

1 Optimizing RCBD with independent treatments

1.1 Specify required input

```
rm(list = ls(all=TRUE)[!grepl("global.var.A", ls(all=TRUE))])
h2=0.6; rhox=0.6; rhoy=0.6; DesN=100; criteria="A"; s20=0
pedigree = "Indep"
blocks=6;rb=5;cb=6;Tr=15;Tc=12;Treatments=1:30
source("final.R")
```

1.2 Generate m=100 initial designs

```
res1 = MultipleDesigns(DesN,blocks,Treatments,
  rb,cb,Tr,Tc,Amat=FALSE,criteria,h2,rhox,
  rhoy,s20,irregular=FALSE)
```

```
names(res1)
```

```
## [1] "newmatdf" "trace0" "min_value" "meanA"
```

1.3 Calculate the variance-covariance matrix

```
res2 <- VarCov.rcbd(matdf=res1$newmatdf,rhox,
  rhoy,h2,s20,Tr,Tc,criteria="A",Amat=FALSE,irregular=FALSE)
Rinv = as.matrix(res2$Rinv)
Ginv = as.matrix(res2$Ginv)
K = as.matrix(res2$K)
names(res2)
```

```
## [1] "traceI" "Ginv" "Rinv" "K"
```

1.4 Apply the simple pairwise algorithm (SP)

```
start <- Sys.time()
ans = Optimize.rcbd(matdf=res1$newmatdf,n=5000,
  traceI=res2$traceI,criteria="A",Rinv,Ginv,K)
tim <- Sys.time() - start
```

```
tim
```

```
## Time difference of 1.05481 mins
```

```
names(ans)
```

```
## [1] "TRACE" "mat" "Design_best"
```

1.5 SP: Calculate design efficiency based on A-optimality

```
Aopt <- ans$mat[nrow(ans$mat),'value']
init0 <- ans$mat[1,'value']
(ODE = abs((init0 - Aopt)/init0)*100)
```

```
##      value
## 3.741123
```

```
head(ans$Design_best)
```

```
##   Row Col Reps Treatments
## 1   1   1    1          16
## 2   1   2    1          21
## 3   1   3    1           7
## 4   1   4    1          28
## 5   1   5    1          17
## 6   1   6    1           1
```

1.6 Plot the initial and final designs

```
plotdesign(res1$newmatdf,rb,cb,Tr,Tc,text.size=5,
          axistextsize=12,
          titlesize=15,axislabsize=12,
          maintitle="Initial Design")
plotdesign(ans$Design_best,rb,cb,Tr,Tc,text.size=5,
          axistextsize=12,
          titlesize=15,axislabsize=12,
          maintitle="Final Design")
```

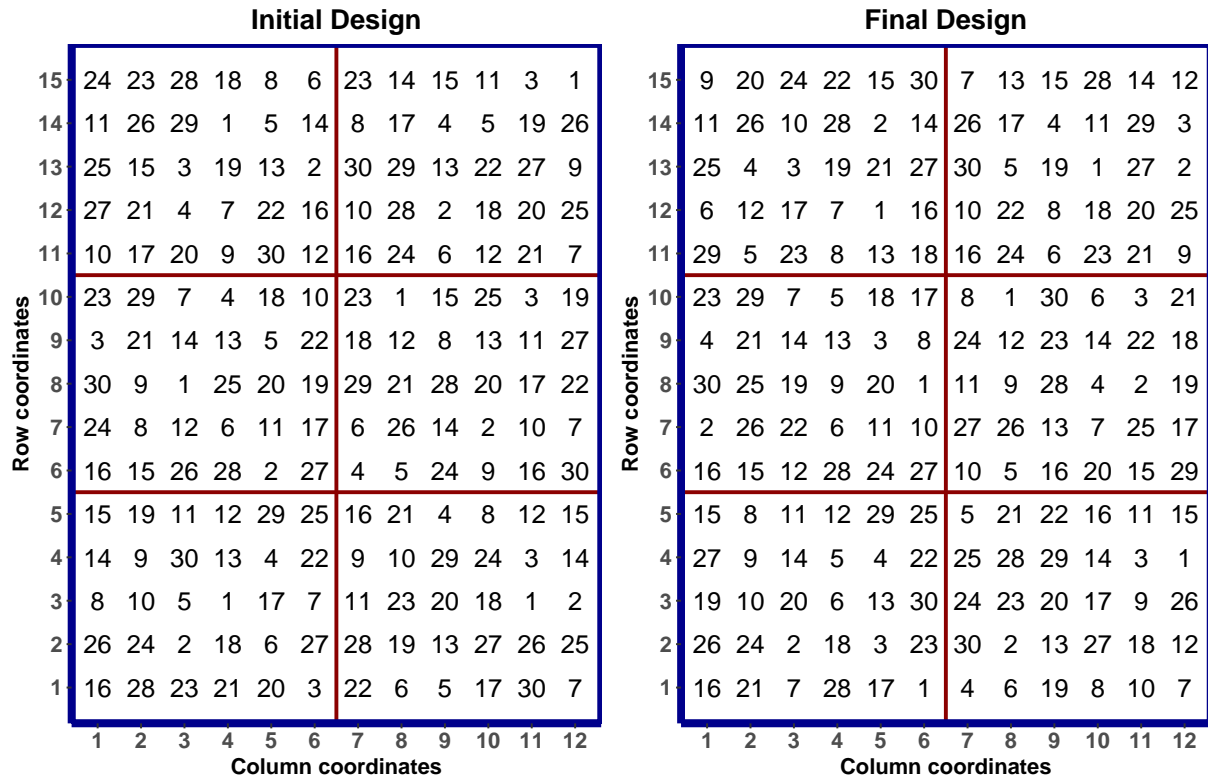


Figure 1: Initial and final designs using SP algorithm for genetically unrelated treatments.

```
library(desplot)
desplot(Reps~Col+Row, ans$Design_best, text=Treatments,
        cex=1, show.key=FALSE, main=NULL)
```

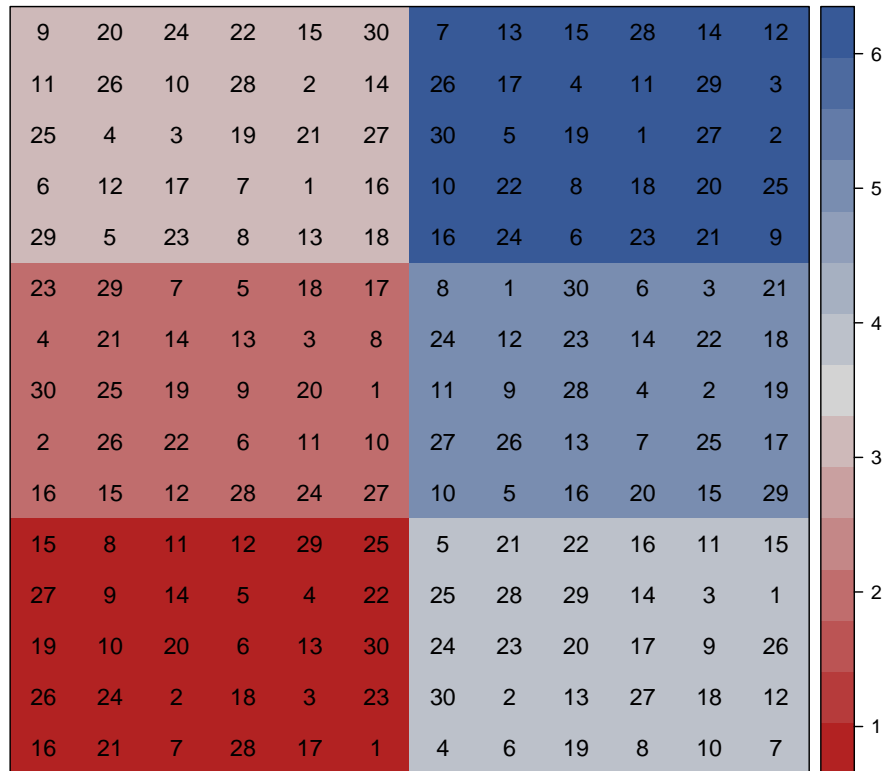


Figure 2: Final Design (using desplot package).

1.7 Apply the simulated annealing algorithm (SA)

```
rm(ans)
rm(tim)
start <- Sys.time()
ans = Optimize_SimAnn_rcbd(matdf=res1$newmatdf,n=5000,
  traceI=res2$traceI,criteria="A",Rinv,Ginv,K)
tim <- Sys.time() - start
```

```
tim
```

```
## Time difference of 1.056938 mins
```

1.8 SA: Calculate design efficiency based on A-optimality

```
Aopt <- ans$mat[nrow(ans$mat),'value']
init0 <- ans$mat[1,'value']
(ODE = abs((init0 - Aopt)/init0)*100)
```

```
## value
## 3.533626
```

1.9 Apply the greedy pairwise swap algorithm (GP4)

```
rm(ans)
rm(tim)
start <- Sys.time()
ans = OptimizeGreedy.rcbd(matdf=res1$newmatdf,
  n=5000,traceI=res2$traceI,
  criteria="A",gsize=4,Rinv,Ginv,K)
tim <- Sys.time() - start
```

```
tim
```

```
## Time difference of 1.059972 mins
```

1.10 GP4: Calculate design efficiency based on A-optimality

```
Aopt <- ans$mat[nrow(ans$mat),'value']
init0 <- ans$mat[1,'value']
(ODE = abs((init0 - Aopt)/init0)*100)
```

```
## value
## 2.991614
```

2 Optimizing RCBD with treatments from Half-sibs family

2.1 Specify required input

```
rm(list = ls(all=TRUE)[!grepl("global.var.A", ls(all=TRUE))])
h2=0.6; rhox=0.6; rhoy=0.6;DesN=100; criteria="A"; s20=0
pedigree = "Half-sib"
blocks=6;rb=5;cb=6;Tr=15;Tc=12;Treatments=1:30
source("final.R")
```

2.2 Read pedigree file and calculate the numerator relationship matrix

```
ped30hs <- read.csv("ped30hs.csv",header=TRUE,as.is = TRUE)
Amat <- GenA(male=ped30hs[, "sire"], female = ped30hs[, "dam"])
Amat <- as.matrix(Amat[-c(1:5), -c(1:5)])
```

2.3 Generate m=100 initial designs

```
res1 = MultipleDesigns(DesN,blocks,  
  Treatments,rb,cb,Tr,Tc,Amat,criteria,  
  h2,rhox,rhoy,s20,irregular=FALSE)
```

```
names(res1)
```

```
## [1] "newmatdf" "trace0" "min_value" "meanA"
```

2.4 Calculate the variance-covariance matrix

```
res2 <- VarCov.rcbd(matdf=res1$newmatdf,rhox,  
  rhoy,h2,s20,Tr,Tc,criteria="A",Amat,irregular=FALSE)  
Rinv = as.matrix(res2$Rinv)  
Ginv = as.matrix(res2$Ginv)  
K = as.matrix(res2$K)  
res2$traceI
```

```
## [1] 1.848873
```

```
names(res2)
```

```
## [1] "traceI" "Ginv" "Rinv" "K"
```

2.5 Apply the simple pairwise algorithm (SP)

```
start <- Sys.time()  
ans = Optimize.rcbd(matdf=res1$newmatdf,  
  n=5000,traceI=res2$traceI,criteria="A",Rinv,Ginv,K)  
tim <- Sys.time() - start
```

```
tim
```

```
## Time difference of 1.05918 mins
```

```
names(ans)
```

```
## [1] "TRACE" "mat" "Design_best"
```

2.6 SP: Calculate design efficiency based on A-optimality

```
Aopt <- ans$mat[nrow(ans$mat),'value']  
init0 <- ans$mat[1,'value']  
(ODE = abs((init0 - Aopt)/init0)*100)
```

```
##      value
## 2.690385
```

```
head(ans$Design_best)
```

```
##   Row Col Reps Treatments
## 1   1   1    1          1
## 2   1   2    1         29
## 3   1   3    1          8
## 4   1   4    1         27
## 5   1   5    1         22
## 6   1   6    1         21
```

2.7 Plot the initial and final designs

```
plotdesign(res1$newmatdf,rb,cb,Tr,Tc,text.size=5,
          axistextsize=12,
          titlesize=15,axislabsize=12,
          maintitle="Initial Design")
plotdesign(ans$Design_best,rb,cb,Tr,Tc,text.size=5,
          axistextsize=12,
          titlesize=15,axislabsize=12,
          maintitle="Final Design")
```

Initial Design														Final Design													
Row coordinates	15	2	30	9	21	19	3	12	29	11	13	18	24	15	5	14	9	30	23	21	25	11	12	26	15	5	
	14	7	15	22	13	23	24	21	27	23	7	6	5	14	8	15	22	12	20	24	21	27	23	7	6	17	
	13	25	29	11	1	8	4	8	16	2	4	3	26	13	13	16	11	1	27	10	2	13	8	4	14	19	
	12	18	20	17	16	14	5	14	17	10	30	1	9	12	17	26	3	25	7	2	29	10	28	16	1	20	
	11	12	10	28	27	6	26	25	20	15	28	22	19	11	18	4	6	19	28	29	24	9	18	30	22	3	
	10	10	3	4	2	13	11	18	8	24	2	26	21	10	10	3	19	2	22	28	9	8	24	18	16	21	
	9	22	24	7	5	28	27	9	20	3	11	7	15	9	11	24	27	5	7	30	28	5	3	11	7	15	
	8	20	15	26	16	14	9	23	25	1	19	5	10	8	20	15	17	1	6	26	23	19	1	4	12	10	
	7	1	12	8	21	6	18	16	29	22	27	17	28	7	9	12	8	21	14	18	26	27	2	20	29	25	
	6	17	19	25	30	29	23	6	30	14	13	4	12	6	4	16	25	13	29	23	6	30	14	17	22	13	
	5	16	27	10	29	13	4	28	4	20	23	14	26	5	16	3	14	23	13	4	10	6	13	14	20	22	
	4	23	6	8	25	30	24	25	5	15	9	21	7	4	24	30	20	25	26	10	30	5	15	9	21	26	
	3	26	2	28	15	11	14	1	18	19	16	6	12	3	6	2	18	15	11	5	1	18	19	16	17	24	
	2	5	17	3	9	20	12	2	27	24	29	3	17	2	28	17	7	9	19	12	27	25	4	29	11	23	
	1	22	18	1	19	7	21	10	30	13	22	8	11	1	1	29	8	27	22	21	12	28	2	7	8	3	
Column coordinates														Column coordinates													

Figure 3: Initial and final designs for half-sib treatments using simple pairwise algorithm.

```
library(desplot)
desplot(Reps~Col+Row, ans$Design_best, text=Treatments,
        cex=1, show.key=FALSE, main=NULL)
```

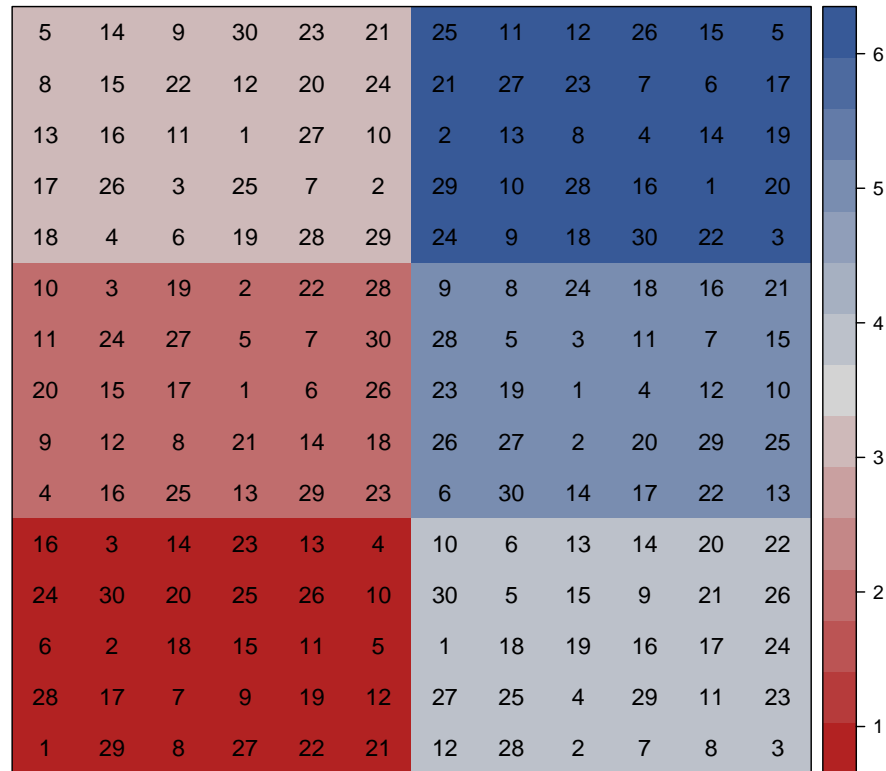


Figure 4: Final Design (using desplot package) for half-sibs family.

2.8 Apply the simulated annealing algorithm (SA)

```
rm(ans)
rm(tim)
start <- Sys.time()
ans = Optimize_SimAnn_rcbd(matdf=res1$newmatdf,
  n=5000,traceI=res2$traceI,criteria="A",Rinv,Ginv,K)
tim <- Sys.time() - start
```

```
tim
```

```
## Time difference of 1.060767 mins
```

```
names(ans)
```

```
## [1] "TRACE"      "mat"        "Design_best"
```

2.9 SA: Calculate design efficiency based on A-optimality

```
Aopt <- ans$mat[nrow(ans$mat),'value']
init0 <- ans$mat[1,'value']
(ODE = abs((init0 - Aopt)/init0)*100)
```

```
##      value
## 2.422064
```

2.10 Apply the genetic neighborhood algorithm (GN)

```
rm(ans)
rm(tim)
start <- Sys.time()
ans = Optimize_GNN_rcbd(matdf=res1$newmatdf,
  n=5000,traceI=res2$traceI,criteria="A",
  Amat,Rinv,Ginv,K)
tim <- Sys.time() - start
```

```
tim
```

```
## Time difference of 1.134075 mins
```

2.11 GN: Calculate design efficiency based on A-optimality

```
Aopt <- ans$mat[nrow(ans$mat),'value']
init0 <- ans$mat[1,'value']
(ODE = abs((init0 - Aopt)/init0)*100)
```

```
##      value
## 0.3834114
```

2.12 Apply the greedy pairwise swap algorithm (GP4)

```
rm(ans)
rm(tim)
start <- Sys.time()
ans = OptimizeGreedy_rcbd(matdf=res1$newmatdf,
  n=5000,traceI=res2$traceI,criteria="A",
  gsize=4,Rinv,Ginv,K)
tim <- Sys.time() - start
```

```
tim
```

```
## Time difference of 1.061638 mins
```

2.13 GP4: Calculate design efficiency based on A-optimality

```
Aopt <- ans$mat[nrow(ans$mat),'value']
init0 <- ans$mat[1,'value']
```

```
(ODE = abs((init0 - Aopt)/init0)*100)
```

```
##      value
## 2.170021
```

3 Optimizing RCBD with treatments from Full-sibs family

3.1 Specify required input

```
rm(list = ls(all=TRUE)[!grepl("global.var.A", ls(all=TRUE))])
h2=0.6; rhox=0.6; rhoy=0.6; DesN=100; criteria="A"; s20=0
pedigree = "Full-sib"
blocks=6;rb=5;cb=6;Tr=15;Tc=12;Treatments=1:30
source("final.R")
```

3.2 Read pedigree file and calculate the numerator relationship matrix

```
ped30fs <- read.csv("ped30fs.csv",header=TRUE,as.is = TRUE)
Amat <- GenA(male=ped30fs[, "sire"], female = ped30fs[, "dam"])
Amat <- as.matrix(Amat[-c(1:5), -c(1:5)])
```

3.3 Generate m=100 initial designs

```
res1 = MultipleDesigns(DesN,blocks,Treatments,
  rb, cb, Tr, Tc, Amat, criteria, h2,
  rhox, rhoy, s20, irregular=FALSE)
```

```
names(res1)
```

```
## [1] "newmatdf" "trace0" "min_value" "meanA"
```

3.4 Calculate the variance-covariance matrix

```
res2 <- VarCov.rcbd(matdf=res1$newmatdf,rhox,
rhoy,h2,s20,Tr,Tc,criteria="A",Amat,irregular=FALSE)
Rinv = as.matrix(res2$Rinv)
Ginv = as.matrix(res2$Ginv)
K = as.matrix(res2$K)
names(res2)
```

```
## [1] "traceI" "Ginv" "Rinv" "K"
```

```
res2$traceI
```

```
## [1] 4.38817
```

3.5 Apply the simple pairwise algorithm (SP)

```
start <- Sys.time()
ans = Optimize.rcbd(matdf=res1$newmatdf,n=5000,
  traceI=res2$traceI, criteria="A",Rinv,Ginv,K)
tim <- Sys.time() - start
```

```
tim
```

```
## Time difference of 1.064047 mins
```

```
names(ans)
```

```
## [1] "TRACE"      "mat"      "Design_best"
```

3.6 SP: Calculate design efficiency based on *A*-optimality

```
Aopt <- ans$mat[nrow(ans$mat),'value']
init0 <- ans$mat[1,'value']
(ODE = abs((init0 - Aopt)/init0)*100)
```

```
##      value
```

```
## 0.9719083
```

```
head(ans$Design_best)
```

```
##   Row Col Reps Treatments
## 1   1   1    1          5
## 2   1   2    1          9
## 3   1   3    1         24
## 4   1   4    1          3
## 5   1   5    1         15
## 6   1   6    1         30
```

3.7 Plot the initial and final designs

```
plotdesign(res1$newmatdf,rb,cb,Tr,Tc,text.size=5,
  axistextsize=12,
  titlesize=15,axislabsz=12,
  maintitle="Initial Design")
plotdesign(ans$Design_best,rb,cb,Tr,Tc,text.size=5,
  axistextsize=12,
```

```

titlesize=15,axislabsize=12,
maintitle="Final Design")

```

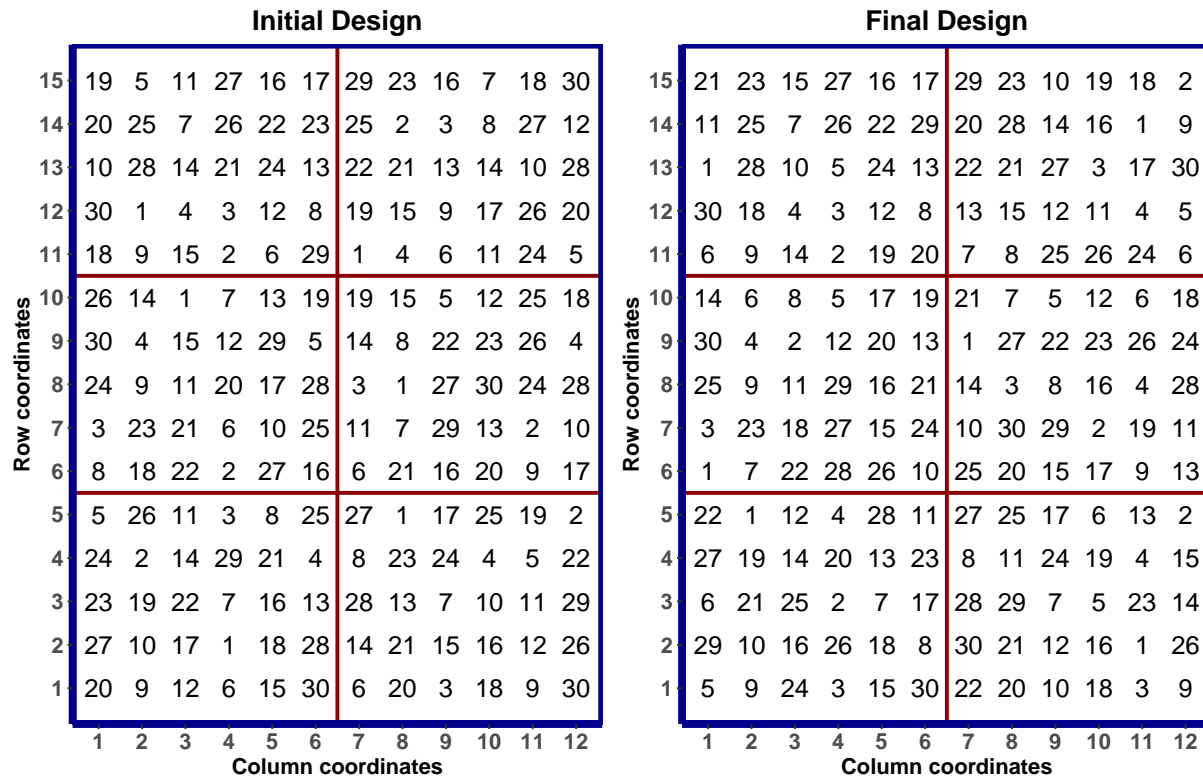


Figure 5: Initial and final designs for full-sib treatments using simple pairwise algorithm.

3.8 Apply the simulated annealing algorithm (SA)

```

rm(ans)
rm(tim)
start <- Sys.time()
ans = Optimize_SimAnn_rcbd(matdf=res1$newmatdf,
  n=5000,traceI=res2$traceI,
  criteria="A",Rinv,Ginv,K)
tim <- Sys.time() - start

```

```
tim
```

```
## Time difference of 1.055937 mins
```

3.9 SA: Calculate design efficiency based on A-optimality

```

Aopt <- ans$mat[nrow(ans$mat),'value']
init0 <- ans$mat[1,'value']

```

```
(ODE = abs((init0 - Aopt)/init0)*100)
```

```
##      value
## 1.061229
```

```
head(ans$Design_best)
```

```
##   Row Col Repts Treatments
## 1   1   1     1         2
## 2   1   2     1         5
## 3   1   3     1        19
## 4   1   4     1        29
## 5   1   5     1        25
## 6   1   6     1        24
```

3.10 Apply the genetic neighborhood algorithm (GN)

```
rm(ans)
rm(tim)
start <- Sys.time()
ans = Optimize_GNN_rcbd(matdf=res1$newmatdf,
  n=5000,traceI=res2$traceI,criteria="A",
  Amat,Rinv,Ginv,K)
tim <- Sys.time() - start
```

```
tim
```

```
## Time difference of 1.292614 mins
```

3.11 GN: Calculate design efficiency based on A-optimality

```
Aopt <- ans$mat[nrow(ans$mat),'value']
init0 <- ans$mat[1,'value']
(ODE = abs((init0 - Aopt)/init0)*100)
```

```
##      value
## 0.1465577
```

```
head(ans$Design_best)
```

```
##   Row Col Repts Treatments
## 1   1   1     1         20
## 2   1   2     1         9
## 3   1   3     1        12
## 4   1   4     1         6
## 5   1   5     1        15
```

```
## 6    1    6    1          30
```

3.12 Apply the greedy pairwise algorithm (GP4)

```
rm(ans)
rm(tim)
start <- Sys.time()
ans = OptimizeGreedy.rcbd(matdf=res1$newmatdf,
  n=5000,traceI=res2$traceI,criteria="A",
  gsize=4,Rinv,Ginv,K)
tim <- Sys.time() - start
```

```
tim
```

```
## Time difference of 1.061718 mins
```

3.13 GP4: Calculate design efficiency based on A-optimality

```
Aopt <- ans$mat[nrow(ans$mat),'value']
init0 <- ans$mat[1,'value']
(ODE = abs((init0 - Aopt)/init0)*100)
```

```
##      value
## 0.8103016
```

```
head(ans$Design_best)
```

```
##   Row Col Reps Treatments
## 1   1   1    1          4
## 2   1   2    1          29
## 3   1   3    1          25
## 4   1   4    1          22
## 5   1   5    1           3
## 6   1   6    1          10
```