Computers and Electronics in Agriculture 122 (2016) 156–160

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

journal homepage: www.elsevier.com/locate/compag

Application note

alfaDRA: A program for automatic elimination of variety self-proximities in alpha-design

J. Janová *, D. Hampel

Department of Statistics and Operation Analysis, Faculty of Business and Economics, Mendel University in Brno, 613 00 Brno, Czech Republic

ARTICLE INFO

Article history:
Received 11 December 2015
Accepted 21 January 2016
Available online 10 February 2016

Keywords:
Alpha-design
Damage robustness
Proximity constraints
Simulation
Variety proximity

ABSTRACT

When using alpha-design for plant variety testing under space restrictions, ex post design modifications must be implemented to prevent variety self-proximity on plots and, consequently, to prevent damage-induced loss of experimental information. This is done ad hoc for each experiment; the unsystematic modification is, however, commonly not only unable to resolve all existing proximities, but may introduce secondary undesired proximities. In this paper, a procedure is developed for the universal construction of modified alpha-design that covers all existing proximity constraints while keeping the efficiency level of the original design. Using extensive real data simulation, we validate the procedure and confirm high damage robustness of the modified designs. The procedure has been implemented as a Matlab function and is available as on-line supplement to the paper. The function enables to design the damage-robust experiments automatically using only standard computer equipment.

1. Introduction

In our contribution we follow the experience and practice of the Central Institute for Supervising and Testing in Agriculture (CISTA)[1] that implements alpha-design for professional plant variety testing experiments in the Czech Republic. Typically, the experiments at CISTA are small-scale, i.e. with a low number of replicates, each often separated into two or more consecutive rows to reflect the restricted experimental area (see Fig. 1). Alpha-design allows for variety self-proximity, i.e. proximity of the same varieties placed in plots in different replicates, which makes the experimental design susceptible to information loss in case of local damage. While for large-scale experiments (with more replicates on a larger area) this may be overlapped by taking advantage of information from other subareas, in case of limited experimental area the partial damage impacts may be fatal.

To avoid undesirable variety self-proximities, a modification of the original alpha-design is essential for small-scale experiments and favorable for large-scale ones. Commonly, the modifications must be done manually ad hoc just before implementing the design into an experiment, which is time demanding and mostly does not facilitate elimination of all proximities. The aim of this paper is to provide a general design modification algorithm that can be directly implemented within the established design planning procedure and that enables to obtain the damage-robust designs automatically while keeping the efficiency level of the original alpha-design.

To the best of our knowledge, no attention has been given to such universal procedure enabling general and automatic elimination of all undesired proximities in alpha-design. Some research on modifications of the designs has been done, however, with different intentions then improving the damage robustness. We may identify the class of neighbor balanced problems as introduced in Wild and Williams (1987), which refer to eliminating the neighbor effects. Spatial balanced designs resolved in van Es et al. (2007) deals with complete block design and soil heterogeneity problem. In Williams (1986) the alpha-design concept was extended into latinized designs and in John and Williams (1998) the method was advanced to generate *t*-latinized row-column designs which are able to automatically prevent some of proximities in our problem. Nevertheless, incorporating latinized structures can affect the efficiency of the design. The new method developed in this contribution automatically eliminates all undesired proximities while keeping characteristics and efficiency of the original alpha-design.

2. Specification of proximity constraints

The undesired localization of the same varieties that is to be prevented we denote as *proximity constraints*. As shown in Fig. 2,

* Corresponding author.
E-mail address: janova@mendelu.cz (J. Janová).

[1] CISTA is a specialized body established by the Ministry of Agriculture of the Czech Republic for expert tasks in plant variety testing. Web pages of the institution: http://eagri.cz/public/web/en/ukzuz/portal/.

http://dx.doi.org/10.1016/j.compag.2016.01.022
0168-1699/

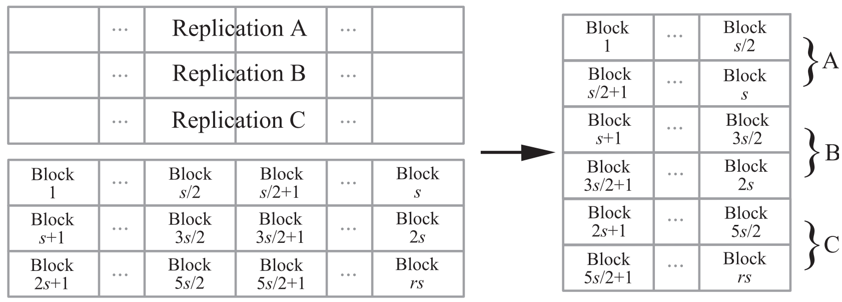

Fig. 1. Breaking the replication into two rows.

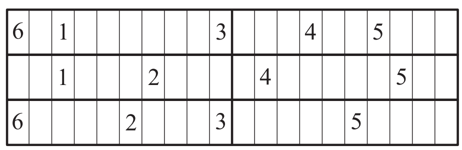

Fig. 2. Undesirable positioning of the same variety.

we identify six prohibited situations in ordinary design organization. Each row in the scheme represents a replicate with two blocks containing ten plots each. The numbers represent a variety; combinations of the same numbers show undesired self-proximities. Further on, localizations 1–5 and 6 will be denoted as *proximities 1–5* and *frontier proximity*, respectively.

Proximities 1–3 are undesired and should not appear in the design implemented. Proximity 4 is acceptable if there is not enough degrees of freedom in the given experiment (i.e. due to a low number of varieties or in case of short blocks). Strictly prohibited proximity 5 is a combination of 2 and 4. Frontier proximity is unacceptable because the frontier area is systematically in higher danger of damage. Note also, that the opportunity for appearance of frontier proximity increases in case of breaking the replicates into two or more consecutive rows as common in space-limited experiments.

3. Modification procedure

3.1. Prerequisites

To obey proximity constraints, we modify the design via rotation of varieties inside the block and/or rearranging the sequence of blocks within a replicate. These operations do not affect the characteristics of alpha-design and therefore the efficiency level of the modified design remains the same as in the original alpha-design.

The alpha-design is generated according to Patterson and Williams (1976) for a given structure of the experiment. We denote v number of species, r number of replicates (we consider $r \leqslant 4$), k the length of the (longer) block (we consider $k > 4$), w the number of rows within a replicate, b the number of blocks in a row and s the number of blocks in a replicate. In alpha-design, we allow for two possible lengths of block differing by one plot. Generally, there is a freedom in lining up the blocks within a row of the design. However, in our modification the arrangement of blocks must obey certain rules. Denote K and D the blocks with length k and $k-1$, respectively, and denote the position of the block in a row from left to right by $1, \ldots, b$; then the arrangement of blocks must obey the following rules.

Rule 1 (Blocks within replicate): In each row (it is either a whole replicate or just its part), the blocks K and D are lined up in the same sequence on positions $1, \ldots, b-1$. If possible, the last block in a row is K.
Rule 2 (Blocks within experiment): The rows from different replicates that are ordered below each other have the same lining up of the blocks.
Rule 3 (First block in row): If in accordance with Rules 1 and 2, the first block in a row is D for $k \neq 6$ and K for $k = 6$.

3.2. Eliminating proximities 1–5

Multiple appearance of one variety is not possible within a replicate of alpha-design. The undesired proximities may appear between the last row of a replicate and the first row of the following one. We will reallocate the variety symbols to plots within all blocks of a replicate in a way that the variety proximities will be eliminated automatically.

For demonstration of our method, let us consider example from Patterson and Williams (1976) under space restrictions. Dividing each replicate in the corresponding experimental design into two rows produces undesired proximities of varieties (see Table 1, proximities of type 1 are in bold). Taking advantage of the notation introduced in Wild and Williams (1987), we index the elements of the alpha-array with their column number and obtain array A in Table 2. An element p_q rotates the varieties on positions $(q-1)s+1$ to qs in a matrix of the varieties. Now we can rearrange the elements in columns of alpha-array A, yet keep information which varieties an element of the array is linked to. To produce modified alpha-design without undesired proximities, we use the following rotation of elements in columns of array A. Denote $A[j,i]$ the element of array A on a row j and in a column i. Alpha-array modification algorithm (further simply *Algorithm*) is of the following form.

Set $A_m = A$
Set $a = 2$
For $i = 2$ to $i = r$
  If $k = 6$
    $a \leftarrow a + (-1)^i$
  Else
    $h = 3 * (i-1)$
    $a = \text{modulo}\,(h/k)$
  For $j = 1$ to $j = a$
    $A_m[j,i] \leftarrow A[k-a+j,i]$
  For $j = 1$ to $j = k-a$
    $A_m[j+a,i] \leftarrow A[j,i]$
Output A_m

In our example, *Algorithm* produces modified alpha-array A_m in Table 2 and consequently the modified experimental design shown in Table 3. *Algorithm* is appropriate for blocks with $k > 4$; for $k > 6$