. Coombes, *On the design of early generation variety trials with correlated data*, Journal of Agricultural, Biological and Environmental Statistics **00** (2007), 000–000.

[4] A.R. Gilmour, B.R. Cullis, and A.P. Verbyla, *Accounting for natural and extraneous variation in the analysis of field experiments*, Journal of Agricultural, Biological, and Environmental Statistics **2** (1997), no. 3, 269–293.

[5] R.J. Martin and J.A. Eccleston, *Efficient block designs for correlated observations*, Australian Journal of Statistics **33** (1991), no. 3, 299–311.