

# Crossover - A search algorithm and GUI for cross-over designs

Kornelius Rohmeyer

September 25, 2014

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Installation . . . . .	2
1.2	GUI Overview . . . . .	2
1.2.1	Catalogue . . . . .	2
1.2.2	Algorithm Search . . . . .	3
1.2.3	Input own design . . . . .	3
<b>2</b>	<b>Models</b>	<b>3</b>
2.1	Standard additive model . . . . .	4
2.2	Full set of interactions . . . . .	6
2.3	Self-adjacency model . . . . .	7
2.4	Placebo model . . . . .	8
2.5	No carry-over into self model . . . . .	9
2.6	Treatment decay model . . . . .	10
2.7	Proportionality model . . . . .	11
2.8	Second-order carry-over effects . . . . .	11
<b>3</b>	<b>Search strategy</b>	<b>14</b>
<b>4</b>	<b>Random Subject Effects Model</b>	<b>19</b>
	<b>Index</b>	<b>20</b>
	<b>Literatur</b>	<b>21</b>
	<b>Table of Symbols</b>	<b>22</b>

# 1 Introduction

This package provides more than two hundred cross-over design from literature, a search algorithm to find efficient cross-over designs for various models and a graphical user interface (GUI) to find/generate appropriate designs.

The computationally intensive parts of the package, i.e. the search algorithm, is written using the R packages Rcpp and RcppArmadillo (Eddelbuettel and François [2011] and Eddelbuettel and Sanderson [2013]). The GUI is written in Java and uses package rJava (Urbanek [2013]).

## 1.1 Installation

Once it is installed, whenever you start R you can load the Crossover package by entering `library(Crossover)` into the R Console. The graphical user interface as shown in figure 1 is started with the command `CrossoverGUI()`.

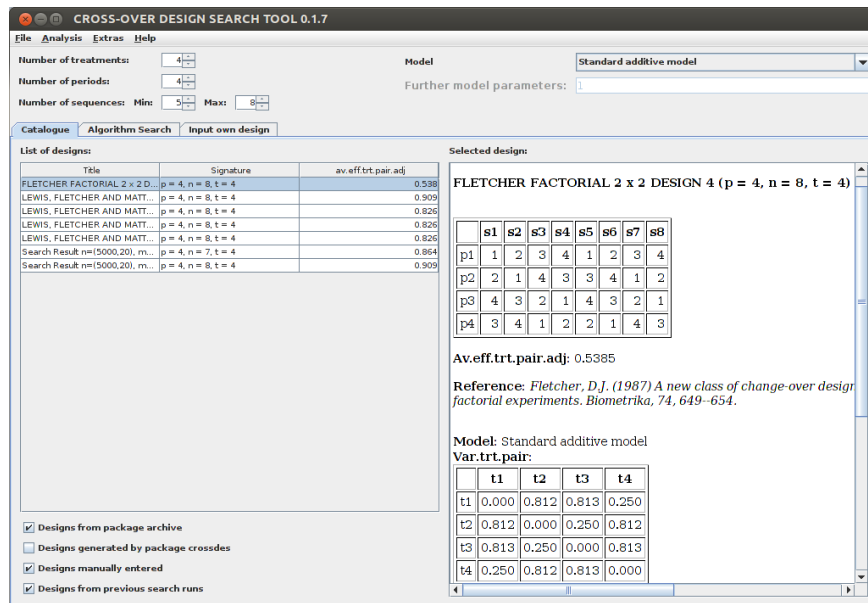


Figure 1: Cross-Over Design GUI.

## 1.2 GUI Overview

### 1.2.1 Catalogue

The catalogue, collected and compiled by Professor Byron Jones, contains 241 designs from the following literature: Anderson and Preece [2002], Archdeacon et al. [1980], Atkinson [1966], Balaam [1968], Berenblut [1964], Blaisdell Jr and Raghavarao [1980], Davis and Hall [1969], Federer and Atkinson [1964], Fletcher [1987], Iqbal and Jones [1994], Lewis et al. [1988], Cochran et al. [1941], Patterson and Lucas [1962], Pigeon [1985], Prescott [1999], Quenouille [1953], Russell [1991], Lucas [1956], Williams [1949], Prescott [1994], Bate and Jones [2002]

Depending on the numbers of treatments, periods and range of sequence you specify in the GUI appropriate designs from the catalogue will be shown.<sup>1</sup> Further functions from package `crossdes` (Sailer [2013]) are called to create designs for the specified values if possible.

<sup>1</sup>You can get a table referencing all designs and the respective number of treatments, periods and sequences by calling `buildSummaryTable()`.

In figure 1 you can see the following four checkboxes, that allow you to see only specific subsets:

**Designs from package archive** The previously noted designs from literature are shown.

**Designs generated by package crossdes** Activating this option will result in short delays when displaying the catalogue, since the crossdes algorithms are called.

**Designs manually entered** All designs entered on tab "*Input own design*" are shown.

**Designs from previous search runs** All designs from previous search runs are shown.

### 1.2.2 Algorithm Search

In figure 2 the preliminary graphical interface for the search algorithm is shown with the following options:

- The drop-down menu for *model* let you specify which model you are interested. These models are described in detail in section 2.
- In case of the placebo or proportionality model you can specify *further model parameters* (namely the number of placebos and the proportionality parameter, respectively).
- Specify the exact *number of sequences*. (The number of treatments and periods is already specified in the top panel of the GUI.)
- Optionally specify the *exact number of treatment assignments*. The GUI default is to let the algorithm figure out good/optimal assignments. But depending on further information (information from theoretical results or treatments more important than the others<sup>2</sup>, etc.) the number of treatment assignments can be specified.
- You can specify that the design should be constructed in a way that *in each sequence/period a treatment occur as evenly as possible*. This restriction will normally decrease the efficiency of the algorithm.
- The GUI default is an all-pair comparison of all treatments with equal weights. Change the *contrast weights* accordingly if you are interested in other contrasts or different weights.  
*User defined contrasts* can be used and a R matrix of  $n$  contrasts (i.e. of dimension  $n \times v$ ) will be used.
- Pressing the "*Compute Design*" button will start the search algorithm described in section 3. After a few seconds the result will be shown in the previous empty text area on the right.

### 1.2.3 Input own design

This tab provides you with the possibility to analyse your own designs easily or use them as starting points for the search algorithm.

## 2 Models

The package and GUI support the following eight models.

---

<sup>2</sup>Different weights of treatment importance should be specified as weighted contrasts. See item *contrasts*.

Figure 2: Panel for algorithmic search of cross-over designs.

## 2.1 Standard additive model

$$Y_{ijk} = \mu + \pi_j + \tau_{d[i,j]} + \lambda_{d[i,j-1]} + s_{ik} + e_{ijk}$$

with<sup>3</sup>

$\mu$  intercept,

$\pi_j$  period effect for period  $j$ ,

$\tau_{d[i,j]}$  direct treatment effect for treatment  $d[i,j]$  in period  $j$  of sequence  $i$ ,

$\lambda_{d[i,j-1]}$  first-order carry-over effect (0 for  $j-1=0$ ),

$s_{ik}$   $k$ th subject effect on sequence  $i$ ,

$e_{ijk}$  random error with zero mean and variance  $\sigma^2$ .

which we can write as

$$E(Y) = \mu + X \begin{pmatrix} \tau \\ \lambda \end{pmatrix} + Z \begin{pmatrix} \pi \\ s \end{pmatrix}$$

with  $X$  and  $Z$  called the *treatment* and *block design matrices*, respectively.

We call  $H$  a *link matrix* if  $X = X_r H$  where  $X_r$  is the design matrix for the row-column design. The rows of the link matrix specify all possible parameter combinations. Therefore  $H$  has as many columns as there are parameters and in the row-column-setting  $i$  parameter  $j$  is included  $h_{ij}$  times.

```
# Design:
design <- rbind(c(3,2,1),
               c(2,1,3),
               c(1,2,3),
               c(3,2,1))

design
```

<sup>3</sup>cf. Jones and Kenward [2003], page 8

```

##      [,1] [,2] [,3]
## [1,]    3    2    1
## [2,]    2    1    3
## [3,]    1    2    3
## [4,]    3    2    1

v <- 3 # number of treatments
# Link matrix:
H <- Crossover:::linkMatrix(model="Standard additive model", v)
H

##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]    1    0    0    0    0    0
## [2,]    0    1    0    0    0    0
## [3,]    0    0    1    0    0    0
## [4,]    1    0    0    1    0    0
## [5,]    0    1    0    1    0    0
## [6,]    0    0    1    1    0    0
## [7,]    1    0    0    0    1    0
## [8,]    0    1    0    0    1    0
## [9,]    0    0    1    0    1    0
## [10,]   1    0    0    0    0    1
## [11,]   0    1    0    0    0    1
## [12,]   0    0    1    0    0    1

# Row-Column-Design: (cf. John et al. 2004, Table II and page 2649f.)
rcDesign <- Crossover:::rcd(design, v=v, model=1)
rcDesign

##      [,1] [,2] [,3]
## [1,]    3    2    1
## [2,]   11    7    6
## [3,]    7    5   12
## [4,]    6    8   10

# Design Matrix of Row-Column Design:
Xr <- Crossover:::rcdMatrix(rcDesign, v, model=1)
Xr

##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12]
## [1,]    0    0    1    0    0    0    0    0    0    0    0    0
## [2,]    0    1    0    0    0    0    0    0    0    0    0    0
## [3,]    1    0    0    0    0    0    0    0    0    0    0    0
## [4,]    0    0    0    0    0    0    0    0    0    0    1    0
## [5,]    0    0    0    0    0    0    1    0    0    0    0    0
## [6,]    0    0    0    0    0    1    0    0    0    0    0    0
## [7,]    0    0    0    0    0    0    1    0    0    0    0    0
## [8,]    0    0    0    0    1    0    0    0    0    0    0    0
## [9,]    0    0    0    0    0    0    0    0    0    0    0    1

```

```
## [10,] 0 0 0 0 0 1 0 0 0 0 0 0
## [11,] 0 0 0 0 0 0 0 1 0 0 0 0
## [12,] 0 0 0 0 0 0 0 0 0 1 0 0

# Design Matrix of Cross-Over Design:
X <- Xr %*% H
X

##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,] 0 0 1 0 0 0
## [2,] 0 1 0 0 0 0
## [3,] 1 0 0 0 0 0
## [4,] 0 1 0 0 0 1
## [5,] 1 0 0 0 1 0
## [6,] 0 0 1 1 0 0
## [7,] 1 0 0 0 1 0
## [8,] 0 1 0 1 0 0
## [9,] 0 0 1 0 0 1
## [10,] 0 0 1 1 0 0
## [11,] 0 1 0 0 1 0
## [12,] 1 0 0 0 0 1
```

## 2.2 Full set of interactions

This model has further interaction parameters  $\gamma_{ij}$  and a design matrix  $X$  with  $2 * v + v^2$  columns:<sup>4</sup>

$$E(Y) = \mu + X \begin{pmatrix} \tau \\ \lambda \\ \gamma \end{pmatrix} + Z \begin{pmatrix} \pi \\ s \end{pmatrix}$$

```
H <- Crossover:::linkMatrix(model="Full set of interactions", v)
H

##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14] [,15]
## [1,] 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [2,] 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
## [3,] 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
## [4,] 1 0 0 1 0 0 1 0 0 0 0 0 0 0 0
## [5,] 0 1 0 1 0 0 0 0 0 1 0 0 0 0 0
## [6,] 0 0 1 1 0 0 0 0 0 0 0 0 1 0 0
## [7,] 1 0 0 0 1 0 0 1 0 0 0 0 0 0 0
## [8,] 0 1 0 0 1 0 0 0 0 0 1 0 0 0 0
## [9,] 0 0 1 0 1 0 0 0 0 0 0 0 0 1 0
## [10,] 1 0 0 0 0 1 0 0 1 0 0 0 0 0 0
## [11,] 0 1 0 0 0 1 0 0 0 0 0 1 0 0 0
## [12,] 0 0 1 0 0 1 0 0 0 0 0 0 0 0 1
```

<sup>4</sup>Personally I think some of the overspecified parameters should be dropped, for example  $\gamma_{ii}$  - but I'll take a look at this later.

```
# Design Matrix of Cross-Over Design:
X <- Xr %*% H
X

##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14] [,15]
## [1,]    0    0    1    0    0    0    0    0    0    0    0    0    0    0    0
## [2,]    0    1    0    0    0    0    0    0    0    0    0    0    0    0    0
## [3,]    1    0    0    0    0    0    0    0    0    0    0    0    0    0    0
## [4,]    0    1    0    0    0    1    0    0    0    0    0    1    0    0    0
## [5,]    1    0    0    0    1    0    0    1    0    0    0    0    0    0    0
## [6,]    0    0    1    1    0    0    0    0    0    0    0    0    1    0    0
## [7,]    1    0    0    0    1    0    0    1    0    0    0    0    0    0    0
## [8,]    0    1    0    1    0    0    0    0    0    1    0    0    0    0    0
## [9,]    0    0    1    0    0    1    0    0    0    0    0    0    0    0    1
## [10,]   0    0    1    1    0    0    0    0    0    0    0    0    1    0    0
## [11,]   0    1    0    0    1    0    0    0    0    0    1    0    0    0    0
## [12,]   1    0    0    0    0    1    0    0    1    0    0    0    0    0    0
```

## 2.3 Self-adjacency model

In the *self-adjacency model*  $\lambda_i$  is replaced by  $\varphi_i$  in case of carry-over into itself. The case  $\varphi = 0$  represents the *no carry-over into self model*.

$$E(Y) = \mu + X \begin{pmatrix} \tau \\ \lambda \\ \varphi \end{pmatrix} + Z \begin{pmatrix} \pi \\ s \end{pmatrix}$$

```
H <- Crossover:::linkMatrix(model="Self-adjacency model", v)
H

##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]
## [1,]    1    0    0    0    0    0    0    0    0
## [2,]    0    1    0    0    0    0    0    0    0
## [3,]    0    0    1    0    0    0    0    0    0
## [4,]    1    0    0    0    0    0    1    0    0
## [5,]    0    1    0    1    0    0    0    0    0
## [6,]    0    0    1    1    0    0    0    0    0
## [7,]    1    0    0    0    1    0    0    0    0
## [8,]    0    1    0    0    0    0    0    1    0
## [9,]    0    0    1    0    1    0    0    0    0
## [10,]   1    0    0    0    0    1    0    0    0
## [11,]   0    1    0    0    0    1    0    0    0
## [12,]   0    0    1    0    0    0    0    0    1

# Design Matrix of Cross-Over Design:
X <- Xr %*% H
X

##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]
```

```
## [1,] 0 0 1 0 0 0 0 0 0 0
## [2,] 0 1 0 0 0 0 0 0 0 0
## [3,] 1 0 0 0 0 0 0 0 0 0
## [4,] 0 1 0 0 0 1 0 0 0 0
## [5,] 1 0 0 0 1 0 0 0 0 0
## [6,] 0 0 1 1 0 0 0 0 0 0
## [7,] 1 0 0 0 1 0 0 0 0 0
## [8,] 0 1 0 1 0 0 0 0 0 0
## [9,] 0 0 1 0 0 0 0 0 0 1
## [10,] 0 0 1 1 0 0 0 0 0 0
## [11,] 0 1 0 0 0 0 0 0 1 0
## [12,] 1 0 0 0 0 1 0 0 0 0
```

## 2.4 Placebo model

In the placebo model there are no carry-over effects for the placebo treatment(s).

$$E(Y) = \mu + X\left(\begin{smallmatrix} \tau \\ \lambda \end{smallmatrix}\right) + Z\left(\begin{smallmatrix} \pi \\ s \end{smallmatrix}\right)$$

```
# Link matrix:
H <- Crossover:::linkMatrix(model="Placebo model", v, placebos=1)
H

##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,] 1 0 0 0 0 0
## [2,] 0 1 0 0 0 0
## [3,] 0 0 1 0 0 0
## [4,] 1 0 0 0 0 0
## [5,] 0 1 0 0 0 0
## [6,] 0 0 1 0 0 0
## [7,] 1 0 0 0 1 0
## [8,] 0 1 0 0 1 0
## [9,] 0 0 1 0 1 0
## [10,] 1 0 0 0 0 1
## [11,] 0 1 0 0 0 1
## [12,] 0 0 1 0 0 1

# Design Matrix of Cross-Over Design:
X <- Xr %*% H
X

##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,] 0 0 1 0 0 0
## [2,] 0 1 0 0 0 0
## [3,] 1 0 0 0 0 0
## [4,] 0 1 0 0 0 1
## [5,] 1 0 0 0 1 0
```



```
## [6,] 0 0 1 0 0 0
## [7,] 1 0 0 0 1 0
## [8,] 0 1 0 0 0 0
## [9,] 0 0 1 0 0 1
## [10,] 0 0 1 0 0 0
## [11,] 0 1 0 0 1 0
## [12,] 1 0 0 0 0 1
```

## 2.5 No carry-over into self model

This model differs from the standard additive model in the assumption that in the *no carry-over into self model* no carry-over effect occurs if current and previous treatment are the same.

$$E(Y) = \mu + X\left(\begin{smallmatrix} \tau \\ \lambda \end{smallmatrix}\right) + Z\left(\begin{smallmatrix} \pi \\ s \end{smallmatrix}\right)$$

```
H <- Crossover:::linkMatrix(model="No carry-over into self model", v)
```

```
H
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,] 1 0 0 0 0 0
## [2,] 0 1 0 0 0 0
## [3,] 0 0 1 0 0 0
## [4,] 1 0 0 0 0 0
## [5,] 0 1 0 1 0 0
## [6,] 0 0 1 1 0 0
## [7,] 1 0 0 0 1 0
## [8,] 0 1 0 0 0 0
## [9,] 0 0 1 0 1 0
## [10,] 1 0 0 0 0 1
## [11,] 0 1 0 0 0 1
## [12,] 0 0 1 0 0 0
```

```
# Design Matrix of Cross-Over Design:
```

```
X <- Xr %*% H
```

```
X
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,] 0 0 1 0 0 0
## [2,] 0 1 0 0 0 0
## [3,] 1 0 0 0 0 0
## [4,] 0 1 0 0 0 1
## [5,] 1 0 0 0 1 0
## [6,] 0 0 1 1 0 0
## [7,] 1 0 0 0 1 0
## [8,] 0 1 0 1 0 0
## [9,] 0 0 1 0 0 0
## [10,] 0 0 1 1 0 0
```

```
## [11,]    0    1    0    0    0    0
## [12,]    1    0    0    0    0    1
```

## 2.6 Treatment decay model

In contrast to the no carry-over into self model in the *treatment decay model* we assume there are only (negative) carry-over effects if the current and previous treatment are the same.

$$E(Y) = \mu + X\left(\begin{smallmatrix} \tau \\ \lambda \end{smallmatrix}\right) + Z\left(\begin{smallmatrix} \pi \\ s \end{smallmatrix}\right)$$

```
H <- Crossover:::linkMatrix(model="Treatment decay model", v)
```

```
H
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]    1    0    0    0    0    0
## [2,]    0    1    0    0    0    0
## [3,]    0    0    1    0    0    0
## [4,]    1    0    0   -1    0    0
## [5,]    0    1    0    0    0    0
## [6,]    0    0    1    0    0    0
## [7,]    1    0    0    0    0    0
## [8,]    0    1    0    0   -1    0
## [9,]    0    0    1    0    0    0
## [10,]   1    0    0    0    0    0
## [11,]    0    1    0    0    0    0
## [12,]    0    0    1    0    0   -1
```

```
# Design Matrix of Cross-Over Design:
```

```
X <- Xr %*% H
```

```
X
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]    0    0    1    0    0    0
## [2,]    0    1    0    0    0    0
## [3,]    1    0    0    0    0    0
## [4,]    0    1    0    0    0    0
## [5,]    1    0    0    0    0    0
## [6,]    0    0    1    0    0    0
## [7,]    1    0    0    0    0    0
## [8,]    0    1    0    0    0    0
## [9,]    0    0    1    0    0   -1
## [10,]   0    0    1    0    0    0
## [11,]   0    1    0    0   -1    0
## [12,]    1    0    0    0    0    0
```

## 2.7 Proportionality model

For the proportionality model we have no separate carry-over effects, but assume for period  $i > 1$  an additive proportion  $p$  of the effect  $\tau_{d(i-1,j)}$  from the previous treatment  $d(i-1, j)$ . This model is non-linear in  $p$ , therefore we assume  $p$  is known.

$$E(Y) = \mu + X_p\tau + Z\left(\frac{\pi}{s}\right)$$

```
H <- Crossover:::linkMatrix(model="Proportionality model", v)
H

##           [,1] [,2] [,3]
## [1,]    1.0    0.0    0.0
## [2,]    0.0    1.0    0.0
## [3,]    0.0    0.0    1.0
## [4,]    1.5    0.0    0.0
## [5,]    0.5    1.0    0.0
## [6,]    0.5    0.0    1.0
## [7,]    1.0    0.5    0.0
## [8,]    0.0    1.5    0.0
## [9,]    0.0    0.5    1.0
## [10,]   1.0    0.0    0.5
## [11,]   0.0    1.0    0.5
## [12,]   0.0    0.0    1.5

# Design Matrix of Cross-Over Design:
X <- Xr %*% H
X

##           [,1] [,2] [,3]
## [1,]    0.0    0.0    1.0
## [2,]    0.0    1.0    0.0
## [3,]    1.0    0.0    0.0
## [4,]    0.0    1.0    0.5
## [5,]    1.0    0.5    0.0
## [6,]    0.5    0.0    1.0
## [7,]    1.0    0.5    0.0
## [8,]    0.5    1.0    0.0
## [9,]    0.0    0.0    1.5
## [10,]   0.5    0.0    1.0
## [11,]   0.0    1.5    0.0
## [12,]   1.0    0.0    0.5
```

## 2.8 Second-order carry-over effects

In the model with second-order carry-over effects we do have another vector  $\lambda_2$  of carry-over effects:

$$E(Y) = \mu + X \begin{pmatrix} \tau \\ \lambda_1 \\ \lambda_2 \end{pmatrix} + Z \begin{pmatrix} \pi \\ s \end{pmatrix}$$

Therefore the link matrix has  $3 * v$  columns.

```
# Link matrix:
H <- Crossover:::linkMatrix(model="Second-order carry-over effects", v)
H
```

##		[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]
##	[1,]	1	0	0	0	0	0	0	0	0
##	[2,]	0	1	0	0	0	0	0	0	0
##	[3,]	0	0	1	0	0	0	0	0	0
##	[4,]	1	0	0	1	0	0	0	0	0
##	[5,]	0	1	0	1	0	0	0	0	0
##	[6,]	0	0	1	1	0	0	0	0	0
##	[7,]	1	0	0	0	1	0	0	0	0
##	[8,]	0	1	0	0	1	0	0	0	0
##	[9,]	0	0	1	0	1	0	0	0	0
##	[10,]	1	0	0	0	0	1	0	0	0
##	[11,]	0	1	0	0	0	1	0	0	0
##	[12,]	0	0	1	0	0	1	0	0	0
##	[13,]	1	0	0	1	0	0	1	0	0
##	[14,]	0	1	0	1	0	0	1	0	0
##	[15,]	0	0	1	1	0	0	1	0	0
##	[16,]	1	0	0	0	1	0	1	0	0
##	[17,]	0	1	0	0	1	0	1	0	0
##	[18,]	0	0	1	0	1	0	1	0	0
##	[19,]	1	0	0	0	0	1	1	0	0
##	[20,]	0	1	0	0	0	1	1	0	0
##	[21,]	0	0	1	0	0	1	1	0	0
##	[22,]	1	0	0	1	0	0	0	1	0
##	[23,]	0	1	0	1	0	0	0	1	0
##	[24,]	0	0	1	1	0	0	0	1	0
##	[25,]	1	0	0	0	1	0	0	1	0
##	[26,]	0	1	0	0	1	0	0	1	0
##	[27,]	0	0	1	0	1	0	0	1	0
##	[28,]	1	0	0	0	0	1	0	1	0
##	[29,]	0	1	0	0	0	1	0	1	0
##	[30,]	0	0	1	0	0	1	0	1	0
##	[31,]	1	0	0	1	0	0	0	0	1
##	[32,]	0	1	0	1	0	0	0	0	1
##	[33,]	0	0	1	1	0	0	0	0	1
##	[34,]	1	0	0	0	1	0	0	0	1
##	[35,]	0	1	0	0	1	0	0	0	1
##	[36,]	0	0	1	0	1	0	0	0	1
##	[37,]	1	0	0	0	0	1	0	0	1
##	[38,]	0	1	0	0	0	1	0	0	1
##	[39,]	0	0	1	0	0	1	0	0	1

```
# Row-Column-Design: (cf. John et al. 2004, Table II and page 2649f.)
```

```
rcDesign <- Crossover:::rcd(design, v=v, model=8)
```

```
rcDesign
```

```
##      [,1] [,2] [,3]
```

```
## [1,]    3    2    1
```

```
## [2,]   11    7    6
```

```
## [3,]   34   23   21
```

```
## [4,]   24   17   37
```

```
# Design Matrix of Row-Column Design:
```

```
Xr <- Crossover:::rcdMatrix(rcDesign, v, model=8)
```

```
Xr
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14] [,15] [,16] [,17]
```

```
## [1,]    0    0    1    0    0    0    0    0    0    0    0    0    0    0    0    0    0
```

```
## [2,]    0    1    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
```

```
## [3,]    1    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
```

```
## [4,]    0    0    0    0    0    0    0    0    0    0    1    0    0    0    0    0    0
```

```
## [5,]    0    0    0    0    0    0    1    0    0    0    0    0    0    0    0    0    0
```

```
## [6,]    0    0    0    0    0    1    0    0    0    0    0    0    0    0    0    0    0
```

```
## [7,]    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
```

```
## [8,]    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
```

```
## [9,]    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
```

```
## [10,]   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
```

```
## [11,]   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    1
```

```
## [12,]   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
```

```
##      [,22] [,23] [,24] [,25] [,26] [,27] [,28] [,29] [,30] [,31] [,32] [,33] [,34] [,35] [,36] [,37]
```

```
## [1,]    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
```

```
## [2,]    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
```

```
## [3,]    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
```

```
## [4,]    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
```

```
## [5,]    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
```

```
## [6,]    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
```

```
## [7,]    0    0    0    0    0    0    0    0    0    0    0    0    0    1    0    0
```

```
## [8,]    0    1    0    0    0    0    0    0    0    0    0    0    0    0    0    0
```

```
## [9,]    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
```

```
## [10,]   0    0    1    0    0    0    0    0    0    0    0    0    0    0    0    0
```

```
## [11,]   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
```

```
## [12,]   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
```

```
# Design Matrix of Cross-Over Design:
```

```
X <- Xr %*% H
```

```
X
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]
```

```
## [1,]    0    0    1    0    0    0    0    0    0
```

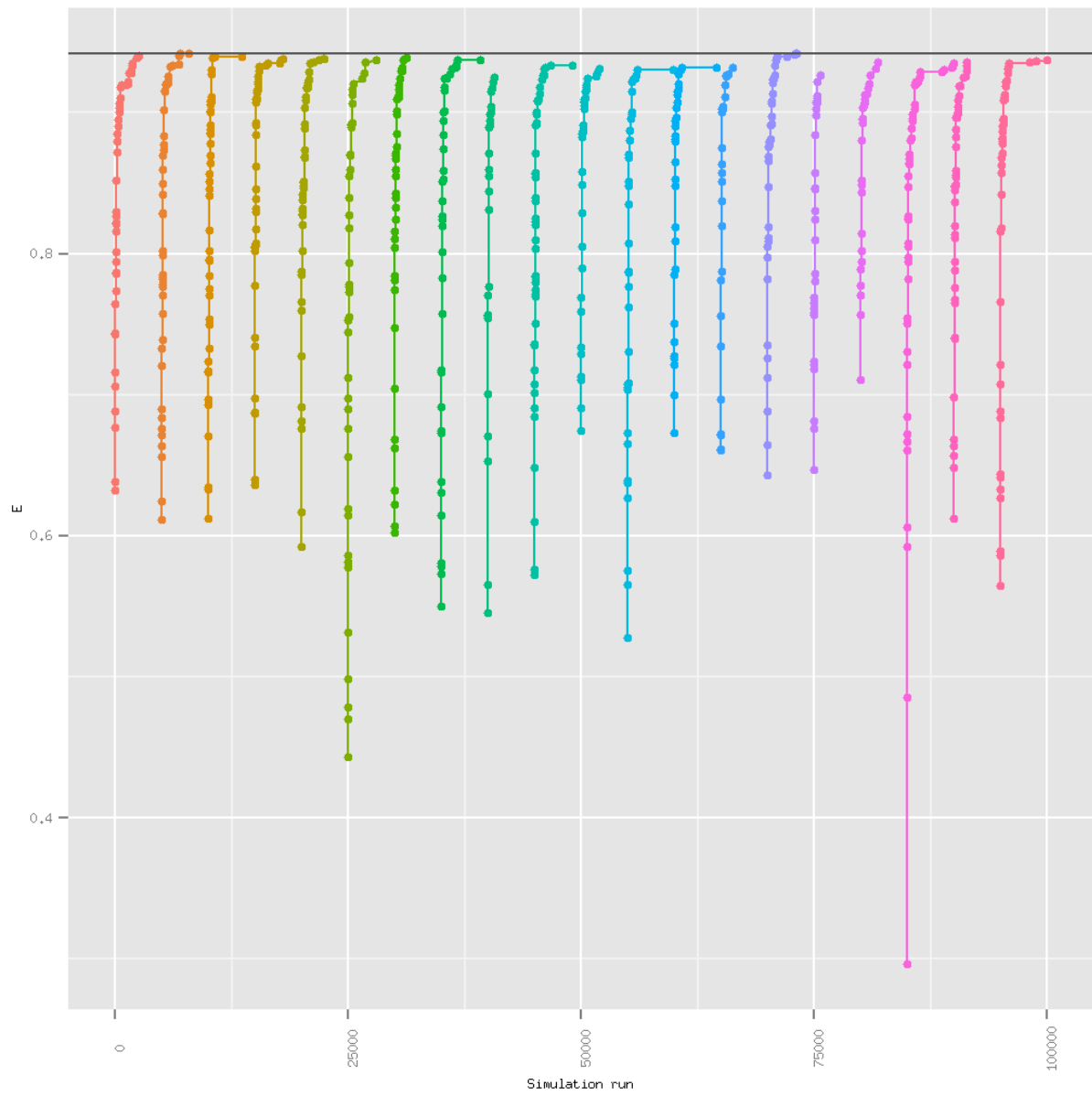
```
## [2,]    0    1    0    0    0    0    0    0    0
```

##	[3,]	1	0	0	0	0	0	0	0
##	[4,]	0	1	0	0	0	1	0	0
##	[5,]	1	0	0	0	1	0	0	0
##	[6,]	0	0	1	1	0	0	0	0
##	[7,]	1	0	0	0	1	0	0	1
##	[8,]	0	1	0	1	0	0	1	0
##	[9,]	0	0	1	0	0	1	1	0
##	[10,]	0	0	1	1	0	0	1	0
##	[11,]	0	1	0	0	1	0	1	0
##	[12,]	1	0	0	0	0	1	0	1

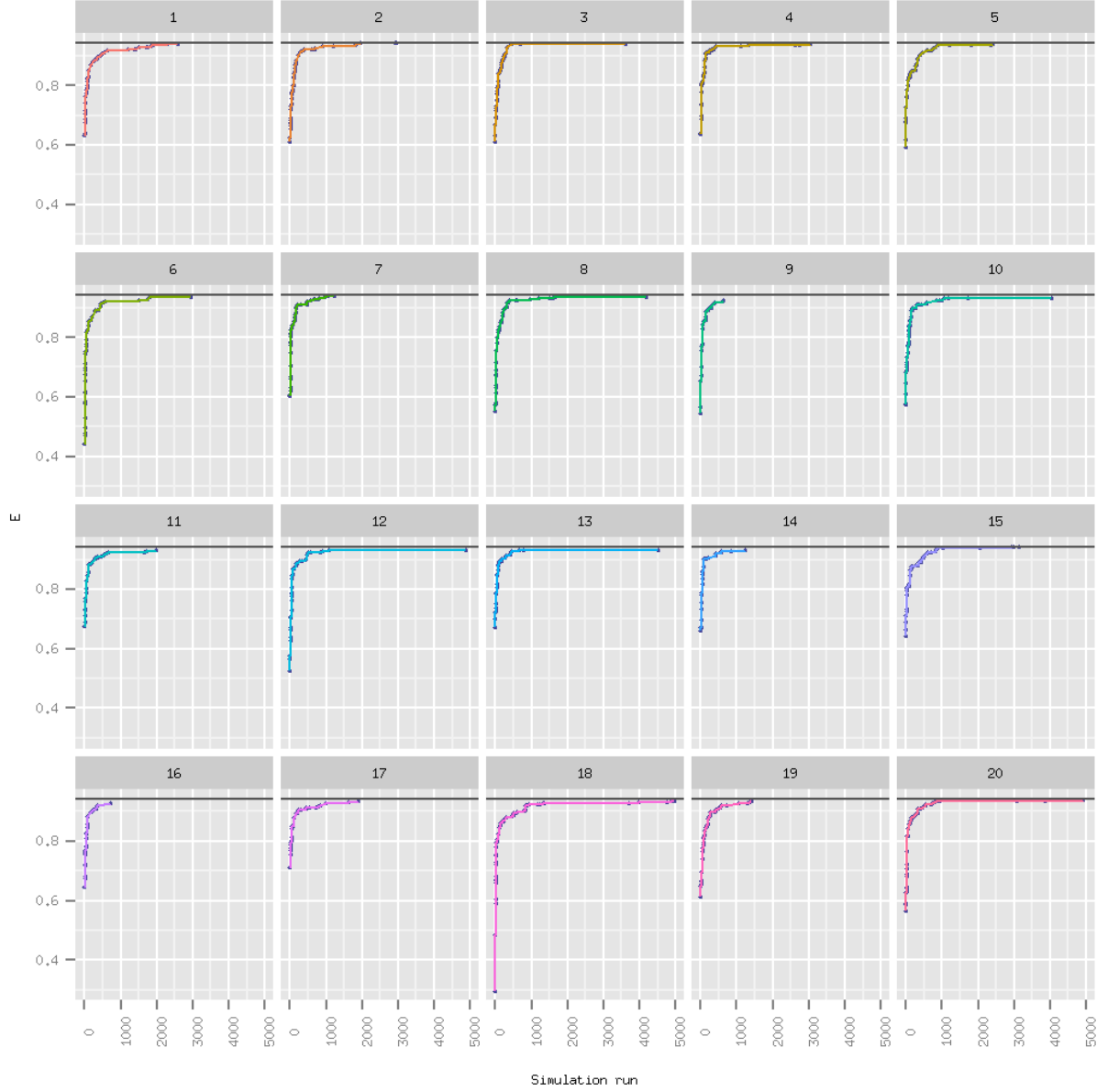
### 3 Search strategy

Monte-Carlo search for first candidates, followed by parallel hill climbing search, extended by long jumps. In the following graphic we see 100.000 design evaluations (that take less than 7 seconds on my computer), consisting out of 20 hill climbing search *runs* from 20 different designs as starting point which were selected by a Monte-Carlo search. We can see that 10 out of the 20 search runs were not able to achieve the efficiency the other 10 achieved. We can see, that there were most likely at least four local maxima the hill climbing search found and got stuck:

```
set.seed(42)
x <- searchCrossOverDesign(s=9, p=5, v=4, model=4)
plot(x)
```



```
plot(x, type=2)
```



There are different approaches to calculate the Information matrix  $A_r$  of the row-column design. In the following code fragment we compare the two different approaches:

$$A_r = r^\delta - \frac{1}{s}N_p N_p' - \frac{1}{p}N_s N_s' + \frac{1}{ps}rr' \text{ and}$$

$$A_r = X_r'(I - P_z)X_r.$$

For details (what are  $N_p$ ,  $N_s$ ,  $P_z$ , etc.) see John and Williams [1995] and John et al. [2004].

```
attach(loadNamespace("Crossover"), name="namespace:Crossover", pos=3, warn.conflicts=FALSE)

s <- 6
p <- 3
v <- 3
model <- 1
data(williams)
design <- williams3t
```



```

rcDesign <- rcd(design, v, model)
# JRW, p 2650, first equation on that page, without number
Ar <- infMatrix(rcDesign, v, model)
Xr <- rcdMatrix(rcDesign, v, model)
# JRW, p 2650, second equation on that page, number 11
Ar2 <- t(Xr) %*% (diag(s*p)-Crossover::getPZ(s,p)) %*% Xr
max(abs(Ar-Ar2))

## [1] 6.661e-16

fXr <- cbind(Xr, getZ(s,p))
Ar3 <- t(fXr) %*% fXr
ginv(Ar3)[1:12,1:12]-ginv(Ar2)

##           [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]      [,8]
## [1,]  8.781e-03  8.781e-03  8.781e-03  1.988e-16 -1.636e-03 -1.636e-03 -1.636e-03  7.277e-19 -1.63
## [2,]  8.781e-03  8.781e-03  8.781e-03  1.247e-16 -1.636e-03 -1.636e-03 -1.636e-03 -2.742e-16 -1.63
## [3,]  8.781e-03  8.781e-03  8.781e-03 -3.317e-16 -1.636e-03 -1.636e-03 -1.636e-03  2.735e-16 -1.63
## [4,]  2.204e-17 -2.691e-16  2.569e-16  1.659e-31  3.291e-16 -3.190e-16 -1.639e-16  7.409e-32  2.5
## [5,] -1.636e-03 -1.636e-03 -1.636e-03  4.232e-16  3.573e-03  3.573e-03  3.573e-03 -5.657e-17  3.57
## [6,] -1.636e-03 -1.636e-03 -1.636e-03 -4.185e-16  3.573e-03  3.573e-03  3.573e-03  2.413e-17  3.57
## [7,] -1.636e-03 -1.636e-03 -1.636e-03 -4.177e-16  3.573e-03  3.573e-03  3.573e-03  2.360e-16  3.57
## [8,]  1.396e-17 -1.009e-16  8.692e-17 -1.578e-32  1.206e-16 -8.094e-17  2.935e-17  8.808e-32 -1.1
## [9,] -1.636e-03 -1.636e-03 -1.636e-03  7.737e-16  3.573e-03  3.573e-03  3.573e-03 -2.168e-16  3.57
## [10,] -1.636e-03 -1.636e-03 -1.636e-03  2.207e-16  3.573e-03  3.573e-03  3.573e-03 -2.368e-16  3.57
## [11,] -1.636e-03 -1.636e-03 -1.636e-03 -5.684e-16  3.573e-03  3.573e-03  3.573e-03  2.501e-16  3.57
## [12,] -2.235e-19 -7.336e-17  7.359e-17  3.827e-31  3.676e-16 -3.392e-16 -3.443e-16 -1.869e-31  2.6
##           [,12]
## [1,]  8.043e-20
## [2,]  5.077e-17
## [3,] -5.085e-17
## [4,] -1.096e-31
## [5,] -1.185e-16
## [6,]  1.206e-16
## [7,]  6.287e-17
## [8,] -3.226e-32
## [9,] -1.585e-18
## [10,] -6.325e-17
## [11,] -7.020e-20
## [12,] -1.026e-31

H <- linkMatrix(model=model,v=v)
fX <- cbind(Xr%*%H, getZ(s,p))
A1 <- t(fX) %*% fX
A2 <- t(H)%*%Ar%*%H

# While A1 and A2 differ:

```

```

ginv(A1)[1:6,1:6]

##           [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
## [1,]  0.14611 -0.06222 -0.06222  0.08535 -0.03965 -0.03965
## [2,] -0.06222  0.14611 -0.06222 -0.03965  0.08535 -0.03965
## [3,] -0.06222 -0.06222  0.14611 -0.03965 -0.03965  0.08535
## [4,]  0.08535 -0.03965 -0.03965  0.25831 -0.11669 -0.11669
## [5,] -0.03965  0.08535 -0.03965 -0.11669  0.25831 -0.11669
## [6,] -0.03965 -0.03965  0.08535 -0.11669 -0.11669  0.25831

ginv(A2)

##           [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
## [1,]  0.13889 -0.06944 -0.06944  0.08333 -0.04167 -0.04167
## [2,] -0.06944  0.13889 -0.06944 -0.04167  0.08333 -0.04167
## [3,] -0.06944 -0.06944  0.13889 -0.04167 -0.04167  0.08333
## [4,]  0.08333 -0.04167 -0.04167  0.25000 -0.12500 -0.12500
## [5,] -0.04167  0.08333 -0.04167 -0.12500  0.25000 -0.12500
## [6,] -0.04167 -0.04167  0.08333 -0.12500 -0.12500  0.25000

max(abs(ginv(A1)[1:6,1:6]-ginv(A2)))

## [1] 0.008314

# The variances for the estimable contrasts are the same:

C <- matrix(0,nrow=15,ncol=1)
C[1:2,1] <- c(-1,1)
tdiff1 <- t(C)%*%ginv(A1)%*%C
tdiff2 <- t(C[1:6,])%*%ginv(A2)%*%C[1:6,]
tdiff1 - tdiff2

##           [,1]
## [1,] 1.665e-16

C <- matrix(0,nrow=6,ncol=1)
C[1:2,1] <- c(-1,1)
tdiff1 <- t(C)%*%ginv(A1)[1:6,1:6]%*%C
tdiff2 <- t(C)%*%ginv(A2)%*%C
tdiff1 - tdiff2

##           [,1]
## [1,] 1.665e-16

```

**Definition 3.1** (Efficiency). If the difference of two treatments, each replicated  $r$  times, is estimated in an *ideal design* with residual variance  $\sigma^2$ , the variance of the estimated difference  $\bar{y}_A - \bar{y}_B$  is

$$V_I = \frac{2\sigma^2}{r}.$$

In a cross-over design the variance of a parameter estimates for  $A$  and  $B$  are given by the corresponding elements  $\sigma^2 x_A$  and  $\sigma^2 x_B$  of

$$\sigma^2(X^T X)^{-1}$$

with  $X$  the crossover design matrix. The variance of the difference is  $\sigma^2(x_A + x_B - 2 * x_{AB})$  where  $x_{AB}$  is the element of  $X^T X$  specifying the covariance of the two treatment parameter estimates. The efficiency is

$$E := \frac{V_I}{V_C} = \frac{2}{rx} \in [0, 1].$$

## 4 Random Subject Effects Model

See Jones and Kenward [2003], 5.3, page 213ff. The model stays the same

$$Y_{ijk} = \mu + \pi_j + \tau_{d[i,j]} + \lambda_{d[i,j-1]} + s_{ik} + e_{ijk},$$

but we also assume that the subject effects follow a normal distribution:  $s_{ik} \sim \mathcal{N}(0, \sigma_s^2)$ .

In matrix notation we have

$$Y = X\beta + Z\gamma + \varepsilon$$

with  $X$  and  $Z$  the fixed and random effects design matrices<sup>5</sup>,  $\varepsilon \sim \mathcal{N}(0, \Sigma)$  and  $\gamma \sim \mathcal{N}(0, D)$ . Then<sup>6</sup>

$$\text{Var}(Y) = ZDZ^T + \Sigma.$$

For known  $V := \text{Var}(Y) = ZDZ^T + \Sigma$  the MLE and BLUE is given by

$$\hat{\beta} = (X^t V^{-1} X)^{-1} X^t V^{-1} Y.$$

---

<sup>5</sup>Note that  $X$  and  $Z$  are different from the ...

<sup>6</sup>See for example [Lee et al., 2006, chapter 5].

## Index

block design matrix, 4

link matrix, 4

model

- no carry-over into self, 9

- placebo model, 8

- proportionality model, 11

- self-adjacency model, 7

- standard additive, 4

- treatment decay, 10

- with full set of interactions, 6

- with second-order carry-over effects, 11

second-order carry-over effects, 11

self-adjacency model, 7

treatment design matrix, 4

## References

- Ian Anderson and DA Preece. Locally balanced change-over designs. *Utilitas Mathematica*, 62:33–59, 2002.
- DS Archdeacon, Jeffrey H. Dinitz, Douglas R. Stinson, and TW Tillson. Some new row-complete latin squares. *Journal of Combinatorial Theory, Series A*, 29(3):395–398, 1980.
- Glenn Francis Atkinson. Designs for sequences of treatments with carry-over effects. *Biometrics*, pages 292–309, 1966.
- LN Balaam. A two-period design with t2 experimental units. *Biometrics*, pages 61–73, 1968.
- S. Bate and B. Jones. The construction of universally optimal uniform cross-over designs. Technical report, GlaxoSmithKline Biomedical Data Sciences, 2002.
- II Berenblut. Change-over designs with complete balance for first residual effects. *Biometrics*, 20(4):707–712, 1964.
- EA Blaisdell Jr and D Raghavarao. Partially balanced change-over designs based on m-associate class pbib designs. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 334–338, 1980.
- Mausumi Bose and Aloke Dey. *Optimal Crossover Designs*. World Scientific Publishing Company, 2nd edition edition, 2009.
- WG Cochran, KM Autrey, and CY Cannon. A double change-over design for dairy cattle feeding experiments. *Journal of Dairy Science*, 24(11):937–951, 1941.
- AW Davis and WB Hall. Cyclic change-over designs. *Biometrika*, 56(2):283–293, 1969.
- Dirk Eddebuettel and Romain François. Rcpp: Seamless R and C++ integration. *Journal of Statistical Software*, 40(8):1–18, 2011.
- Dirk Eddebuettel and Conrad Sanderson. RcppArmadillo: Accelerating R with high-performance C++ linear algebra. *Computational Statistics and Data Analysis*, in press, 2013. URL <http://dx.doi.org/10.1016/j.csda.2013.02.005>.
- WT Federer and GF Atkinson. Tied-double-change-over designs. *Biometrics*, pages 168–181, 1964.
- David J Fletcher. A new class of change-over designs for factorial experiments. *Biometrika*, 74(3):649–654, 1987.
- I Iqbal and B Jones. Efficient repeated measurements designs with equal and unequal period sizes. *Journal of statistical planning and inference*, 42(1):79–88, 1994.
- J Aneurin John and Maurice Henry Quenouille. *Experiments: design and analysis*. Griffin London, 1977.
- J.A. John and E.R. Williams. *Cyclic and Computer Generated Designs*. Monographs on Statistics and Applied Probability. Chapman & Hall/CRC, 1995. ISBN 9780412575808.
- JA John, KG Russell, and D Whitaker. Crossover: an algorithm for the construction of efficient cross-over designs. *Statistics in medicine*, 23(17):2645–2658, 2004.
- Byron Jones and Michael G Kenward. *Design and analysis of cross-over trials*, volume 98. Chapman & Hall, 2nd edition edition, 2003.
- Eugene M Laska and Morris Meisner. A variational approach to optimal two-treatment crossover designs: application to carryover-effect models. *Journal of the American Statistical Association*, 80(391):704–710, 1985.

- Youngjo Lee, John A Nelder, and Yudi Pawitan. *Generalized linear models with random effects: unified analysis via H-likelihood*. CRC Press, 2006.
- SM Lewis, DJ Fletcher, and JNS Matthews. Factorial cross-over designs in clinical trials. *Optimal Design and Analysis of Experiments*, pages 133–140, 1988.
- HL Lucas. Switchback trials for more than two treatments. *Journal of Dairy Science*, 39(2):146–154, 1956.
- JNS Matthews. Optimal crossover designs for the comparison of two treatments in the presence of carryover effects and autocorrelated errors. *Biometrika*, 74(2):311–320, 1987.
- HD Patterson and Henry Lawrence Lucas. *Change-over designs*. North Carolina Agricultural Experiment Station and US Department of Agriculture, 1962.
- Joseph G Pigeon. *Residual effects designs for comparing treatments with a control*. PhD thesis, Temple University, 1985.
- P. Prescott. Construction of sequentially counterbalanced designs formed from two or more latin squares. In R. Dutter and W. Grossmann, editors, *Proceedings of the 11th Symposium on Computational Statistics held in Vienna, Austria*, pages 435–440. Physica-Verlag: Heidelberg, 1994.
- P Prescott. Construction of sequentially counterbalanced designs formed from two latin squares. *Utilitas Mathematica*, 55:135–152, 1999.
- H. Quenouille. *The Design and Analysis of Experiment*. Hafner Pub. Co., 1953.
- KG Russell. The construction of good change-over designs when there are fewer units than treatments. *Biometrika*, 78(2):305–313, 1991.
- Martin Oliver Sailer. *crossdes: Construction of Crossover Designs*, 2013. URL <http://CRAN.R-project.org/package=crossdes>. R package version 1.1-1.
- Simon Urbanek. *rJava: Low-level R to Java interface*, 2013. URL <http://CRAN.R-project.org/package=rJava>. R package version 0.9-4.
- EJ Williams. Experimental designs balanced for the estimation of residual effects of treatments. *Australian Journal of Chemistry*, 2(2):149–168, 1949.

# Table of Symbols

## Sets

$\mathbb{R}$	set of real numbers
$\mathbb{N}_0$	set of natural numbers (including 0)
$\mathfrak{Pot}(X)$	power set of set $X$ , i.e. the set of all subsets of $X$

## Variables

$v$	number of treatments
$p$	number of periods
$s$	number of sequences
$\mu$	intercept
$\pi_j$	period effect for period $j$
$\tau_{d[i,j]}$	direct treatment effect for treatment $d[i,j]$ in period $j$ of sequence $i$
$\lambda_{d[i,j-1]}$	first-order carry-over effect (0 for $j-1=0$ )
$s_{ik}$	$k$ th subject effect on sequence $i$
$e_{ijk}$	random error with zero mean and variance $\sigma^2$

## Functions

$X'$	transpose of matrix $X$
$X^+$	Moore-Penrose pseudoinverse of $X$
$\langle \cdot, \cdot \rangle$	standard direct product $\langle x, y \rangle = \sum_{j=1}^n x_j \cdot y_j$ for $x, y \in \mathbb{R}^n$
$\text{id}_X$	identity on $X$ , i.e. $\text{id}_X : X \rightarrow X, x \mapsto x$

## Other Symbols

$\mathcal{N}(\mu, \sigma^2)$	Normal distribution with mean $\mu$ and variance $\sigma^2$ .
$\mathcal{N}(\mu, \Sigma)$	Multivariate normal distribution with mean $\mu$ and covariance matrix $\Sigma$ .