Published October 6, 2016

Augmented Row–Column Designs for a Small Number of Checks

Hans-Peter Piepho* and Emlyn R. Williams

ABSTRACT

Augmented designs allow the testing of a large number of unreplicated entries. Replication of interspersed check cultivars according to a blocked design allows estimation of an error variance as well as local control for unreplicated entries. The most commonly used augmented designs use one-way blocking, but extensions have been suggested using blocking in rows and columns. We propose a flexible strategy to generate such designs, which is particularly suitable when the number of check cultivars is small. A salient feature of the approach is a grouping of adjacent rows and columns and the imposition of an additional blocking unit formed by the intersection of row and column groups. These superimposed blocking structures cater for an even distribution of replications for checks across the experimental field. We provide SAS code using the OPTEX procedure to generate such designs and discuss three examples illustrating the flexibility of the approach.

Core Ideas

- For augmented designs, the number of unreplicated entries and blocking unit dimensions needs flexibility.
- Blocking in rows and columns is desirable for local error control.
- Super-blocks can be defined based on groups of adjacent rows and columns.
- Units formed by the intersection of row and column groups can be a further blocking unit.
- These blocking units provide a convenient mechanism to distribute replications of check cultivars.

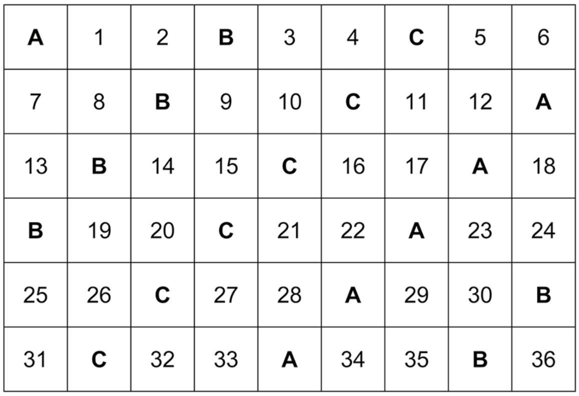
Published in Agron. J. 108:2256–2262 (2016)
doi:10.2134/agronj2016.06.0325
Received 3 June 2016
Accepted 23 Aug. 2016
Supplemental material available online

In early generations of a plant breeding program, a large number of entries need to be tested. Also, seed availability is often limited, meaning that only one plot can be planted per entry. To be able to estimate an error variance and allow for local adjustments by blocking, a common strategy is to allocate a certain fraction of the plots to check cultivars. For a good review of early developments in the design and analysis of unreplicated field trials, see Kempton (1984). Federer (1956, 1961) proposed augmented designs for this purpose. The general idea is to use a blocked design for the replicated check cultivars and then to augment the blocks of this design with unreplicated entries. Randomization of entries and checks within each augmented block provides the basis for an estimate of the error variance and adjustments for block effects.

The first proposals of augmented designs involved randomized complete block designs and resolvable incomplete block designs for the checks, which permit one-way elimination of heterogeneity. Later, the basic idea was extended to row–column designs where the numbers of rows and columns are equal (Federer et al., 1975). Lin and Poushinsky (1983) proposed row–column designs for check cultivars where check plots are arranged on a regular grid and unreplicated entry plots are systematically arranged around the replicated checks. Federer (2002) advocated the use of augmented lattice square designs, which were later extended to so-called α-α-designs by Williams and John (2003). These randomized designs are specifically suited for augmented designs with a larger number of check cultivars, whereas those by Federer et al. (1975) and Lin and Poushinsky (1983) work for a smaller number of check cultivars. All of these proposed designs are somewhat restricted regarding the number of check plots, the number of new cultivars, and the number of rows and columns that can be accommodated.

In this study, we were specifically concerned with augmented row–column designs for a smaller number of check cultivars that allow randomized allocation of genotypes (entries and checks) to plots and an estimation of row and column effects using a linear model with simple effects for genotypes, rows, and columns. Our aim was to improve on the flexibility compared with the designs mentioned above. For illustration, consider an example given in the review by Federer and Crossa (2005) for $k = 6$ rows, $s = 9$ columns, $v_c = 3$ check cultivars, each of which is replicated six times, and $v_e = 36$ unreplicated entries. The plan is given in Fig. 1. With respect to the check cultivars, each row is a complete block and each column is an incomplete block, thus potentially permitting the estimation

H.P. Piepho, Biostatistics Unit, Institute for Crop Science, Univ. of Hohenheim, 70593 Stuttgart, Germany; E.R. Williams, Statistical Consulting Unit, Australian National Univ., Canberra, ACT 0200, Australia. *Corresponding author (piepho@uni-hohenheim.de).



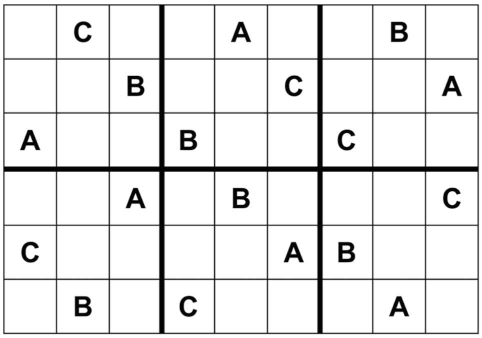
A	1	2	B	3	4	C	5	6
7	8	B	9	10	C	11	12	A
13	B	14	15	C	16	17	A	18
B	19	20	C	21	22	A	23	24
25	26	C	27	28	A	29	30	B
31	C	32	33	A	34	35	B	36

Fig. 1. An augmented row-column design, reproduced from Federer and Crossa (2005). Checks: A–C. Entries: 1–36. The design is unrandomized.

of row and column effects (although with the particular allocation of check plots in Fig. 1, they turn out to be non-estimable; see Example 1 below). Federer and Crossa (2005) required the checks (letters) to appear on diagonals of the unrandomized layout. Randomization of the design involves randomized allocation of unreplicated entries to plots and the permutation of rows and columns. The distribution of replications for the same check cultivar may turn out to be somewhat uneven due to this randomization procedure.

Inspection of Fig. 1 suggests that a third blocking structure can be imposed that regards a 3 × 3 array of plots as a block for the checks. In Fig. 1, each such block has three check plots, and so these blocks could be required to provide one plot for each check. An example of this alternate allocation is given in Fig. 2, which for simplicity shows only treatment labels for the replicated checks. In this design, each row is a complete block for checks and each column an incomplete block, as in Fig. 1. In addition, each 3 × 3 array of plots forms a complete block for checks. As a result of the third blocking structure imposed, the layout provides a more regular distribution of the replications for each check. Randomization of check plots for this layout can proceed by permuting rows within each group of three adjacent rows and by an analogous permutation of columns within groups of three adjacent columns. Furthermore, groups of rows as well as groups of columns can be permuted.

Our objective in this study was to provide a strategy for generating designs like the one in Fig. 2 such that all row and column effects, as well as effects for blocks given by a superimposed rectangular array of plots, can be estimated. Moreover, there should be flexibility regarding the total number of entries that can be tested and a provision for the randomized allocation of checks. To exemplify the versatility of our approach, we consider three examples. The first is the small design shown in Fig. 2. The second example (Fig. 3) concerns a larger design for 576 unreplicated entries and 648 plots with superimposed blocks of six rows and columns, each constituting a complete block with respect to four checks. The third example (Fig. 4) deals with a design involving blocks of unequal size, some of which are incomplete with respect to a set of four checks and some others of which have extra replication for one of the four checks.



	C			A			B	
		B			C			A
A			B			C		
		A		B				C
C					A	B		
	B		C				A	

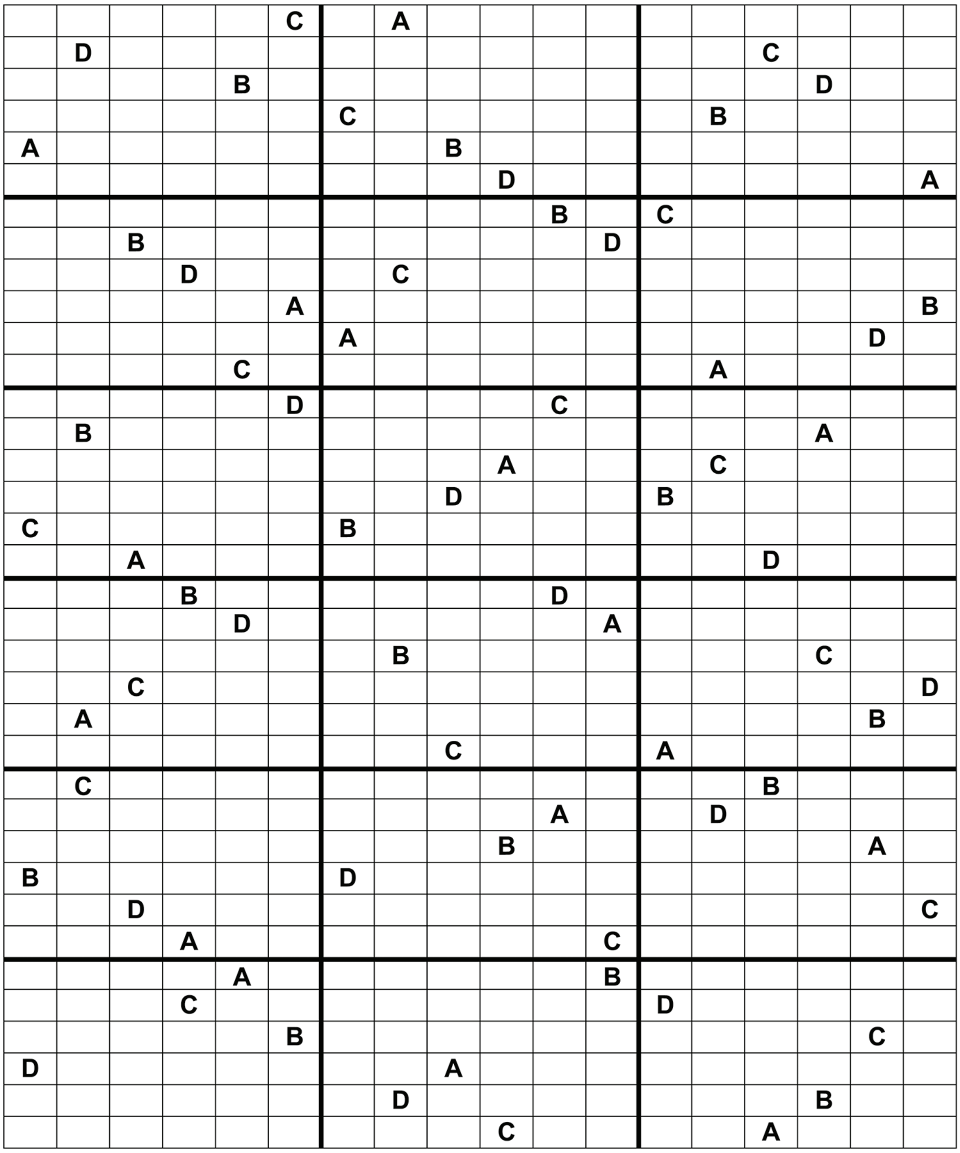
Fig. 2. An augmented row–column design with a third block factor defined by 3 × 3 arrays of plots (blocks delineated by solid lines). Checks: A–C. The 36 unreplicated entries are allocated to the empty cells (labels omitted for simplicity). Specification (v_c, g_k, g_s, k_b, s_b, p_b) = (3, 2, 3, 3, 3, 3).

A STRATEGY TO GENERATE AN AUGMENTED ROW–COLUMN DESIGN WITH SUPERIMPOSED RECTANGULAR BLOCKS

The proposed design strategy proceeds in six steps as follows:

1. For a row–column layout, define blocks as the units formed by the intersection of groups of adjacent rows (row groups, super-rows) and groups of adjacent columns (column groups, super-columns) (groups can be unequal size). When grouping rows and columns, note that the size of the groups determines the size of the blocks formed by the intersection of row and column groups. Define the number of plots for each block that are to be allocated to checks (this number need not be a constant).
2. Allocate check cultivars to check plots based on this model for design factors: rowgroup + colgroup + rowgroup·colgroup, where *rowgroup* and *colgroup* are factors identifying groups of adjacent rows and groups of adjacent columns defined in Step 1, and hence the crossed effect *rowgroup·colgroup* identifies the blocks formed by the intersection of row groups and column groups.
3. In each row group, consider the classification of rows by column groups (colgroup). From Step 2, we have a design for the check cultivar-by-column groups classification. Consider both check cultivar and column group of this design as block factors and optimize the allocation of rows so that effects for rows can be estimated with good efficiency. This produces a row–column group design for check cultivars.
4. In each column group, consider the classification of row groups (rowgroup) by columns. From Step 2, we have a design for the check cultivar-by-row group classification. Consider both check cultivar and row groups of this design as block factors and optimize the allocation of columns so that effects for columns can be estimated with good efficiency. This produces a row group–column design for check cultivars.
5. Merge the two designs obtained in Steps 3 and 4 to obtain the final design for checks.
6. Allocate entries to free plots in completely randomized order.

We implemented the optimizations required in Steps 2, 3, and 4 using the OPTEX procedure of SAS. This procedure performs a numerical search for an efficient design based on the D-optimality criterion (Atkinson et al., 2009). One may



					C		A										
	D													C			
				B											D		
						C							B				
A								B									
									D								A
										B		C					
		B									D						
			D				C										
					A												B
						A										D	
				C									A				
					D					C							
	B														A		
									A				C				
								D				B					
C						B											
		A												D			
			B							D							
				D							A						
							B								C		
		C															D
	A															B	
								C				A					
	C													B			
										A			D				
									B							A	
B						D											
		D															C
			A								C						
				A							B						
			C									D					
					B											C	
D								A									
							D								B		
									C					A			

Fig. 3. An augmented row–column design for the specification (v_c, g_k, g_s, k_b, s_b, p_b) = (4, 6, 3, 6, 6, 4) (blocks delineated by solid lines). Checks: A–D. The v_e = 576 unreplicated entries are allocated to the empty cells. There are 72 plots allocated to the check cultivars and 648 plots overall.

specify a model for the given blocking structure and provide the set of treatments to be used in the design. Optimization is then based on the determinant of the treatment information matrix. Details on the use of OPTEX may be found in Piepho (2015). Generation using the OPTEX procedure ensures that the design is randomized. The implicit randomization involves permutation of groups or rows (columns) and of rows (columns) within groups of rows (columns). The use of two macros written for this purpose is described in the Appendix, and the macros are made available in the supplementary material (file: augmented design.sas). The notation used to describe the designs is summarized in Table 1.

A few observations can be made on designs generated by this algorithm:

1. By crossing the g_k row groups with the g_s column groups, we obtain $g_k g_s$ blocks. If k and s are multiples of g_k and g_s, respectively, the blocks have $k_b = k/g_k$ rows and $s_b = s/g_s$ columns each. More flexibility is possible by allowing the number of rows per row group and the number of columns per column group to vary between groups.
2. The blocks will be complete blocks for the checks when the number of check plots per block equals the number of check cultivars. But it is also possible to have incomplete blocks or blocks with extra replication for some or all of the check cultivars. Furthermore, the number of check plots need not be constant across blocks, giving great flexibility regarding the total number v_e of unreplicated entries that can be accommodated.
3. The number of entries that can be tested with this design is

$$v_e = ks - \sum_{i=1}^{g_k} \sum_{j=1}^{g_s} p_{b(i,j)}$$

where $p_{b(i,j)}$ is the number of check plots in the block formed by the intersection of the ith row group and jth column group. In case each block has the same number p_b of check plots, this becomes

D								B	
		B			A			C	
						C			A
			A				D		
	C					D			
A				B					
		D				A			B
			C				A		
	B		C						
				B				D	
C					D				
D				A					
					B			A	
	A			C					
						B	C		
		C							D
			D				B		
		A		D					
					C		B		
					D	A			C
B								A	
	D		B						

Fig. 4. An augmented row–column design for the specification (v_c, g_k, g_s) = (4, 4, 3) (blocks delineated by solid lines). Checks: A–D. The v_e = 173 entries are allocated to the empty cells. There are 47 plots allocated to the check cultivars and 220 plots overall.

$$v_e = ks - g_k g_s p_b$$

4. Restrictions must be observed regarding the number of check cultivars in relation to the number of levels for the design factors. For example, in each row group, the effects of the row–column group design for check cultivars must be estimable for the design Step 3 to return a valid design. This means that the total degrees of freedom for effects of intercept, check cultivars, column groups, and rows must not exceed the number of plots in the row group. Thus, we require for the ith row group that

$$1+(v_c-1)+(g_s-1)+\left[k_{b(i)}-1\right]\le\sum_{j=1}^{g_s}p_{b(i,j)}$$

Similarly, in order for Step 4 to produce a valid row group–column design, for the jth column group we must have

$$1+(v_c-1)+(g_k-1)+\left[s_{b(j)}-1\right]\le\sum_{i=1}^{g_k}p_{b(i,j)}$$

5. The idea to superimpose a third blocking factor in addition to rows and columns is akin to SuDoKu-type experimental designs proposed for replicated experiments by Saba and Sinha (2014), but they did not consider augmented designs.

ANALYSIS

Augmented row–column designs may be analyzed using models with effects for genotypes on the one hand and block factors row and column on the other hand (Wolfinger et al., 1997). For the designs proposed here, further block effects are also needed for row group, column group, and row group–column group combinations due to the restriction on randomization imposed in Step 2. Thus, the full model for a response y can be written as

$$\begin{aligned} y = \mu &+ \text{rowgroup} + \text{colgroup} \\ &+ \text{rowgroup}\cdot\text{colgroup} + \text{rowgroup}\cdot\text{row} \qquad [1] \\ &+ \text{colgroup}\cdot\text{col} + \text{genotype} + e \end{aligned}$$

where μ denotes a general intercept and e is a residual plot error. Table 2 displays a skeleton analysis of variance table showing all sources of variation and their associated degrees of freedom. Block effects, i.e., all effects in Eq. [1] except that for genotype, can be taken as random for recovery of interblock information (Wolfinger et al., 1997). Generally, when a block effect does not represent complete replicates for the checks, there will be information to be recovered and it will usually be worthwhile to take the effect as random (Möhring et al., 2015). An alternative approach is to consider spatial models for plot error (Clarke and Stefanova, 2011). In particular, spatial model components can be used as an add-on to the baseline effects model Eq. [1]. Thus, we may fit model Eq. [1], imposing a spatial covariance model for the plot errors (e). Spatial models assume that the covariance among plots is a decreasing function of spatial

Table 1. Notation used to describe the augmented row–column designs with superimposed blocks.

Symbol	Description
k	no. of rows
s	no. of columns
g_k	no. of row groups
g_s	no. of column groups
k_b	no. of rows per row group [if this is not constant, this needs to be indexed by row group, i.e., $k_{b(i)}$]
s_b	no. of columns per column group [if this is not constant, this needs to be indexed by column group, i.e., $s_{b(j)}$]
p_b	no. of plots per block [if this is not constant this needs to be indexed by blocks, i.e., $p_{b(i,j)}$]
v_c	no. of check cultivars
v_e	no. of unreplicated entries

Table 2. Skeleton analysis of variance table for an augmented design with sources of variation and their degrees of freedom.

Source	General†	df		
		Example 1	Example 2	Example 3
Row groups	$g_k - 1$	1	5	3
Rows, nested within row groups	$k - g_k - 1$	4	30	18
Column groups	$g_s - 1$	2	2	2
Columns, nested within column groups	$s - g_s - 1$	6	15	7
Row groups × column groups (blocks)	$(g_k - 1)(g_s - 1)$	2	10	6
Genotypes	$v_c + v_e - 1$	38	579	176
Error	$(k - 1)(s - 1) - (g_k - 1)(g_s - 1) - (v_c + v_e - 1)$	0	6	7
Corrected total	$ks - 1$	53	647	219

† See Table 1 for the definition of symbols.

distance, and they may improve the precision of the analysis (Piepho and Williams, 2010).

It is advisable to check that a generated design can be analyzed using model Eq. [1], taking all effects as fixed. This can be done by generating a simulated response value for each plot in the design and subjecting these data to a dummy analysis. This was done for all three examples (see SAS files in the supplemental material).

Model Eq. [1] has a simple effect for genotypes. This may be further partitioned to separate out check cultivars and entries (Piepho et al., 2006). In particular, there may be interest in modeling entries as random because they stem from a common breeding population, while check cultivars are fixed because they are purposely selected. To illustrate how the extended model can be coded, assume that there are three replicated check cultivars (Genotypes 1, 2, and 3) and four unreplicated entries (4, 5, 6, and 7). Table 3 shows the coding of variables needed.

The genotype effects can be modeled using the following fixed and random effects:

Fixed: conf + conf.check

Random: conv.entry

The variate *conv* acts as a switch, which switches random effects on for entries (conv = 1) and off for checks (conv = 0). In using this device, it is important to make sure the mixed model package does not automatically mean-center covariates. Estimates for entry effects can be obtained by best linear unbiased prediction (BLUP). Means for the effect conf.check will produce a mean for each check cultivar and an additional mean for the mean of the population of entries.

EXAMPLES

We here consider three examples for the generation of augmented designs using the proposed method. The SAS files for all three examples are provided in the supplemental material.

Example 1

The example in Fig. 2 has the specification $(v_c, g_k, g_s, k_b, s_b, p_b) = (3, 2, 3, 3, 3, 3)$. We find that for each row group

$$1+(v_c-1)+(g_s-1)+\left[k_{b(i)}-1\right]=$$
$$1+2+2+2=$$
$$7<\sum_{j=1}^{g_s} p_{b(i,j)}=9$$

and for each column group

$$1+(v_c-1)+(g_k-1)+\left[s_{b(j)}-1\right]=$$
$$1+2+1+2=$$
$$6=\sum_{i=1}^{g_k} p_{b(i,j)}=6$$

Due to the equality in the second equation, this design leaves no error degrees of freedom (Table 2). This will become apparent in the dummy analysis for this design in that no *F* tests are produced for any of the effects in model Eq. [1] and the denominator degrees of freedom are reported as zero for all tests (not shown). Thus, if a valid estimate of error is required, this design cannot be used and a larger design needs to be chosen.

Occasionally for the settings of this example, our algorithm returns a design such as that in Fig. 1. The key feature of that design is that for all rows, the same type of row appears in both row groups, where the type of row refers to the columns in which the check plots occur (this implies that identical column types appear in column groups as well). Thus, Rows 1 and 4 are identical types, as are Rows 2 and 5, as well as Rows 3 and 6. As noted by Federer and Crossa (2005), row and column effects are not estimable in this case, even when effects pertaining to row and column groups are not fitted. Essentially this occurs because the design is not connected, i.e., there is no closed path from each row to each other row moving from check plot to check plot only within rows or within columns (Searle, 1987, p. 140). A simple check of estimability is to compute adjusted means for all effects in model Eq. [1] using simulated data and taking all effects as fixed. In the case of a design as in Fig. 1, means for rows and columns as well as for the unreplicated entries will be reported to be nonestimable. In the rare event that this should happen, our macro reruns the algorithm

Table 3. Variables and their coded levels for separating out checks (fixed effects) and entries (random effects). All variables are coded as qualitative factors except conv, which is coded as a quantitative variate.

genotype	conv	conf	entry	check
1	0	0	0	1
2	0	0	0	2
3	0	0	0	3
4	1	1	4	0
5	1	1	5	0
6	1	1	6	0
7	1	1	7	0

to generate a new design. If after 10 reruns no connected design is found, a message to this effect is printed in the log window.

Example 2

Consider a design for the specification (v_c, g_k, g_s, k_b, s_b, p_b) = (4, 6, 3, 6, 6, 4) displayed in Fig. 3. The superimposed 6 × 6 blocks are complete blocks with regard to the four check cultivars. Each column is a complete block with four check plots and each row an incomplete block with two check plots. The design permits estimation of all effects for rows, columns, and blocks. It accommodates 576 unreplicated entries and has 648 plots in total. We find that for each row group

$$\begin{aligned} 1+\left(v_c-1\right)+\left(g_s-1\right)+\left[k_{b(i)}-1\right]= \\ 1+3+2+5= \\ 11<\sum_{j=1}^{g_s} p_{b(i,j)}=12 \end{aligned}$$

and for each column group

$$\begin{aligned} 1+\left(v_c-1\right)+\left(g_k-1\right)+\left[s_{b(j)}-1\right]= \\ 1+3+5+5= \\ 14<\sum_{i=1}^{g_k} p_{b(i,j)}=24 \end{aligned}$$

as required.

Example 3

Assume there are v_e = 173 entries to be tested and v_c = 4 check cultivars. Forty-seven plots are to be allocated to checks, so the layout must have 220 plots in total (Fig. 4). These are arranged in k = 22 rows and s = 10 columns. The number of check plots allows 12 blocks to be superimposed, seven of which comprise all four checks (and hence are complete) and five of which comprise only three check plots. The blocks are 6 × 3, 6 × 4, 5 × 3, and 5 × 4 arrays of plots as delineated in Fig. 4. The number of plots per block was chosen deliberately to range from three to five, with the number of plots allocated depending on block size. Thus, e.g., the two 6 × 4 blocks have five check plots each, whereas the four 5 × 3 blocks have three check plots. Each column comprises each check at least once. In each of the four blocks down the central group of columns, there are five check plots and one check is replicated twice. The rows are incomplete blocks, each comprising two or three check plots.

DISCUSSION

The design strategy proposed here involves several levels of blocking to ensure a regular distribution of check plots across the experiment. Alternatively, one may just consider row and column blocking in non-resolvable row–column designs. Such designs are easily generated using software with capabilities to accommodate unequal treatment replication such as OPTEX (Piepho, 2015, Box 10) and CycDesigN 5.1 (VSN International). While these row–column designs also allow adjustments for row and column effects, the distribution of check plots is not as even as with the designs proposed here.

The designs we propose here can be randomized, as is the case with many of the contending augmented designs. An alternative strategy is to use a systematic arrangement of check plots, such as planting a check on every *n*th plot or arranging check plots in a diagonal or knight's move pattern. Clarke and Stefanova (2011) provided a very good summary of such options. Based on analysis of several uniformity trials, they concluded that "the arrangement of check plots, within reason, has little influence" compared with the number of check plots. We consider the randomization protection afforded in augmented designs as an advantage compared with unrandomized designs.

Kempton (1984) asserted that allocating more than 20% of the plots to checks is seldom worthwhile. Clarke and Stefanova (2011) recommended determining the number of check plots so that around 50 error degrees of freedom are available. The designs considered in the examples have substantially fewer degrees of freedom (Table 2). A general disadvantage with augmented designs is that a considerable share of the plots is devoted to genotypes that are not of primary interest. Also, if the checks respond to environmental gradients differently than the entries of interest, the adjustments made in the analysis may be biased. If seed availability permits, a very useful alternative is to partially replicate some of the entries of interest instead of the checks (Smith et al., 2006). Partially replicated designs can be generated using any blocked design for the replicated entries and then augmenting the design with the unreplicated entries in much the same way as with augmented designs (Williams et al., 2011, 2014). In early generations, however, seed may be so limited that partial replication is not an option, and it is in these circumstances that augmented designs such as the ones proposed here have their greatest merit.

APPENDIX

Steps 2, 3, and 4 of the method proposed here were implemented with the SAS procedure OPTEX, which comes with the QC module (not STAT!). There are two macros, which are available as supplemental material. The main one is %get_design. It requires the provision of three input files, as described below. When the groups of rows and groups of columns are equal in size and the number of check plots is constant, these files can be generated using the macro %get_initial_files, specifying the design parameters (g_k, g_s, k_b, s_b, p_b) (otherwise these three files need to be provided separately, as detailed further below). For example, to generate the design in Fig. 2, we used this code:

```
%get_initial_files(g_k=2, g_s=3, k_b=3,
s_b=3, p_b=3);
```

To generate a design with three check cultivars as in Fig. 2, the main macro is subsequently called as

```
%get_design(v_c=3);
```

The generated design is printed as a table that is essentially equivalent in form to those shown in Fig. 2 to 4, showing only the replicated checks. Also, tables are printed showing the number of plots for each of the checks as well as the total number of plots for checks and unreplicated entries. In addition, a SAS file named *design* is generated that contains the full design and includes the block factors *row*, *col*, *rowgroup*, and *colgroup* as well as the treatment factor *trt* for the cultivars (genotypes). The checks are labeled by numbers (levels of trt) ranging from 1 to v_c. The unreplicated entries are labeled $v_c + 1$ to $v_c + v_e$. Note that this code will not reproduce the design exactly as reported in Fig. 2 to 4 because we cannot control the seed of the random number generator used in the repeated calls of OPTEX needed for our method. Thus, each time the macro is run, a different design is generated with the seed initialized from the computer clock.

When at least one of the design parameters g_k, g_s, and p_b is not constant, the three initial files need to be provided manually. This is illustrated with the code used for generating the design in Fig. 4. The first file (layout) specifies the block sizes $p_{b(i,j)}$ for the different blocks (rowgroup–colgroup combinations). The second and third files (rows_per_rowgroup and cols_per_colgroup) specify the number of rows and columns $[k_{b(i)}, s_{b(j)}]$ per row and column group, respectively. After generating these files, the macro %get_design is called:

```
data layout;
input rowgroup colgroup no_of_plots_per_
block;
datalines;
1 1 4
1 2 5
1 3 4
2 1 3
2 2 5
2 3 3
3 1 3
3 2 5
3 3 4
4 1 3
4 2 5
4 3 3
;
data rows_per_rowgroup;
input rowgroup no_of_rows;
datalines;
1 6
2 5
3 6
4 5
;
data cols_per_colgroup;
input colgroup no_of_cols;
datalines;
1 3
2 4
3 3
;
%get_design(v_c=4);
proc print data=design;
run;
```

SUPPLEMENTAL MATERIAL

The following files are made available as supplemental material:

- augmented design.sas: contains the two macros
- Example1.sas, Example2.sas, and Example3.sas: contain the code for generating designs as in Examples 1 to 3.

To run the code for the examples, the macro must first be made available. This can be done by loading the file augmented design.sas into the program editor window and submitting all code. Alternatively, an %include statement can be used, specifying the path to where the file augmented design.sas is located. This is shown in the code for the three examples.

ACKNOWLEDGMENTS

We thank Paul Schmidt (Biostatistics Unit, University of Hohenheim, Germany) for critically checking the SAS files.

REFERENCES

Atkinson, A.C., A.N. Donev, and R.D. Tobias. 2009. Optimum experimental designs with SAS. Oxford Univ. Press, Oxford, UK.

Clarke, G.P.Y., and K.T. Stefanova. 2011. Optimal design for early-generation plant-breeding trials with unreplicated or partially replicated test lines. Aust. N.Z. J. Stat. 53:461–480. doi:10.1111/j.1467-842X.2011.00642.x

Federer, W.T. 1956. Augmented (or hoonuiaku) designs. Hawaii. Sugar Plant. Rec. 55:191–208.

Federer, W.T. 1961. Augmented designs with one-way elimination of heterogeneity. Biometrics 17:447–473. doi:10.2307/2527837

Federer, W.T. 2002. Construction and analysis of an augmented lattice square design. Biom. J. 44:251–257. doi:10.1002/1521-4036(200203)44:2<251::AID-BIMJ251>3.0.CO;2-N

Federer, W.T., and J. Crossa. 2005. Designing for and analyzing results from field experiments. J. Crop Improv. 14:29–50. doi:10.1300/J411v14n01_04

Federer, W.T., R.C. Nair, and D. Raghavarao. 1975. Some augmented row–column designs. Biometrics 31:361–375. doi:10.2307/2529426

Kempton, R.A. 1984. The design and analysis of unreplicated field trials. Vortr. Pflanzenzuecht. 7:219–242.

Lin, C.S., and G. Poushinsky. 1983. A modified augmented design for an early stage of plant selection involving a large number of test lines without replication. Biometrics 39:553–561. doi:10.2307/2531083

Möhring, J., H.-P. Piepho, and E.R. Williams. 2015. Inter-block information: To recover or not to recover it? Theor. Appl. Genet. 128:1541–1554. doi:10.1007/s00122-015-2530-0

Piepho, H.-P. 2015. Generating efficient designs for comparative experiments using the SAS procedure OPTEX. Commun. Biometry Crop Sci. 10:96–114.

Piepho, H.-P., and E.R. Williams. 2010. Linear variance models for plant breeding trials. Plant Breed. 129:1–8. doi:10.1111/j.1439-0523.2009.01654.x

Piepho, H.-P., E.R. Williams, and M. Fleck. 2006. A note on the analysis of designed experiments with complex treatment structure. HortScience 41:446–452.

Saba, M.F.A., and B.K. Sinha. 2014. SuDoKu as an experimental design: Beyond the traditional Latin square design. Stat. Appl. 12:15–20.

Searle, S.R. 1987. Linear models for unbalanced data. John Wiley & Sons, New York.

Smith, A.B., P. Lim, and B.R. Cullis. 2006. The design and analysis of multi-phase plant breeding experiments. J. Agric. Sci. 144:393–409. doi:10.1017/S0021859606006319

Williams, E.R., and J.A. John. 2003. A note on the design of unreplicated trials. Biom. J. 45:751–757. doi:10.1002/bimj.200390046

Williams, E.R., J.A. John, and D. Whitaker. 2014. Construction of more flexible and efficient p-rep designs. Aust. N.Z. J. Stat. 56:89–96. doi:10.1111/anzs.12068

Williams, E.R., H.-P. Piepho, and D. Whitaker. 2011. Augmented p-rep designs. Biom. J. 53:19–27. doi:10.1002/bimj.201000102

Wolfinger, R.D., W.T. Federer, and O. Cordero-Brana. 1997. Recovering information in augmented designs, using SAS PROC GLM and PROC MIXED. Agron. J. 89:856–859. doi:10.2134/agronj1997.00021962008900060002x