Abschlussbericht
- Versuchsdesign -

Nutzerhandbuch für das Verfahren ‚DESIGN'
als Softwarelösung unter PIAFStat

Forschungs-Nr.: 6/16

Laufzeit: 1 / 2017 - 3 / 2018

verantw. Themenbearbeiter: **Dr. V. Michel, Dr. A. Zenk**

Beteiligte Einrichtungen: **Universität Hohenheim, Biostatistik & Universität Canberra (Australien)**

BioMath GmbH Rostock

Bund-Länder-Gemeinschaft PIAF

31.3.2018

_____

Themenbearbeiter

_____

Institutsleiter

GLIEDERUNG Seite

1 Zusammenfassung

Im Rahmen der Pflege und Weiterentwicklung der Softwarelösung der Länder und des Bundes für das Feldversuchswesen PIAF wurde die Softwarelösung DESIGN erstellt, die die Anwender in der methodischen Versuchsplanung, insbesondere der Optimierung des Versuchsdesigns und in der Bereitstellung von Randomisationsplänen effizient unterstützt. Die Gesamtkonzipierung und Koordinierung lag bei der LFA MV, SG Sortenwesen und Biostatistik. Wissenschaftlichen Grundlagen wurden in Kooperation zwischen der Universität Hohenheim, der LFA MV und der Universität Canberra (Australien) erarbeitet und veröffentlicht. Die Fa. BioMath (Rostock) übernahm Auftragsleistungen zur Erstellung von SAS®-Makros - die Finanzierung dieser Teilleistungen erfolgt durch Mittel der Bund-Länder-Gemeinschaft PIAF.

Das Verfahren DESIGN unterstützt Versuchsansteller im angewandten Versuchswesen nutzerfreundlich, effizient und auf aktuellstem wissenschaftlichem wie ebenso praktischem Niveau.

Die LFA stellt den PIAF-Anwendern in MV und in den anderen Bundesländern sowie in Bundeseinrichtungen die Software DESIGN und mit diesem Abschlussbericht ein Nutzerhandbuch zur Verfügung. Grundlegende Hinweise wurden in Ampelfarben herausgestellt, im Sinne von zu- oder abratenden Empfehlungen (grün vs. rot) sowie Mahnungen zur Vorsicht (gelb).

Die Nutzeroptionen werden in Abschnitt 6 vorgestellt und erläutert. Im Abschnitt 7 werden zunächst die Zielkriterien für gegenüber verschiedensten Störungsarten robuste Versuchsdesigns definiert. Nachfolgend wird - beginnend mit simplen und dann zunehmend komplexen Designs - die Wirksamkeit von Designvorgaben für die Verbesserung der Zielkriterien aufgezeigt. Die möglichen Kombinationen von Designvorgaben sind sehr vielfältig, sie gehen weit über die bisher verbreiteten ‚Anlagemethoden' (wie einfaktorielle Blockanlage, Lateinisches Rechteck, Alpha-Gitter ...) hinaus. Daher werden Empfehlungen bzw. Strategien für Softwarenutzer gegeben. Abschnitt 8 stellt die Schnittstellen zwischen den Modulen PIAFStat / Verfahren DESIGN und der PIAF-Datenbank vor. Darauf aufbauend wird In Abschnitt 9 gezeigt, wie die Designvorgaben dann in der späteren Einzelversuchsauswertung mit PIAFStat (EVA bzw. ZVA) wirksam werden - das Ziel ist ein in sich stimmiges Gesamtsystem und die Einheit von Design und Auswertungsmodell.

Die Palette der für PIAFStat entwickelten Verfahren wird durch DESIGN komplettiert. Die Abfolge der Versuchsbearbeitung in dem Verfahrens-Komplex DESIGN → PLAUSI → EVA/ZVA → HOHENHEIM-GÜLZOWER-SERIENAUSWERTUNG baut logisch aufeinander auf und stellt einen in Nutzerführung und Funktionalität stimmigen Gesamtkomplex dar. Dies gewährleistet, dass von Versuchsplanung über Plausibilitätsprüfung, Einzelversuchsauswertung bis hin zu hochkomplexen Serienauswertungen ein Prozess stattfindet, welcher die Chancen auf Fehlerminimierung und Verbesserung von Wiederholbarkeit und Reproduzierbarkeit verdichteter Ergebnisse für die Praxis erhöht.

2 Ziele

- Erhöhung der Präzision und Reproduzierbarkeit von Versuchsergebnissen
- Minderung von Störungen durch Bodenheterogenität, durch versuchstechnische Einflüsse und durch Nachbarschaftseffekte zwischen Prüfgliedern
  (Verbindung des Erfahrungswissens um die typischsten Fehlerquellen mit Prinzipien der Randomisationstheorie)
- Entwicklung einer nutzerfreundlichen Softwarelösung im PIAF-System
  (Integration eines PIAFStat-Verfahrens zur Konstruktion des Versuchsdesigns, der Randomisationspläne und zum Einlesen der Randomisationspläne und Blockungs-Variablen in PIAF)
- Bereitstellung eines entsprechenden PIAFStat-Verfahrens mit Nutzerhandbuch für die Bund-Länder-Gemeinschaft PIAF

3 Einführung

Situation im PIAF - Softwarekonstrukt

PIAF als umfassende Softwarelösung im Feldversuchswesen (F) soll Planung (P), Informationssystem (I) und Auswertung (A) beinhalten. Ein essentieller Bestandteil jeder Versuchs-Planung ist die Gestaltung des Versuchsdesigns und daraus abgeleitet die Erstellung von Randomisationsplänen für jeden Einzelversuch. Gerade dieser Aspekt war aber vom konzeptionellen Beginn (1992) bis 2017 nicht implementiert - weder innerhalb des PIAF-Systems, noch in Form einer Einleseschnittstelle aus existierenden Softwarelösungen (CycDesign®, FELD_VA u.a.).

Es bot sich an, bei der Konzipierung einer Softwarelösung im PIAF-System die jüngsten Innovationen von vornherein zu berücksichtigen. Dabei werden konkret für den PIAF-Nutzerkreis die häufigsten und typischsten Situationen im Design der Feldversuche besonders berücksichtigt. Bei einer Vielzahl möglicher, aber jeweils äußerst selten verwendeter Ausnahme-Designs (z.B. dreifaktorielle zweistufige Streifen-Spaltanlage A+[B/C]-Bl uvm.) sollte aber die bisherige, etwas aufwändigere Herangehensweise fortgeführt werden (Nutzung anderer Planungsgrundlagen und Eingabe des Randomisationsplanes per Hand).

Veranlassung für Innovationen in Design und Randomisation

Die Erstellung von Versuchsdesigns erfolgte bei den meisten Nutzern auf sehr veraltetem Stand, Randomisations-Prinzipien wurden oft verletzt und in der Auswertung stimmten Design und Modell häufig nicht überein.

Um die 80'er Jahre des 20. Jh. entstand folgende Divergenz:

- einerseits: biometrische Randomisations-Prinzipen mit immer weitergehender kombinatorischer Verfeinerung (halbblancierte Gitterquadrate, Youden-Quadrate, Kubische Gitter, Rechteckgitter …), aber zunehmend abgekoppelt von realen praktischen Fehlerquellen; diese Anlagen hielten in praxi kaum Einzug / hatten wenig praktische Resonanz; derartige unvollständige Blockungsprinzipien können zwar ggf. einen Teil der Varianz fassen, passen aber im Ansatz kaum noch zu den Strukturen der Fehlerursachen (diskrete Blockgrenzen versus gleitenden Bodentrends u.v.m.)
- anderseits: im angewandten Versuchswesen Tendenz zur Nutzung immer gleicher veröffentlichter ‚gerechter Pläne', die den typischen Fehlerquellen lt. praktischen Erfahrungen gerechter werden sollten, aber häufig mit der Randomisationstheorie kollidierten und dann suboptimal im Ergebnis sind (Güte der Fassung der Varianzen; Qualität von Wahrscheinlichkeitsaussagen, korrelierte Fehler in Versuchsserien).

Das Verfahrens DESIGN soll diese Divergenz auflösen, also die begründeten praktischen Anforderungen mit der Randomisationstheorie in Einklang und Design, Randomisation und Auswertung in Deckung bringen.

4 Aufgabenteilung

4.1 LFA, Sachgebiet Sortenwesen und Biostatistik

- Konzeption und Koordinierung
- Mitwirkung an den wissenschaftlich-methodischen Grundlagen, insbesondere Einbringen der praxisrelevanten Anforderungen und Erfahrungen aus dem angewandten Versuchswesen
- Öffentlichkeitsarbeit in Biometrischer Fachwelt und im angewandten Versuchswesen
- Begleitung, Abstimmung und Testung der Programmierleistungen der Fa. BioMath
- Erstellung des PIAFStat-Verfahrens ‚DESIGN'
- Schulung der Bund-Länder-Gemeinschaft PIAF
- Erstellung dieses Nutzerhandbuches für das der PIAFStat-Verfahren ‚DESIGN'

4.2 Bund-Länder-Gemeinschaft PIAF

- Abstimmung über die Konzepte in den Arbeitsgruppen PIAF-Auswertung, PIAF-Allgemein und PIAF-PSM
- Beschluss über Gemeinschaftsfinanzierungen in PIAF-Koordinierungsgruppe
- Erweiterung der SAS-Gemeinschafts-Lizensierung für PIAF um das QC-Modul; Abruf dieses Moduls und Umlage-Finanzierung durch jede Länderdienststelle; Koordinierung durch Leiter der PIAF-Koordinierungsgruppe (erledigt)

Auszug aus dem Protokoll der Sitzung der AF PIAF-Auswertung, 10./11.10.16, Würzburg: „Maßnahme ‚Design und Randomisation' wird von allen Teilnehmern als wichtigster Schwerpunkt für 2017 ausgewählt, andere Maßnamepunkte wurden vorgestellt, werden aber zurückgestellt ..."

4.3 Universität Hohenheim, Biostatistik

- Mitwirkung bei wissenschaftlich-methodischen Grundlagen, insbesondere Federführung bei biometrisch-mathematischen Aspekten
- methodische Abstimmung und Einbeziehung der Universität Canberra (Australien), E.R. Williams
- Federführung bei internationalen wissenschaftlichen Veröffentlichungen
- Umsetzung in einem Grundmakro unter dem SAS - QC-Modul

4.4 Fa. BioMath GmbH, Rostock

Die Fa. BioMath erstellt im Auftrag der PIAF-Gemeinschaft SAS-Makros als Bausteine für Das PIAFStat-Verfahren DESIGN (Tab. 1-3).

Tabelle 1: SAS-Makros für das Verfahren DESIGN (Stand 2017)

Makro Name	**Funktionalität**
MST	Hilfs-Makro für GenDesign (mitgeliefert über Uni Hohenheim)
GenDesign	erzeugt ein Versuchsdesign nach Algorithmus der Uni Hohenheim
GenNDesigns	erzeugt N Versuchsdesigns
CalcBestDesigns	berechnet die Bewertung der Designs und rangordnet sie
GetBestDesigns	wählt die k besten Designs nach Bewertung aus
AddPermutationsDesigns	generiert aus einem Design weitere Permutatins-Designs
AddRepTreatments	fügt Wdh der Behandlungen hinzu
AddRepTreatmentsAndBlocks	fügt Wdh der KT/GT hinzu
AddPlotCodes	fügt Parzellenkürzel hinzu
MakeFirstRepSystematic	erste Wdh systematisch

Tabelle 2: Bewertungsalgorithmus im Makro CalcBestDesigns (Stand 2017)

Variable Name	**Bedeutung**
RunID	Nummer des Runs
max_row_neighbor	max. Anzahl der paarweisen Nachbarschaft zweier treatments unter allen treatment-Paaren (Nachbarschaft nur in Zeilen berücksichtigt) (z.B.: 5 und 7 sowie 3 und 9 sind je 5 mal direkt benachbart)
n_max_row_neighbor	Anzahl treatment-Paare mit ‚max_row_neighbor' (im vorigen Beispiel kommt die max. Nachbarschaft 2 mal vor)
Maßzahlen für gleichmäßige Verteilung der Wiederholungen je treatment (ED - *even distribution*)	
NND_1	NND - *nearest neighbour distance* = Mittelwert der euklidischen Distanzen der (w) *nearest neighbour plots* eines treatments; Default: w=2; NND_1 - minimale NND unter allen treatments
trt_1_NND_1	Nummer des treatment mit NND_1
NNLD_1	wie NND, aber geometrisches Mittel der Distanzen (log-transformierte Distanzen & Rücktransformation); Ziel: große Abstände gedämpft wichten
trt_1_NNLD_1	Nummer des treatment mit NNLD_1
MST_1	MST - *minimum spanning tree* (nach Moser 1992) = Mittelwert der Längen der Linien eines *minimium spanning tree*, welcher alle Wiederholungen eines treatments verbindet; MST_1 - minimaler MST unter allen treatments
trt_1_MST_1	Nummer des treatment mit MST_1
MSTL_1	wie MST, aber geometrisches Mittel der Distanzen
trt_1_MSTL_1	Nummer des treatment mit MSTL_1
mean_MSTL	Mittelwert der MSTL über alle treatments
var_MSTL	Varianz der MSTL über alle treatments
f_bar_A	mittlerer Effizienzfaktor gegenüber einem Zeilen-Spalten-Plan mit festen Zeilen- und Spalteneffekten; s.a. *baseline efficiency* nach Piepho et al. 2016
seed	Anfangswert des Zufallsgenerators

Tabelle 3: Variablenerweiterung für Property-Ergebnis-Dataset (Stand 2017)

Variable Name	**Bedeutung**
DesignID	Nummer des Ranking
ED_weighted	gewichtetes Abstandsmaß = 0.5 * MSTL_1 + 0.2 * NNLD_1 + 0.2 * MST_1 + 0.1 * NND_1
Mean_n_max_row_neighbor	Mittelwert von n_max_row_neighbor je max_row_neighbor
Mean_ED_weighted	Mittelwert der gewichteten Abstandsmaße je max_row_neighbor
NB	relative Abweichung = -1 * (n_max_row_neighbor - Mean_n_max_row_neighbor) * 100 / Mean_n_max_row_neighbor
ED	relative Abweichung = (ED_weighted - Mean_ED_weighted) * 100 / Mean_ED_weighted
NB_weight	vorgegebenes Gewicht für die Nachbarschaften
Rating	Bewertung = NB_weight * NB + ED

Das Ranking erfolgt aufgrund einer vierstufig-hierarchischen Sortierung und wird in der Variablen DesignID gespeichert(Stand Sep. 2017, Änderungen ab ca. Juni 2018).

1. übergeordnet aufsteigend nach ‚max_row_neighbor'
2. darunter (bei Gleichheit in 1.) absteigend nach ‚Rating'
3. darunter (bei Gleichheit in 1.-2.) absteigend nach ‚f_bar_A'

   (in 2018 soll eine direkte Einbeziehung in die Wichtung / Berechnung des 'Rating' erfolgen)
4. darunter (bei Gleichheit in 1.-3.) aufsteigend nach ‚RunID'

Die Zielstellung des Makros *MakeFirstRowSystematic* ist in 6.9.4 / zur O64 beschrieben.

5 Prophylaktische Berücksichtigung typischer Fehlerquellen durch das Verfahren ‚DESIGN'

5.1 Typische Quellen des Versuchsfehlers (Fehlerarten)

Die unterschiedlichen Herangehensweisen der Versuchsansteller in der Gestaltung des Versuchsdesigns stehen in engem Zusammenhang mit der prinzipiellen Art zu erwartenden Störungen. Hierbei wird allerdings davon ausgegangen, dass es vor Versuchsanlage Anhaltspunkte oder Vorkenntnisse für die konkrete Art und Ausprägung möglicher Störungen gibt. Gerade dies ist allerdings sehr häufig nicht oder nicht hinreichend gegeben. Vielmehr können erfahrungsgemäß die meisten eingetretenen Problemarten rückblickend als nicht konkret erwartet eingeschätzt werden. Allerdings ist die Anzahl der maßgeblichen Arten von Störungen überschaubar (Tab. 4). Konkret und individuell ist dann letztlich die Ausprägung im einzelnen Versuch.

Entgegen dem Herangehen, bei der Wahl des Versuchsdesigns im Wesentlichen eine dominante Störwirkung zu berücksichtigen, beinhaltet das nachfolgend diskutierte Konzept den Ansatz, drei wesentlichen Arten von Störungen gleichzeitig zu berücksichtigen. Tritt dann die eine oder andere Störung auf, ggf. auch in Kombination, so soll (1) diese Störung im Auswertungsmodell berücksichtigt werden können und (2), wenn letzteres aus unterschiedlichen Gründen nicht erfolgt (z.B. weil nicht erkannt, überlagert, Modell überfrachtet / konvergiert nicht uvm.) so soll das Design und die konkrete Randomisation bewirken, dass die Mittelwertdifferenzen auch ohne Berücksichtigung im Auswertungsmodell möglichst kaum verzerrt geschätzt werden (auch wenn es sich dann nicht im geschätzten Schätzfehler niederschlägt, letzteres wäre allerdings optimal).

Störungen, welche nur Einzelparzellen betreffen, ohne das ein räumlichen Zusammenhang über mehrere Parzellen besteht, können nicht durch die Designwahl gegriffen werden, sondern nur durch intensive Beobachtung und Dokumentation und ggf. im Zuge der Auswertung. Im Weiteren werden daher nur Störungen berücksichtigt, bei denen ein räumlicher Zusammenhang besteht. In Tab. 4 werden die drei typischen Arten räumlich verbundener Störungen klassifiziert und Möglichkeiten für die Berücksichtigung aufgezeigt.

Selbstverständlich kann ein derart komplexes kombiniertes Design nicht bis zu der Güte z.B. der Nachbarschaftsbalancierung geführt werden wie ein Design, dass ausschließlich Nachbarschaftsbalancierung anstrebt. Gleiches gilt im Vergleich zu reinen Zeilen-Spalten-Plänen etc. Ein gewisser Effizienzverlust bei jedem Einzelziel wird im Interesse einer harmonisch ausgewogenen vorbeugenden Risikominderung hingenommen. Insofern ist der hier vorgestellte Ansatz vom eher praxisbezogenen pragmatischen Gesichtspunkt der Risikominimierung hergeleitet (Relativer Gewinn bei vorher unberücksichtigtem Parameter übersteigt die relative Effizienzeinbuße bei vorher allein betrachtetem Parameter.

5.1.1 Bodenheterogenität - Latinisierung und räumliche Verteilung

Es wird davon ausgegangen, dass Bodenheterogenität i.d.R. in der Fläche gleitend und nicht distinkt vorliegt - insofern ist Blockbildung a priori nur bedingt geeignet und kann nur anteilig Fehler minimierend wirken. Der Natur dieser Fehlerquelle könnten räumliche Ansätze - Autokorrelation oder Trends - auswertungsseitig häufig eher entsprechen.

Abgeleitet aus dieser Betrachtung sollte eine gleichmäßige räumliche Verteilung der Wiederholungen je Prüfglied prophylaktisch wirkungsvoll sein. Zur Bewertung dieser Verteilung sollen geeignete Abstandsmaße je Prüfglied gefunden und ausgewiesen werden, welche zu maximieren sind (umgesetzt: siehe Maßzahlen für gleichmäßige Verteilung der Wiederholungen je Prüfglied in 3.3.4.).

Dieses Ziel soll durch zwei parallele Strategien berücksichtigt werden:

- Einbindung eines Parameters für die räumliche Korrelation in der Versuchsfläche (Parameter im Verfahren: ‚spatial_corr_ED'). Diese Korrelation ist nicht nur für die unmittelbaren Nachbarn definiert, sondern setzt sich abnehmend in der Fläche in alle Richtungen fort.
- Latinisierung in ein bis drei Hauptrichtungen

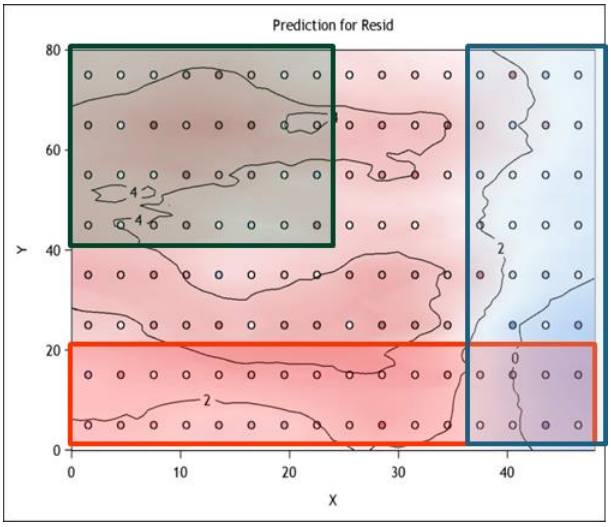


Abb. 1 zeigt die Bodenheterogenität in einem realen Versuch der LFA (geschätzt aus den Residuen). Des Weiteren sind drei unterschiedliche Ansätze für die Bildung von Blocks mit unterschiedlicher Blockform (Zeilen x Spalten-Struktur) gezeigt. Je Blockform lassen sich in diesem Fall vier vollständige Wiederholungen bilden. Die Blockungen können beliebig kombiniert werden. es können also 0 bis 3 latinisierte Blockungen im Design berücksichtigt werden.

rot - Zeilengruppen
blau - Säulen bzw. Spaltengruppen
grün - ‚gebrochene Blocks' bzw. Superblocks

Abbildung 1: Bodenheterogenität und drei kombinierbare Latinisierungsformen

Die simultane Nutzung einer dreifachen Latinisierung wird in Abb. 2 am Beispiel der Konstellation $v\_r\_k\_s = 15\_3\_6\_10$ beispielhaft gezeigt (bezüglich der Anzahl der Zeilen- und Spaltengruppen und der Blockform bereits voroptimiert).

- Sofern vollständige Wiederholungen einbezogen wurden, so waren dies $4 = 2 * 2$ ‚Superblocks' (SB) (Abb. 2 links)
- Zeilengruppen (ZG) wurden ggf. als 3 unbalancierte rechteckige Gruppen vorgegeben (Abb. 2 Mitte).
- Spaltengruppen (SG) wurden ggf. als 4 unbalancierte rechteckige Gruppen vorgegeben (Abb. 2 rechts).

Z	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6	Sp7	Sp8	Sp9	Sp10
6	14	8	12	7	3	15	2	13	11	9
5	6	4	11	5	13	1	8	7	10	12
4	1	15	10	2	9	6	5	4	14	3
3	9	14	5	3	1	7	15	6	13	10
2	2	11	15	12	6	3	4	8	9	5
1	4	10	8	13	7	2	1	11	12	14

11	13	3	6	1	7	14	9	4	12
15	9	5	4	14	8	2	3	1	10
2	7	8	10	12	11	15	5	13	6
1	15	4	11	3	9	13	14	10	7
13	12	6	9	8	1	5	11	2	4
14	2	10	5	7	12	3	8	6	15

11	13	3	6	1	7	14	9	4	12
15	9	5	4	14	8	2	3	1	10
2	7	8	10	12	11	15	5	13	6
1	15	4	11	3	9	13	14	10	7
13	12	6	9	8	1	5	11	2	4
14	2	10	5	7	12	3	8	6	15

Abbildung 2: Simultane Dreifach-Latinisierung in einem Beispiel-Design

In der Auswertung mit PIAFStat -EVA oder -ZVA können ggf. beide Ansätze (räumliche Korrelation und/oder Latinisierung) in Kombination in einem Ausgangsmodell berücksichtigt werden. Im Zuge einer objektivierten Modellreduktion (AICC) können letztendlich kein, einer oder beide Ansätze im Endmodell verbleiben.

Die Kombination von Latinisierung mit der Nutzung der Kovarianz_ED ist der Schlüssel für eine gleichmäßige Verteilung aller Prüfglieder.

Dabei zielt der räumliche Kovarianz-Parameter eher auf den Nahbereich ab, während die Latinisierung eher die Verteilung über die Gesamtfläche, also den Fernbereich positiv beeinflusst.

5.1.2 Nachbarschafts - Beeinflussungen

Bei vielen Kulturarten berühren sich die Prüfglieder ohne Einmantelung an der langen Parzellenkante. Zwischen diesen benachbarten Prüfgliedern können konkurrierende Interaktionen eintreten, die die Schätzung von Prüfgliedmittelwerten und - differenzen verzerren. Werden regelmäßig relevante Nachbarschaftseffekte erwartet, so ist eine Ummantelung zwingend erforderlich (plot-in-plot-Parzellen bei Raps, Kern- und Randreihen bei Mais, Ummantelung von Düngungsparzellen oder Kontrollparzellen etc.). Aber auch wenn nach Abwägung von Aufwand-Nutzen eine Einmantelung unterbleibt, sind Nachbarschaftseffekte nicht gänzlich auszuschließen. Da häufig von einzelnen Prüfgliedern besonders intensive Nachbarschaftseffekte ausgehen (z.B. sehr kurze oder sehr lange Sorte), sollten die paarweisen Nachbarschaften bestmöglich ausbalanciert werden, sodass kein Prüfgliedpaar öfter als vermeidbar miteinander benachbart ist.

Das Beispiel in Abb. 3. soll die Relevanz demonstrieren. In einem Triticale-LSV der LFA (2015/16, Gülzow) winterte die Sorte ‚18' komplett aus. Zwar war sie danach sachlich ohnehin regional nicht mehr empfehlungswürdig. Leider können bei derart gravierenden Beeinträchtigungen dann auch die benachbarten Parzellen nicht mehr gewertet werden, da sie von dem unbewachsenen Nachbarraum extrem profitieren oder aber ins Lager gehen etc. Wäre keine Sorte mehr als einmal zur Sorte 18 benachbart, so wäre die Auswertbarkeit des Versuches noch gegeben, wenn auch beeinträchtigt. Allerdings erwies sich die Randomisation im Nachhinein als abträglich, da das Prüfglied 5 dreimal zur 18 benachbart war. Daraufhin war auch die Auswertbarkeit des Prüfglieds 5 nicht mehr gegeben. Unter Berücksichtigung einer bestmöglichen Nachbarschaftsbalancierung im Prozess der Versuchsplanung wäre dies nicht eingetreten.

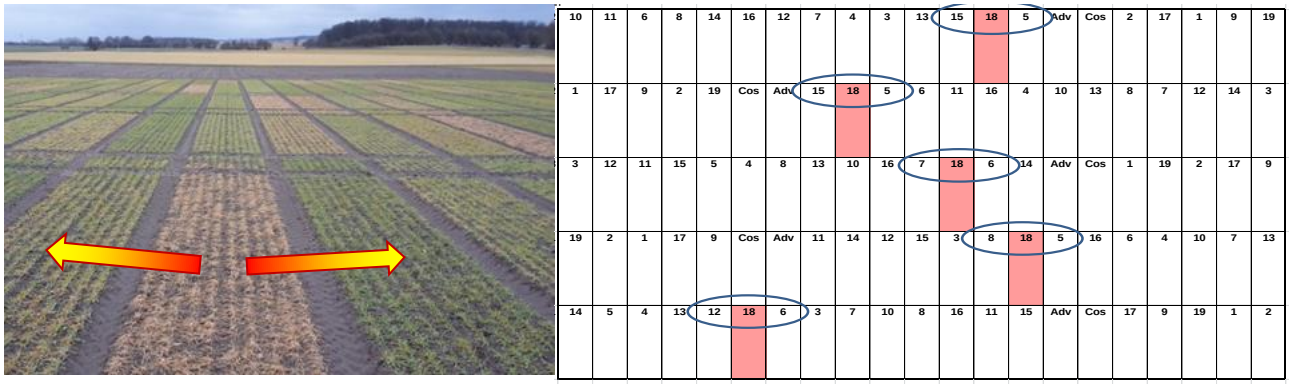


Abbildung 3: Nachbarschaftswirkung nach Auswinterung und Randomisationsmangel

Das Ziel der Nachbarschaftsbalancierung soll in Kombination mit den anderen Designzielen durch Einbindung eines Korrelation-Parameters innerhalb von Zeilen erzielt werden (Parameter im Verfahren: ‚spatial_corr_N'). Diese Korrelation ist nur für jeweils zwei unmittelbaren Zeilen-Nachbarn definiert, setzt sich also nicht fort.

Die Kombination eines Zeilen-Spalten-Planes mit der Nutzung der Kovarianz_NB ist der Schlüssel für eine stabile Minimierung wiederholter Nachbarschaften.

5.1.3 Zeilen- und Spalten - Effekte

Versuchsansteller wollen i.d.R. vermeiden, dass ein Püfglied mehrmals bzw. öfter als vermeidbar in der gleichen Zeile oder Spalte steht. Hintergrund ist die Erwartung, dass in gleicher Zeile oder Spalte Störeffekte korreliert sein können und nicht mehrmals auf das gleiche Prüfglied einwirken sollen. Diese Korreliertheit kann zum einen durch die unter 3.4.1 bereits beschriebene Bodenheterogenität entstehen. In diesem Fall wäre aber z.B. eine entfernte Position des gleichen Prüfgliedes in gleicher Zeile weniger schädlich, als eine nähere Diagonal-Nachbarschaft. Trotzdem werden Zeilen- und Spaltendopplungen als besonders riskant betrachtet. Hintergrund ist, dass nahezu alle technischen Arbeiten während des Versuches und auch in der Vorgeschichte genau im Parzellenraster mit den Zeilen oder mit den Spalten erfolgen. Bei technischen Arbeiten ist aber absolute Gleichheit über die Fläche nicht erreichbar. Traktorreifen hinterlassen Spuren, kein Düngerstreuer und keine Feldspritze hat eine absolut exakte Querverteilung, Drillscharre können in einer Fahrt verstopfen, nach dem Wenden aber wieder frei sein u.v.m. (Abb. 4).

Abbildung 4: Störende Zeileneffekte: alte Schlepperspur (links) und unzureichende Quer-Verteilung bei Düngung (oben-rechts); Spalteneffekte: zeitweilig verstopftes Drillschar (unten rechts), Fotos Michel

Die Folge sind distinkte Effekte für komplette Zeilen bzw. Spalten (Abb. 5). Je besser die Zeilen- und Spalten-Balancierung ist, desto weniger verzerrend wirken sich derartige Störeffekte aus und desto größer sind die Chancen für eine Effektbereinigung.

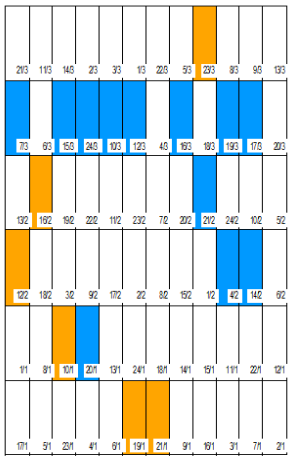


Abb. 5 zeigt, wie sich in diesem Falle (realer Versuch) Zeileneffekte auswirken können. die fünfte Zeile (von unten) hat auffällig negative (blaue) Residuen. Das bedeutet, die in Zeile 5 stehenden Parzellen sind benachteiligt. Da in diesem Versuchsdesign Zeilen keine vollständigen Wiederholungen darstellen, sondern eine Wiederholung aus zwei Zeilen besteht, ist die Hälfte der Prüfglieder durch den Zeileneffekt benachteiligt, die andere folglich indirekt bevorteilt.
In diesem Fall hat die Zeilenbalancierung immerhin bewirkt, dass kein Püfglied doppelt in der Zeile steht. Die Zeilenbalancierung sorgt aber zudem dafür, dass die Prüfglied-‚Verpaarungen' über alle Zeilen hinweg ausbalanciert sind, also dass alle Prüfgliedpaare annähernd gleich oft in Zeilen gemeinsam stehen. Diese Balanciertheit verbessert die Schätzbarkeit von Zeileneffekten (analog Spalteneffekten). Und damit erhöhen sich die Chancen, im Zuge der Auswertung durch ein adäquates Auswertungsmodell die verzerrenden Wirkungen auszugleichen (in diesem Fall quasi als Bonus für Parzellen in Zeile 5). Nur im Extremfall ist dann ein Ausschluss der Zeile begründet.

blau eingefärbt - Parzellen mit negativen Residuen
gelb eingefärbt - Parzellen mit positiven Residuen

Abbildung 5: Zeileneffekt - visualisiert in den Residuen im Lageplan.

Das Ziel der Zeilen-Spalten-Balancierung soll in Kombination mit den anderen Designzielen durch Einbindung einer Zeilen- und einer Spaltenvarianz erzielt werden.

5.2 Kombination der drei Prinzipien

In aller Regel ist zum Zeitpunkt der Versuchsplanung nicht vorhersehbar, ob, in welchem Maße und welche Art Störungen / Versuchsfehlern eintreten. Wie in 5.1 dargelegt, werden im Weiteren nur die Fehlerquellen für die Designwahl berücksichtigt, welche nicht nur auf Einzelparzellen wirken (der eigentliche Restfehler) sondern auf in unterschiedlicher Weise räumlich verbundene Strukturen. Anzunehmen ist dann, dass es sich immer wieder um Fehlermuster / Fehlerarten handelt, die in 5.1.1 bis 5.1.3 beschrieben wurden. Insofern erscheint es aus prophylaktischen Risikoerwägungen am aussichtsreichsten, alle drei Grundmuster im Versuchsdesign simultan zu berücksichtigen. Und zwar selbst dann, wenn man in dieser Kombination nicht völlig, aber annähernd den Grad z.B. der Zeilen-Spalten-Balancierung erreicht wie ein reiner Zeilen-Spalten-Plan, oder die Nachbarschaftsbalancierung, wie ein ausschließlich nachbarschaftsbalancierter Plan etc.

Abb. 6 soll zusammenfassend die gleichzeitige Berücksichtigung der drei Fehlerquellen im Versuchsdesign sowie die Berücksichtigung in der Auswertung erkennen lassen.

Die möglichen Kombinationen von Designvorgaben sind sehr vielfältig, sie gehen weit über die bisher verbreiteten ‚Anlagemethoden' (wie einfaktorielle Blockanlage, Lateinisches Rechteck, Alpha-Gitter ...) hinaus. Allerdings kann festgestellt werden, dass auch bisher in nahezu allen als Bock- oder Spaltanlage bezeichneten Anlagen zumindest eine manuelle Spaltenbalancierung vorgenommen wurde (Prüfglieddopplungen in gleicher Spalte wurden vermieden) - korrekterweise hätten bereits diese Pläne als Zeilen-Spalten-Pläne bezeichnet werden müssen.

Insofern ist es wenig sinnvoll, für jede denkbare Kombination von Designvorgaben weiterhin einen klassifizierenden Begriff für die ‚Anlagemethode' einzuführen. Vielmehr ergibt sich aus den Designvorgaben das adäquate Auswertungsmodell, wodurch de facto auch die ‚Anlagemethode' definiert ist. Das Auswertungsmodell kann in üblicher biometrisch-wissenschaftlicher Syntax oder durch die Syntax der verwendeten Statistiksoftware geschrieben werden. Im Falle der PIAF-Anwender wäre letzteres die Syntax der Formulierung in der PROC MIXED. Alle im Ergebnis dieser Arbeit empfohlenen Designs haben die Grundstruktur eines Zeilen-Spalten-Planes (s. 7.2.3). Alle Designerweiterungen (7.2.4 bis 7.4.3) führen zu Anlagen, die mit dem Sammelbegriff ‚Erweiterte Zeilen-Spalten-Plänen' bezeichnet werden können.

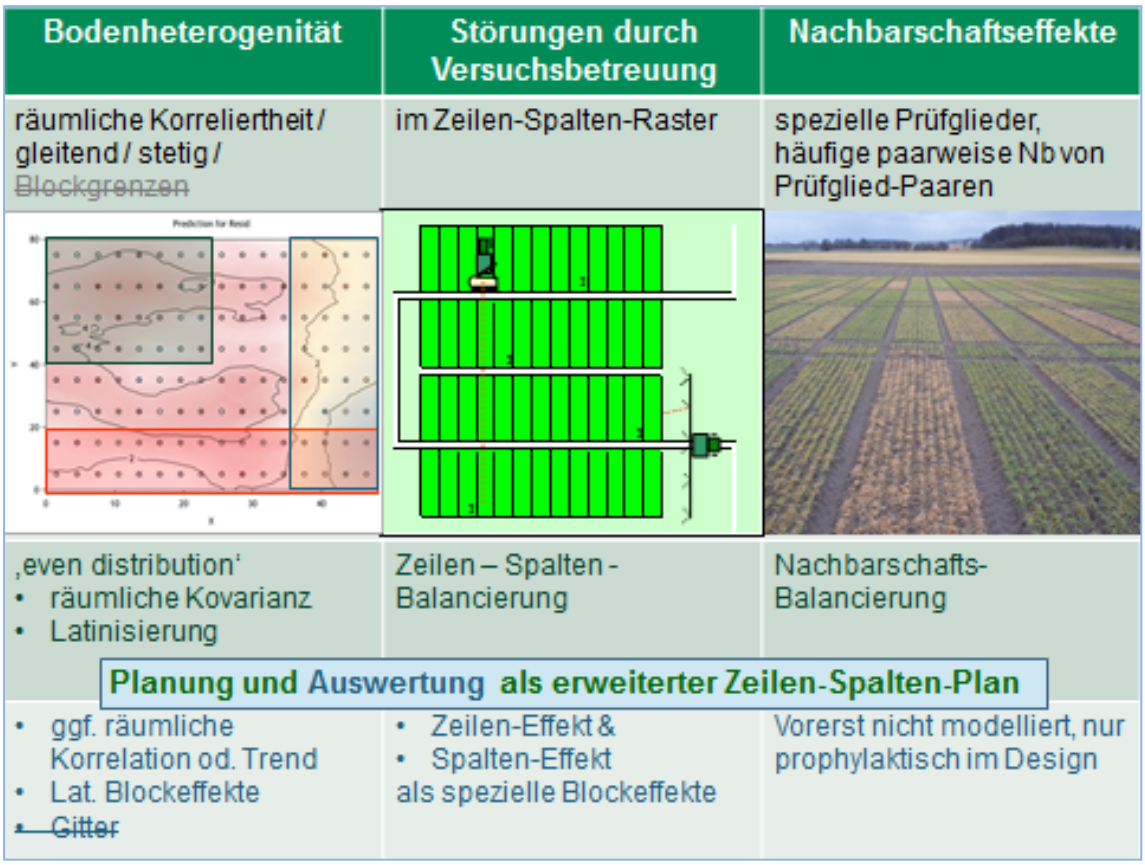


Abbildung 6: Drei typische Quellen des Versuchsfehlers und Berücksichtigung im Versuchsdesign und in der Auswertung

Tabelle 4: Typische Quellen des Versuchsfehlers und Ansätze zu ihrer Berücksichtigung

Art der Störung	**allgemeine Prinzipien zur Verringerung der Störwirkung auf Mittelwertdifferenzen**	**vorgeschlagene Ansätze im Versuchsdesign** **Erläuterungen**
Wirkungen technisch bedingter Ungleichmäßigkeiten im Vorfeld oder während der Versuchsbetreuung auf Zeilen bzw. Spalten: orthogonales Parzellenraster liegt in 0° bzw. 90°-Richtung zu fast allen Bewirtschaftungmaßnahmen	Zeilen- und Spalten-Balanzierung: ▪ Prüfglieder so selten wie möglich mehrfach in Zeile / Spalte ▪ alle Prüfgliedpaare so ausgeglichen häufig in gleichen Spalten /Zeilen	Zeilen und Spalten mit Varianzkomponenten im Planungsmodul Zeilen und Spalten erstrecken sich in Planung und Auswertung über Gesamtversuch (auch bei lösbaren Strukturen – im Gegensatz zum Ansatz bei resolvable row-column-Design in CycDesign®), da derartige Störungen nicht an Blockgrenzen springen
Bodenheterogenität → räumliche Korrelation der Residuen / Trends	Blockung: (aber Widerspruch stetiger Art der Störung gegenüber diskret definierten Blockgrenzen)	Latinisierung in 1-3 Richtungen, Im balancierten Fall entstehen vollständige Wiederholungen in 1-3 Strukturen. Dabei kann die Blockform ‚rechteckig' oder ‚krumm' sein.
	‚gleichmäßige' Verteilung der Wiederholungen eines Prüfglieds auf der Versuchsfläche	autokorrelative Varianzkomponente über die Gesamtfläche im Planungsmodul
Nachbarschaftseffekte zwischen Prüfgliedern	Nachbarschaftsbalancierung: Verringerung überproportional häufiger Nachbarschaft von Wdh. zweier Prüfglieder; in typischen Parzellen-Feldversuchen nur innerhalb von Zeilen relevant	autokorrelative Varianzkomponente innerhalb von Zeilen im Planungsmodul

Erste Analysen haben gezeigt, dass sich die drei Ziele (a) gleichmäßige Verteilung, (b) Zeilen- und Spalten-Balancierung und (c) Nachbarschafts-Balancierung im Sinne einer Gesamtoptimierung, von Ausnahme-Konstellationen abgeshen, meist gut verknüpfen lassen (Abb. 7). Die Einbußen je Einzelziel sind meist marginal. Am ehesten konkurrieren (a) und (b) mit (c), während (a) und (b) sogar in einer Weise gesteuert werden können, dass sich positive Synergieeffekte ergeben.

In Abschnitt 8 sind für kleine Versuche mögliche Probleme in der Spaltenbalance aufgezeigt, die durch Konkurrenz mit dem Ziel ‚Nachbarschaftsbalancierung' hervorgerufen werden. Es werden Hinweise gegeben, wie der Nutzer durch Änderung von Vorgaben das Problem ausräumen kann.

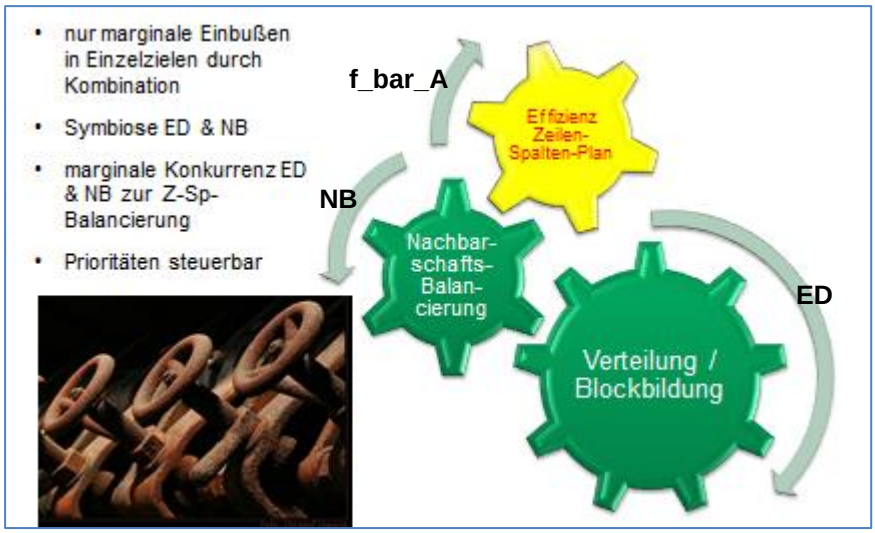


Abbildung 7: Interaktionen zwischen den Zielen ‚gleichmäßige Verteilung', ‚Zeilen-und Spalten-Balancierung' und ‚Nachbarschafts-Balancierung'

Bei vielen, insbesondere größeren Designs führt die Kombination von Zeilen und Spalten, zusätzlicher Latinisierung und Nutzung der Kovarianzen ED und NB zu besonders guten Ergebnissen in der Gesamtbewertung.

Werden Probleme mit ungewollt systematischer Spaltenzusammensetzung beobachtet, so liegt eine Konkurrenz zur Nachbarschaftsbalancierung vor - es wird auf Abschnitt 8 verwiesen.

Mehrfach-Latinisierung kann das Minimierungspotenzial bezgl. NB beschränken. Ein Best-Ergebnis kann dann oft trotzdem erzielt werden, wenn mehrere Run's ablaufen (O7).

Komplexe Design-Vorgaben sollten aufgrund des Vergleichs von Szenarien fallweise geprüft und entschieden werden, um nachteilige Modellüberfrachtungen zu vermeiden.

6 Nutzeroberfläche, Optionen und optionale Blocks

6.1 Eingangsbildschirme

Auswahl und Start des Verfahrens aus der Verfahrens-Bibliothek von PIAFStat erfolgen durch folgende Nutzer-Auswahlen (Abb. 8):

PIAFStat öffnen → „Daten“ → „Ausführen SAS“ → Verfahren „DESIGN“ → OK

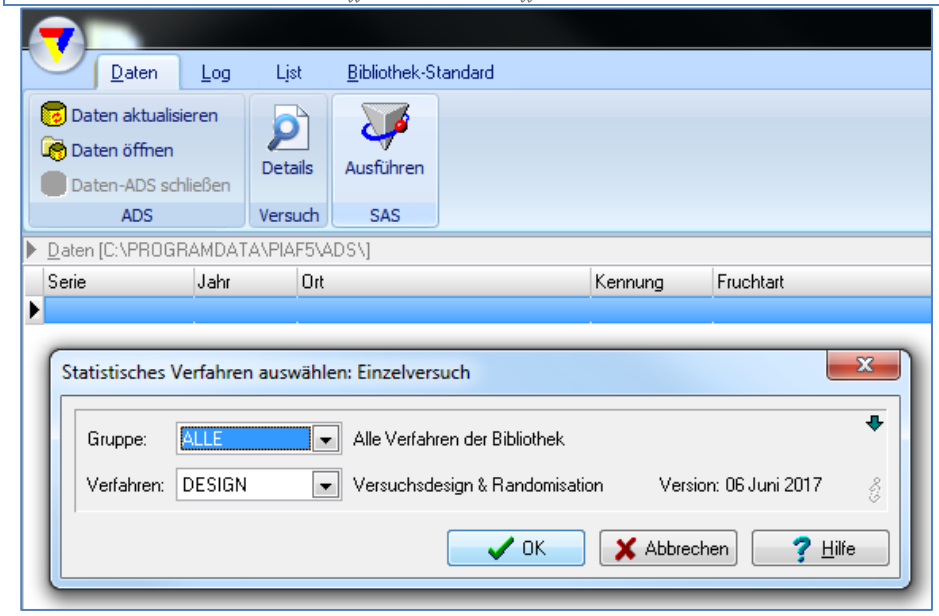


Abbildung 8: Auswahl des Verfahrens DESIGN aus der PIAFStat Verfahrensbibliothek

Das Verfahren DESIGN ist im Gegensatz zu den üblichen PIAFStat-Verfahren nicht an importierte Versuchsdaten gebunden, es muss also keine ADS geladen sein. In Abb. 2 zeigt sich dies darin, dass die blau unterlegte Zeile für geladene Versuchs- oder Seriendaten nicht gefüllt ist. Nach diesem Eingangsbildschirm ist auf die linke Registerkarte „Optionen“ zu klicken (Abb. 9).

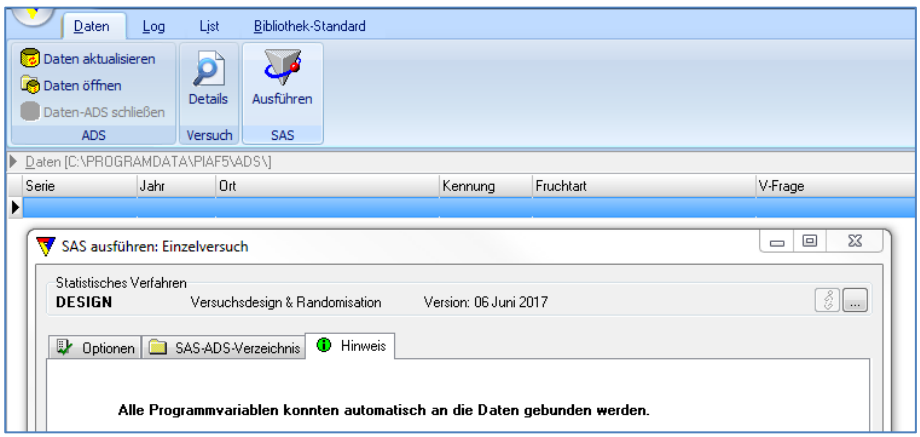


Abbildung 9: Eingangsbildschirm für das Verfahren DESIGN unter PIAFStat

6.2 Übersicht über alle Optionen

Abb. 10 zeigt das Gesamtangebot aller Optionen. Im dargestellten Screen wurden alle Ober- und Unteroptionen zur Veranschaulichung manuell geöffnet. Beim Öffnen sind die Unteroptionen aber zunächst geschlossen - dies kann im Verfahrens-Code individuell geändert werden.

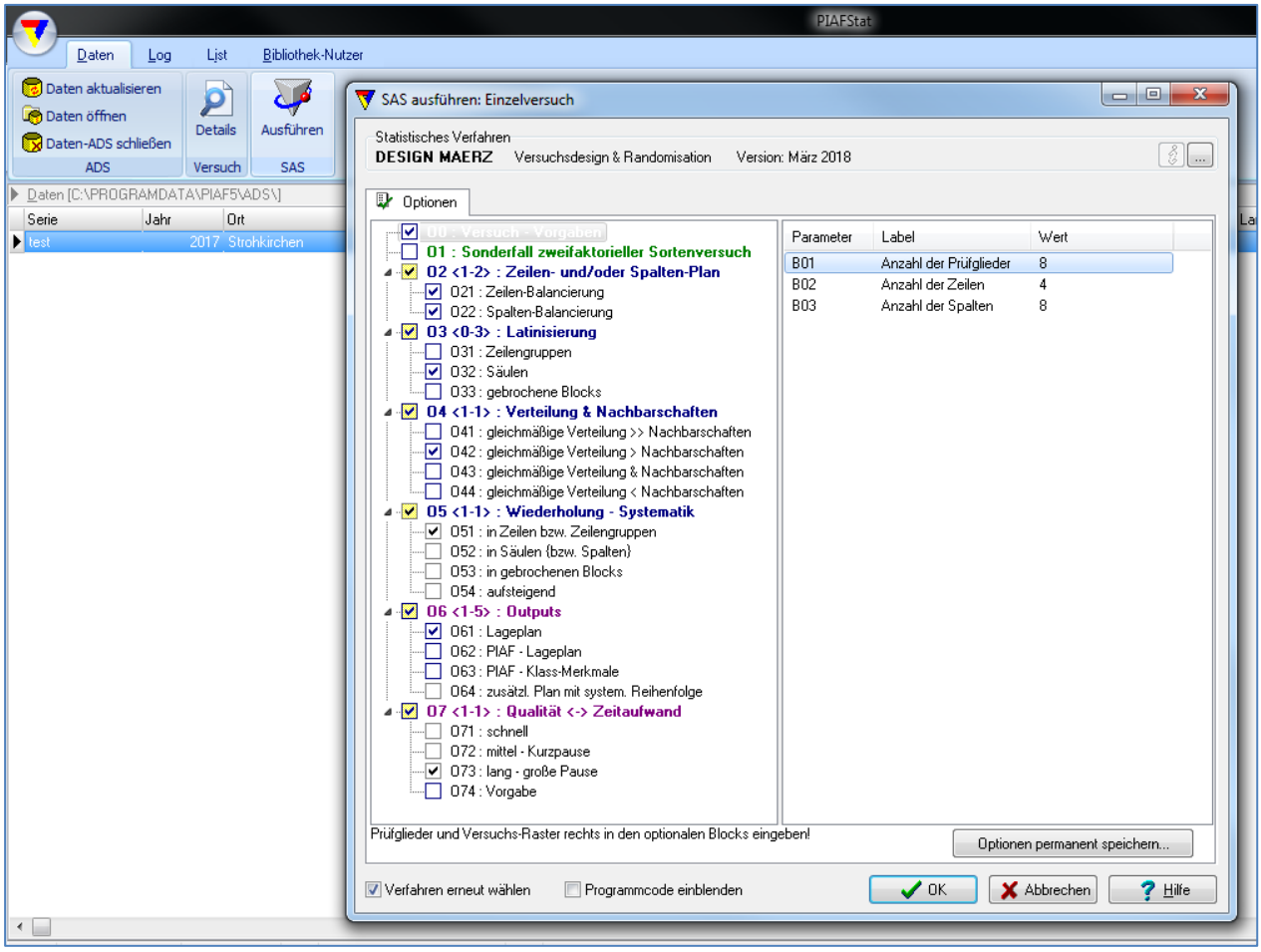


Abbildung 10: Nutzeroberfläche und Optionen des Verfahrens DESIGN (März 2018)

Optionale Blocks für zu editierende Nutzervorgaben zu den einzelnen Optionen sind in Abb. 10 bis auf O0 noch nicht zu ersehen. Diese lassen sich nur je Option ansprechen. Sie werden im Folgenden systematisch besprochen.

6.3 Option O0: Versuch - Vorgaben

Die Prüfglied(v)-, die Zeilen(k)- und die Spalten(s)- Zahl werden im optionalen Block der Option O0 vorgegeben (Abb. 3). Die Wiederholungszahl(r) ergibt sich aus: $\frac{k \times s}{v} = r$. Da r also durch die Nutzervorgaben für v, k, s eindeutig definiert ist und zudem nicht zwingend ganzzahlig sein muss, wurde r bewusst nicht als Eingabe-Parameter vorgesehen, sondern wird intern gebildet und ausbalanciert.

Die Nutzeroberfläche zur Eingabe von v, k, s ist auch in 6.2 und 7.2.1 als Screen dargestellt und beschrieben - in diesen Fällen für ganzzahlige r. Bei nicht ganzzahligem r erfolgt eine Ausbalancierung: die Prüfglieder unterscheiden sich in der Anzahl Wiederholungen dann um max. ‚1'. Ein Beispiel dafür ist in 7.3.7 vorgestellt.

6.4 Option O1: Sonderfall zweifaktorieller Sortenversuch

Nutzeroberfläche und Hintergründe sind in 7.4 vorgestellt.

6.5 Option O2: Zeilen- und/oder Spalten-Plan

Nutzeroberfläche und Hintergründe für Zeilen-Spalten-Pläne sind in 7.2.3 oder auch 7.2.10 vorgestellt.

Die Option O2 ‚Zeilen- und/oder Spaltenplan' ist auf permanent ‚ON' eingestellt, kann also nicht mehr ausgeschaltet werden. Hintergrund ist der DESIGN-Grundansatz erweiterter Zeilen-Spalten-Pläne. Wenn im Einzelfall doch Zeilen- oder Spaltenbalancierung bewusst ausgeschaltet werden soll, so kann dies durch Vorgebe der jeweiligen $Varianz = 0$ erfolgen (Optionen B21 und B22, s.u.). Achtung: dann nicht durch ‚Optionen permanent speichern' verstetigen, so dass in Folge evtl. unbemerkt bei allen weiteren Versuchen keine Balancierung mehr erfolgt.

Die Zeilen-Balancierung wird durch die Unteroption O21 gesteuert. Nutzeroberfläche und Hintergründe sind in 7.2.2 vorgestellt. In 7.4.1 wird auf Anforderungen in zweifaktoriellen Sortenversuchen hingewiesen. Die Spalten-Balancierung wird in Unteroption O22 gesteuert. Dieser Fall ist in 7.2.3 vorgestellt. In 7.2.1 bis 7.2.3 werden die Auswirkungen fehlender Zeilen- und/oder Spaltenbalancierung aufgezeigt und diskutiert.

Die Default-Einstellung 'auto' in den optionalen Blöcken B21 und B22 zur Festsetzung einer Zeilen- und einer Spaltenvarianz erbringt i.d.R. ein ausgewogenes Ergebnis für ED und NB.

Zeilen- und Spaltenbalancierung ist als Default für die meisten Situationen zu empfehlen.

Werden Probleme mit ungewollt systematischer Spaltenzusammensetzung beobachtet, liegt eine Konkurrenz zur Nachbarschaftsbalancierung vor - es wird auf Abschnitt 8 verwiesen. Eine Erhöhung der Spaltenvarianz ist aber auch dann nicht förderlich.

6.6 Option O3: Latinisierung

Latinisierung wird in der zugrundeliegenden Strategie vor allem angeboten, um außer Zeilen und Spalten ggf. zusätzliche, meist übergeordnete / größere Blockungseinheiten zu bilden. Mit dieser Strategie kann die Verteilung der Wiederholungen je Prüfglied über die Gesamtversuchsfläche erheblich verbessert werden (ED). Dies korrespondiert mit dem Anliegen derartiger Blockung, möglichst alle Prüfglieder in Blocks nah beieinander zuhalten und somit weitgehend ähnliche Bedingungen für alle Prüfglieder zu erzielen. Damit werden die Prüfgliedvergleiche gegenüber Flächen-Heterogenität robuster. Außerdem werden diese im Design vorgegebenen Blockungsfaktoren dann auch Teil des Auswertungsmodells (vor möglicher Modellreduktion mit EVA bzw. ZVA) und können ggf. zu weiter verbesserten Schätzwerten für Mittelwerte, Differenzen und Fehlermaße führen (Adjustierung, lsmeans).

Im Idealfall führen Blockbildungen durch Latinisierung zu rechteckigen und gleichzeitig vollständigen Blocks, wenn man für die Anzahl der Blockungsstufen die ‚auto'-Funktion nutzt (s. 7.2.5 und 7.3.1 bis 7.3.4). Es kann aber manuell jeweils auch eine individuell gewählte Anzahl Blockungsstufen vorgegeben werden (7.3.6). Auch für die jeweiligen Varianzen ist als Default ‚auto' vorgesehen, kann aber numerisch überschrieben werden.

Kann diese Gleichzeitigkeit von rechteckigen und vollständigen Blocks nicht erfüllt werden und soll trotzdem latinisiert werden, so muss dem Verfahren eine Vorgabe für die Blockform vorliegen, ob vollständige Besetzung Vorrang vor Rechteckigkeit haben soll (1 - krumme Blockform) oder ob Rechteckigkeit zwingend gefordert ist (2 - rechteckige Blockform). Auf Ebene der Ober-Option O3 kann im optionalen Block B3 ‚Blockform' einer dieser beiden Algorithmen vorgewählt werden. Nutzeroberfläche und Hintergründe sind in 7.2.6 und 7.2.7, aber z.B. auch in 7.3.5 vorgestellt. Die meisten Institutionen werden eine Variante für die Blockform als Default wählen und dauerhaft speichern.

Latinisierung in Zeilengruppen erfolgt in der Unteroption O31 (s. 7.3.1). Zeilengruppen bestehen aus durchgehenden Zeilen über die gesamte Versuchsbreite - bei ‚krummer' Blockform ist max. eine Zeile je Zeilengruppe zweigeteilt (s.a. 7.2.6 vs. 7.2.7).

Latinisierung in Säulen (Spaltengruppen) erfolgt in der Unteroption O32 (s. 7.3.2, 7.2.5). Säulen bestehen aus durchgehenden Spalten über die gesamte Versuchstiefe - bei ‚krummer' Blockform ist max. eine Spalte je Säule zweigeteilt (s.a. 7.2.6 vs. 7.2.7).

Latinisierung in Gebrochenen Blocks erfolgt in der Unteroption O33 (s. 7.3.3, 7.2.8, 7.2.9). Gebrochene Blocks haben weder durchgehende Zeilen noch Spalten, die Gesamtversuchsfläche wird also in beiden Dimensionen unterteilt. Bei Gebrochenen Blocks ist die auto-Funktion für die Anzahl Blocks derzeit nur für vier Wiederholungen optimal ausgerichtet. Es entstehen dann vier Quadranten als Gebrochene Blocks. Andere Konstellationen müssen per Hand vergeben werden (s.a. 7.2.11, 7.5). Gebrochene Blocks entstehen aus dem Kreuzprodukt von $Superzeilen \times Superspalten$. Mit diesem Konstruktionsprinzip lassen sich nicht nur (annähernd) vollständige Blocks erzeugen, sondern auch diverse andere Strukturen: so z.B. Teilblocks innerhalb von Zeilen (also Gitterstrukturen) oder bei Spaltanlagen die Einpassung des Kleinteilstücksfaktors in Großteilstücke des übergeordneten Faktors.

Die drei beschriebenen Formen der Latinisierung können einzeln erfolgen oder auch zur Zweifach oder Dreifach - Latinisierung kombiniert werden. Dreifach-Latinisierung wurde unter der Bezeichnung ‚Gerechte Pläne' oder ‚Gerechte Designs' veröffentlicht und angewandt (s.a. 7.3.4. bis 7.3.7).

Für die häufige Konstellation ‚wenige Zeilen - viele Spalten' wird die Nutzung der Latinisierung in Säulen - unabhängig von Balanciertheit - als Default empfohlen.

Je größer ein Versuch ist (Parzellenzahl), desto vorteilhafter wird häufig Zweifach- bis Dreifach-Latinisierung.

6.7 Option O4: Verteilung und Nachbarschaften

In dieser Ober-Option O4 kann auf die Verteilung der Wiederholungen der einzelnen Prüfglieder in der Fläche (ED, optionaler Block B41) und auf die Nachbarschaftsbalancierung (NB, optionaler Block B42) durch Vorgabe jeweils eines Kovarianz-Wertes Einfluss genommen werden.

Während Latinsierung die ED eher im Fernbereich - bei Betrachtung der Gesamtfläche - verbessert, nimmt der hier vorzugebende Parameter für ED eher nur auf den Nahbereich Einfluss. So wird z.B. tendenziell vermieden, dass zwei Wiederholungen eines Prüfgliedes diagonal benachbart sind, u.U. links / rechts bzw. unter / über einer Blockgrenze.

Prüfgliedpaare sollten so selten wie möglich miteinander in der Zeile direkt benachbart sein, damit Nachbarschaftseffekte nicht systematisch überhöht zu Verzerrungen führen. Da in den üblichen Feldversuchen der PIAF-Nutzer übereinander liegende Parzellen sich nicht unmittelbar berühren und die Parzellenbreiten gering sind, wurde auf die Berücksichtigung von Spalten-Nachbarschaft verzichtet.

Zusätzlich vermindert die Kovarianz für NB, dass einzelne Prüfgliedpaare aufgrund wiederholter NB korrelierte Residuen haben und damit besser vergleichbar sind, was aber automatisch zulasten anderen Prüfgliedvergleiche ginge. Die Prüfgliedvergleiche werden also harmonischer.

Als Default für die verwendeten Kovarianzen ist die ‚auto'-Funktion eingeschaltet, welche empirisch als besonders geeignet gefundene Werte nutzt. Die Szenarien in 7.2.1 bis 7.2.3 demonstrieren die Auswirkungen fehlender ED- und NB-Vorgaben (d.h. ‚auto' wurde durch ‚0' überschrieben). Ab 7.2.4 ist in allen Szenarien die ‚auto'-Funktion für die Kovarianzen eingeschaltet (Abb. 11).

Die Nutzung der räumlichen Parameter für ED und NB kann als Default für die meisten Situationen empfohlen werden.

Werden bei kleinen Versuchen ungewollt systematische Spaltenzusammensetzungen beobachtet, liegt eine Konkurrenz zur Nachbarschaftsbalancierung vor - es wird auf Abschnitt 8 verwiesen, ggf. muss im Einzelfall auf diese Kovarianzen verzichtet werden.

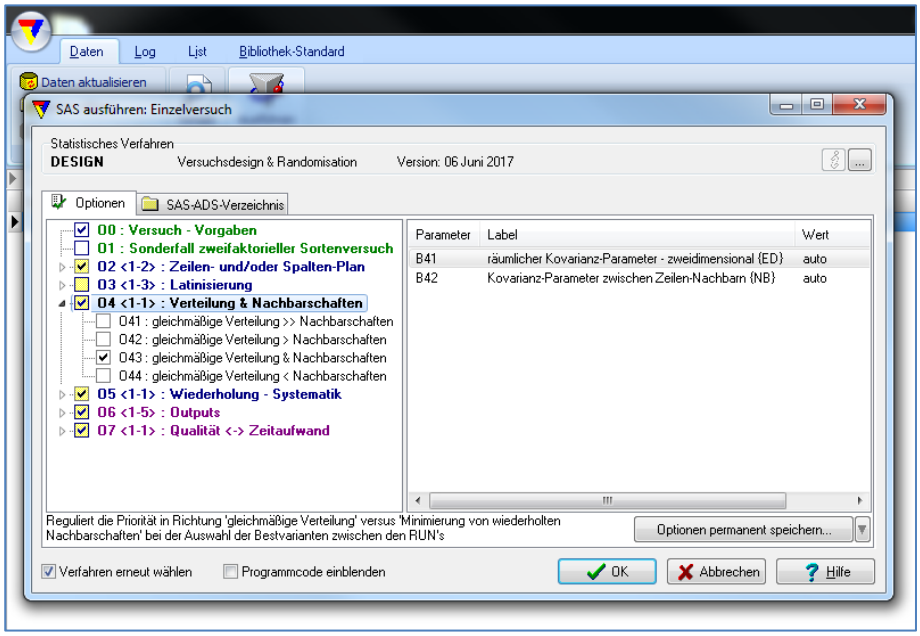


Abbildung 11: Optionaler Block für Kovarianzen für ED und NB (Stand 2017)

In den Unteroptionen O41 bis O44 kann vom Nutzer gesteuert werden, ob die Selektion einer Best-Randomisation aus mehreren Runs (s. 6.10; O7) eher in Richtung besserer ED-Werte oder besserer NB-Werte gewichtet. In 7.3.7 ist diese Abstufung bei gewählter Option O43 (gleichmäßige Verteilung & Nachbarschaften) zu ersehen und diskutiert.

In Richtung O41 könnte die Priorität gesteuert werden, wenn eher nicht mit Nachbarschafts–Effekten gerechtet wird - z.B. bei Maisparzellen aus vier Reihen, von denen nur die zwei inneren untersucht werden, oder bei Raps Versuchen in plot-in-plot - Saat. In Richtung O44 sollte es eher gehen, wenn schmale Ernteparzellen seitlich nicht abgeschirmt sind, auch keine Teilrandomisation in Gruppen erfolgt, aber z.B. in einem Sortenversuch deutliche Pflanzen-längen-Unterschiede erwartet werden. Als Default ist derzeit mit O43 eine ausgewogene (subjektiv empirisch bewertet) Gewichtung zwischen NB und ED gewählt.

Auch wenn bei kleinen Versuchen mit wenigen Zeilen keine Pläne mit einer Zellen-Spalten-Balance $f_{bar_A} > ca.\,0.3$ gefunden werden, sollte die Vorgabe eher in Richtung O41 gehen (weitere Hinweise in Abschn. 8).

6.8 Option O5: Wiederholung - Systematik

In PIAF angelegte Versuche zeichnen sich prinzipiell durch Wiederholungen je Prüfglied und Randomisation aus - dies sind Mindestanforderungen an biometrisch auswertbare Exakt–versuche und somit auch an das Verfahren DESIGN. Insofern lässt sich jede Parzelle in einem Versuch durch die Kombination einer eindeutigen Prüfgliedcodierung und einer eindeutigen Wiederholungsnummer ansprechen - alternative Ansprachen wären aber auch Feldnummern, Druschnummern u.v.m. Das PIAF-System erwartet grundsätzlich eine Parzellen-Codierung durch Prüfgliedkürzel in Kombination mit einer Wiederholungsnummer. Traditionell waren Wiederholungen gleichzeitig auch Blockungsstufen im Auswertungsmodell, z.B. in einfak–toriellen Blockanlagen, zweifaktoriellen Spaltanlagen etc.

In komplexen Designs, wie sie dieses Verfahren DESIGN unterstützt, gibt es dahingehend aber zwei Besonderheiten:

- Es können mehrere Blockungen simultan vorgegeben werden, die alle die Eigenschaft „vollständige Wiederholung“ erfüllen. Dann ist zu entscheiden, welche dieser Blockungs–faktoren außer zur Blockung zusätzlich zur Nummerierung der Wiederholungen heran-gezogen werden soll. So ist z.B. in 7.3.4 zu ersehen, dass sowohl Zeilengruppen als auch Säulen als auch Gebrochene Blocks als Wiederholungen dienen könnten - eine davon aber zu definieren ist. In 7.2.11 könnten Zeilen, Spalten oder Gebrochene Blocks Wieder-holungen bilden.
- Wenn die Wiederholungszahl im Versuch nicht über alle Prüfglieder konstant ist, kann es sein, dass keine einzige Blockungseinheit eine vollständige Wiederholung bildet - die Blockungseinheiten sind entweder über- oder unterbesetzt. Dann sind trotzdem alle Parzellen mit einer eindeutigen Wiederholungsnummer zu versehen und diese sind nach einem sinnvollen, übersichtlichen Prinzip in der Fläche zuzuordnen. In 7.3.7 ist ein der-artiger Fall gezeigt, wobei vier ‚überzählige‘ Parzellen eine fünfte Wiederholung bilden. Dagegen sieht das Ordnungsprinzip für die Blockbildung nur vier Gebrochene Blocks vor. In diesem Falle stellen die fünf Wiederholungen keine geeignete Blockungsgrundlage für die Auswertung dar, sondern sind reine Ordnungsnummern für PIAF (z.B. als Voraussetzung für Dateneingabe, -zuordnung und -ausgabe). Geeignete Blockungsstufen für die Auswer-tung sind dagegen die vier Gebrochenen Blocks, obgleich sie unbalanciert besetzt sind (jeweils ein Prüfglied doppelt im Block).

Häufig werden Zeilen oder Zeilengruppen die übliche, pragmatisch zu bevorzugende Grundlage der Wdh-Bildung sein. Dies trifft hinsichtlich Zeilen für alle Beispiel in 7.2 und 7.4 zu bzw. hinsichtlich Zeilengruppe für 7.3.1 und 7.3.4.

In 7.3.2 bilden weder Zeilen noch Zeilengruppen vollständige Wiederholungen, dafür aber Säulen. Hier sollte die Option O52 gewählt werden, um eine räumlich sinnvolle Anordnung der Wiederholungen von links nach rechts zu erzielen.

In 7.3.3 bilden ebenfalls weder Zeilen noch Zeilengruppen vollständige Wiederholungen, dafür aber Gebrochene Blocks. Hier sollte die Option O53 gewählt werden, um eine räumlich sinnvolle Anordnung der Wiederholungen in Quadranten zu erzielen.

Es gibt aber auch komplexe unbalancierte Konstellationen, bei denen keines dieser 3 Grund-ordnungsprinzipien befriedigt. Dann kann mit O54 vorgegeben werden, dass die Wieder-holungsnummerierung für jedes Prüfglied aufsteigend von unten links zeilenweise aufsteigend bis oben rechts erfolgen soll.

6.9 Option O6: Outputs

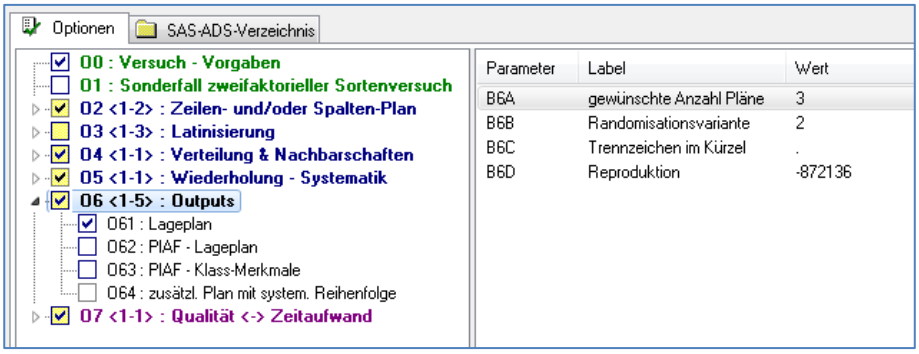


Abbildung 12: Parameter im Optionalen Block für Outputs (Stand 2017)

Auf Ebene der Option O6 können in optionalen Blocks folgende Vorgaben erfolgen (Abb. 12):

B6A <u>gewünschte Anzahl Pläne</u>

Häufig werden nicht nur Einzelversuche geplant, sondern mehrere Versuche einer Versuchsserie mit gleichen Designvorgaben. Trotz gleichem Grund-Design (Auswertungsmodell) sollen die Randomisationsergebnisse unterschiedlich sein, u.a. um systematische Störeffekte zu vermeiden. Letztere sind z.B. durch Nachbarschaftseffekte zwischen zwei Prüfgliedern denkbar, die bei identischer Verteilung in jedem Versuch die Verzerrung durch systematische Wiederkehr verstärken. Die unterschiedliche Verteilung ist ein Grundprinzip der Randomisationstheorie. Ist sie nicht gegeben, sind streng genommen alle Wahrscheinlichkeitsaussagen (in diesem Fall bezüglich der Serienauswertung), also auch GD etc. fragwürdig.

In Abb. 5 wurden z.B. 3 Pläne abgefordert, in Abb. 9 sind entsprechend 3 unterschiedliche Pläne in den Reitern Plan_1 bis Plan_3 ausgegeben. Auf die Ausgabe eines zusätzlichen Planes Plan_0 mit systematischer Prüfgliedreihenfolge in der ersten Wiederholung wird in 6.9.4 eingegangen.

B6B <u>Randomisationsvariante</u> (Abb. 13)

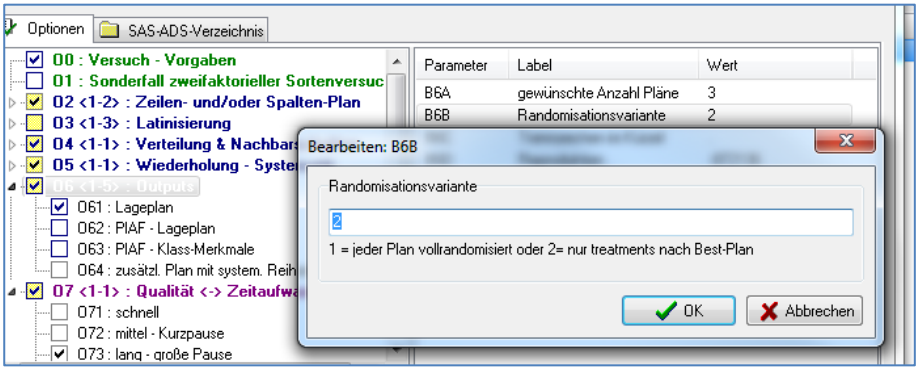


Abbildung 13: Wahl der Randomisationsvariante (Stand 2017)

ad Eingabe ‚1' = jeder Plan vollrandomisiert

> Eine uneingeschränkte Übereinstimmung mit der Randomisationstheorie erfordert, dass für jeden abgeforderten Plan ein komplett unabhängiger Run des Verfahrens stattfindet. Dies wäre daran zu erkennen, dass für jeden Plan (jede PlanID) und gleichzeitig für jede RunID ein individueller Startwert ausgewiesen wird - welcher mit der Bezeichnung ‚seed' je RunID in dem Excel-Output/Tabellenreiter ‚Maßzahlen' ausgewiesen wird (s.a. Abb. 21, 22). Zu favorisieren wäre diese Variante unbedingt bei wissenschaftlichen Vorhaben, bei welchen die Belastbarkeit von Wahrscheinlichkeitsaussagen ggf. sogar Vorrang vor der Minimierung des Schätzfehlers von Prüfglieddifferenzen hat. Bei drei abgeforderten

Plänen würde das Verfahren DESIGN also mindestens drei Runs ablaufen lassen (s. 3.10; O7). Dies kostet allerdings erstens u.U. mehr Zeit und zweitens würden sich die drei Pläne in den Güteparametern für das Design unterscheiden (Verschlechterung mit aufsteigender mit PlanID).

Allerdings kann es sein, dass die individuelle Erwartung an eine ‚guten' Lageplan in bestimmten Zielvorstellungen von den automatischen Vorgaben für die Sortierung der Runs nach Güte abweichen. Falls ein Anwender unterschiedliche Randomisationsergebnisse vergleichen und individuell einen Plan selektieren möchte, empfiehlt sich diese Vorgabe - besonders bei kleineren Versuchen.

So erzeugte Pläne erhalten den $DesignTyp = RUN$ (s.a. Abb. 21, 22).

ad Eingabe ‚2' = nur treatments nach Best-Plan

Bei dieser Strategie basieren alle Pläne auf der gleichen Grund-Randomisation von Plan_1=DesignID(1). Es wird lediglich ein Randomisationsschritt angefügt, in welchem die Prüfgliednummern zufällig neu verteilt werden (quasi getauscht), es entstehen Permutations-Pläne des Grundplanes ‚Plan_1'. Damit wären die typischen Nachteile fehlender Neuverteilung jedes Versuches behoben - gleichwohl ist die Randomisationstheorie nicht in absoluter Weise erfüllt. Diese Strategie bietet aber gerade für das angewandte Versuchswesen pragmatische Vorteile durch Zeit- oder/und Qualitätsgewinn. Zeitgewinn entsteht, wenn nur ein Run abläuft, aus dem mehrere Permutations-Pläne rekrutiert werden. Qualitätsgewinn entsteht, wenn in Option O7 trotzdem mehrere Runs gefordert werden, aber nur der ‚Best-Run' für alle Permutationen herangezogen wird. Dann haben alle Pläne zwar unterschiedliche Prüfgliedverteilungen, aber die gleiche, beste erzielte Design-Qualität.

So erzeugte abgeleitete Pläne erhalten den $DesignTyp = PER$ (s.a. Abb. 21, 22).

Bei besonders komplizierten Designvorgaben, welche u.U. auch im Best-Run unbefriedigende Prüfglied-Positionierungen zeigen, kann es allerdings nützlich sein, Variante 1 (jeder Plan vollrandomisiert) zu wählen (s.o.).

Das Prinzip der Auswahl eines oder mehrerer Best-Runs aus einer ggf. überzähligen Anzahl Runs wird u.a. in 6.9.4 und 6.10 diskutiert.

B6C Trennzeichen im Kürzel

Die Codierung jeder Parzelle erfolgt in allen Ausgaben durch die Kombination von Prüfgliednummern und Wiederholungsnummern. Im zweifaktoriellen Fall sind die Stufen für zwei Faktoren zu codieren. Zwischen den Nummern der Faktorenstufen untereinander und danach zur Wiederholungsnummer ist die Syntax eines Trennzeichens zu definieren. Die Wahl dieses Zeichens kann Nutzer spezifisch erfolgen und kann auch permanent gespeichert werden. So könnte in einem zweifaktoriellen Beispiel eine Parzelle, die in Faktor A die Stufennummer 6, im Faktor B die Stufennummer 2 und die Wiederholungsnummer 4 hat z.B. durch 6.2.4 oder 6/2/4 oder viele andere individuell gewählte Trennzeichen codiert werden (s.a. 3.9.1, Tabellenreiter ‚*Plan_1*').

B6D Reproduktion

In der Regel sollte hier eine völlig beliebige negative Zahl als Startwert für die Randomisation stehen. Dieser Eintrag wird in der Output-Excel-Datei im Tabellenreiter ‚*Protokoll*' festgehalten (s. Tab. 1; Zeile ‚Startwert'). Default ist eine Negativzahl - negative Zahlen werden vom System nicht als tatsächlicher Startwert verwendet, sondern führen dazu, dass für jeden Run ein zufälliger positiver Startwert erzeugt wird. Es ist dann praktisch ausgeschlossen, dass zwei Runs den gleichen Startwert erhalten - dadurch wird gewährleistet, dass die Runs unterschiedlich randomisieren. Somit wird

der Zufallsansatz einer Randomisation gesichert. Der dann je Run tatsächlich verwendete Startwert wird in der Excel-Ausgabe im Tabellenreiter ‚*Maßzahlen*' für jeden Run ausgewiesen.

Falls für sekundäre Analysen eine Randomisation identisch reproduziert werden soll, so müssen erstens alle Vorgaben identisch zu den Einträgen im Tabellenreiter ‚*Protokoll*' erfolgen und zweitens muss der positive Startwert aus Tabellenreiter ‚*Maßzahlen*' eingeführt werden (anstelle des negativen Default).

6.9.1 Unteroption O61: Lageplan

Diese Option liefert eine Excel-Datei zur nutzerfreundlichen Anschauung der Versuchspläne und zur Protokollierung und Archivierung der Designvorgaben. Folgende Einstellungen in optionalen Blocks können erfolgen (Abb. 14):

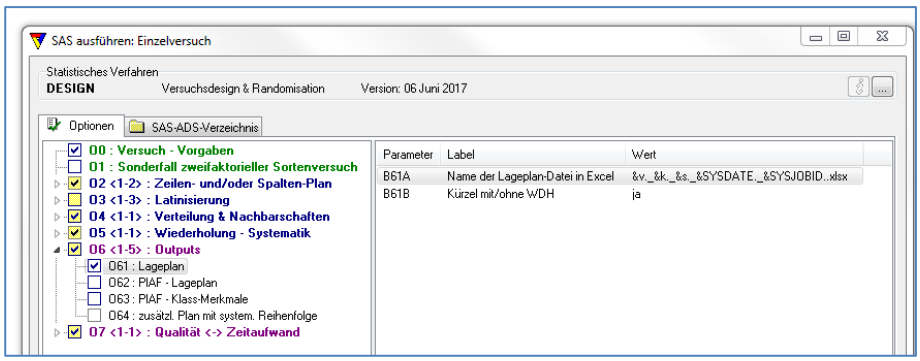


Abbildung 14: Optionale Vorgaben für die Output-Excel-Datei (Stand 2017)

B61A Name der Lageplandatei in Excel

In Abb. 7 ist zu ersehen, dass die Vorgaben für den Dateinamen im Default parametrisiert sein können, d.h., sie werden dann dynamisch aus den konkreten Designvorgaben gebildet. Es könnten aber auch nicht parametrisierte individuelle Dateinamen, andere Parametrisierungen und Reihenfolgen gebildet werden und als neuer Default permanent gespeichert werden.

Der Pfad ist automatisch das in der PIAFStat-Grundeinstellung eingestellte Verzeichnis für die Rücklese-Ergebnis-Schnittstelle (RES). Zur systematischen Ablage bietet es sich an, diese Dateien an anderem Speicherort in nutzerspezifisch strukturierten Verzeichnissen abzulegen.

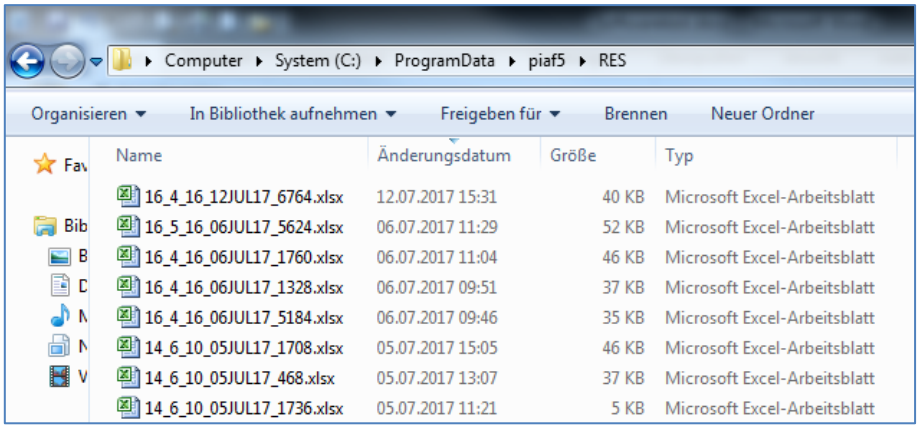


Abbildung 15: Verzeichnis und Dateinamen von Output-Excel-Datei

Das in Abb. 15 zeitlich jüngste erzeugte Design hatte nach vorgegebener Syntax 16 Prüfglieder (v), 4 Zeilen (k) und 16 Spalten (s), es wurde am 12.7.2017 erzeugt, und würde sich im Code ‚6764' von einem Design gleicher Vorgaben vom gleichen Tag unterscheiden. Die Bedeutung der Unterscheidbarkeit mittels dieses Codes wird in Abb. 8 bei den Designs vom 5.7.17 deutlich.

B61A Kürzel mit / ohne WDH

In den meisten Outputs in Abschnitt 7 enthalten die Parzellen-Kürzel an letzter Stelle die Wiederholungsnummer. Dies kann in diesem optionalen Block ausgeschaltet werden, was den Lageplan-Output ggf. übersichtlicher gestaltet. Insbesondere ist die Verteilung der Prüfglieder dann leichter überschaubar. Ein vollständiges Kürzel 6.2.4 würde dann auf 6.2 reduziert werden (s.a. in 3.9 B6C - Trennzeichen im Kürzel).

Tabellenreiter in der Output-Excel-Datei Aktualisieren

Die ausgegebene Excel-Datei enthält die Tabellenreiter laut Abb. 16 Diese Reiter werden nachfolgend besprochen.

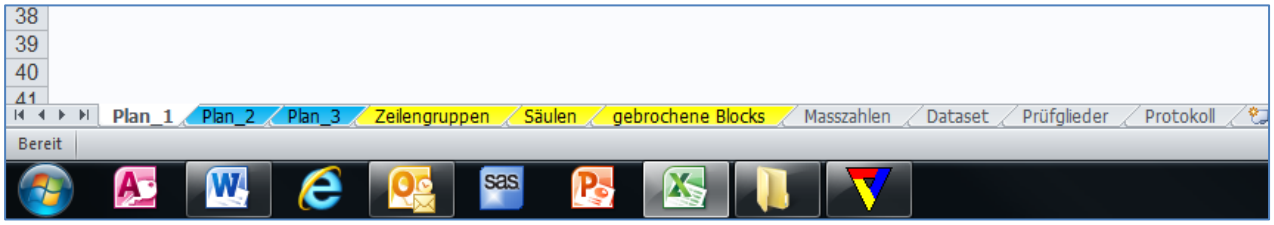


Abbildung 16: Tabellenreiter in der Output-Excel-Datei (Stand 2016)

Tabellenreiter ‚mögliche Fehler'

In Abb. 16 ist dieser Reiter nicht zu sehen - er wird nur gebildet, wenn das Verfahren Auffälligkeiten erkennt, die dem Nutzer mitgeteilt werden sollen. Dieser Reiter ist zunächst geöffnet - dadurch können mögliche Fehler schnell wahrgenommen werden. Es ist zu empfehlen, diesen Reiter nicht unbeachtet zu lassen, sondern zu prüfen, ob die Information Veranlassung für veränderte Designvorgeben gibt.

Tabellenreiter ‚Protokoll'

Entgegen der Reihenfolge der Reiter in Abb. 16 wird der Tabellenreiter ‚Protokoll' ab 2018 ganz links positioniert und ist zu Beginn geöffnet, sofern es keinen Reiter ‚mögliche Fehler' gibt.

Im Kopf des Protokolls sind - unabhängig von den anstehenden Designvorgaben - die an der Verfahrensentwicklung Beteiligten Einrichtungen und Berufskollegen aufgeführt.

Der eigentliche Kern ist dann die Dokumentation aller Nutzervorgaben und der letztlich verwendeten Parameter für die Prozedur. Ein komplettes Protokoll ist beispielhaft in Tab. 6 (7.2.1) dargestellt. In allen weiteren Beispielen ab 7.2.2 ist es hier nur auszugsweise mit den jeweils relevanten Zeilen ausgegeben. In der Rubrik ‚Zusatzinformationen' in der Zeile ‚zufällige Blockungsfaktoren' sind alle Blockungen aufgeführt, die bei der Einzelversuchsauswertung in PIAFStat (EVA, ZVA) angehakt werden sollten, damit sich Design und Auswertungsmodell decken.

Die Archivierung dieser Datei mit dem Protokoll gewährleistet, dass rückwirkend alle Vorgaben für ein Design abrufbar sind und dass ggf. Pläne mit identischem Design, ggf. sogar eine identische Randomisation (dann positiven Startwert aus Tabellenreiter *Maßzahlen* übernehmen), exakt reproduziert werden können.

Tabellenreiter ‚Plan_1', ggf. bis ‚Plan_x'

In den Tabellenreitern Plan_? wird die Prüfglied-Verteilung in der Fläche als Matrix im Zeilen-Spalten-Raster dargestellt. Die Darstellung erfolgt entsprechend der Nutzervorgaben für das Parzellen-Kürzel in den optionalen Blöcken B6C und B61A. Es wird mindestens ein Plan (Plan_1) dargestellt, insgesamt aber so viele, wie laut optionalem Block B6A angefordert sind, ggf. noch ein zusätzlicher Plan (Plan_0 mit systematischer Reihenfolge in der ersten Wiederholung).

Die Darstellungen finden sich in sämtlichen Beispielen des Abschnittes 7.

Zeilen- und Spaltenzuordnungen werden mit Unteroption O62 an PIAF übermittelt und stehen dann zur Lageplanerfassung in PIAF sowie als Blockungsfaktoren in der Auswertung zur Verfügung (s. Abschnitt 5).

Tabellenreiter ‚Zeilengruppen', ‚Säulen' und ‚Gebrochene Blocks'

Diese Tabellenreiter werden jeweils nur gebildet, wenn in der Option O3 Latinisierungen angewiesen wurden. Es werden dementsprechend 0 bis 3 Reiter für die Latinisierung gebildet. Die Ausgabe ist adäquat zur Darstellung in Plan_1, nur das anstelle der Parzellenkürzel die Blockzuordnungen je Parzelle ausgewiesen werden. In diesen Reitern kann die räumliche Lage der Blockstrukturen in der Fläche visuell gut nachvollzogen werden.

Darstellungen finden sich u.a. in 7.2.5 - 7.2.9, 7.2.11, in 7.3 und 7.4.2.

Blockzuordnungen für alle Parzellen werden mit Unteroption O63 an PIAF übermittelt und stehen dann als Blockungsfaktoren in der Auswertung zur Verfügung (s. Abschnitt 8).

Tabellenreiter ‚Maßzahlen'

Dieser Tabellenreiter liefert Informationen zur ‚Güte' des Designs hinsichtlich der angestrebten Zielkriterien (s.a. 7.1). Insbesondere können verschiedene Runs, Designs und Randomisationspläne (RunID, DesignID, PlanID) in diesen Kriterien verglichen werden. Die RunID zeigt nur die Reihenfolge der abgelaufenen Runs aufsteigend an. Diese Runs werden intern der Güte nach bewertet, absteigend sortiert und in der DesignID aufsteigend durchnummeriert. Lief mehr als ein Run ab und als Randomisationsvariante in B6B wurde $1 = Vollrandomisation$ für jeden Plan gefordert, dann wird $PlanID = DesignId$, wobei der ‚Bestplan oben steht und $PlanID = DesignID = 1$ ist.

Wurde dagegen $2 = permuationen\ des\ Bestplanes$ gewählt, so werden alle PlanID mit der $DesignID = 1$ ‚gefüttert'.

Tab. 5 definiert die Spalten-Inhalte im Tabellenreiter *Maßzahlen* (Stand 2017). Darin sind in den rechten Spalten Variablen definiert, welche die Grundlage eines Rankings für die Selektion eines oder mehrerer ‚Bestpläne' bilden. Das Konzept des ‚Bestplanes' und die Zusammenhänge zwischen RunID, DesignID, PlanID und DesignTyp sind in 6.9.4 / 6.10 veranschaulicht. Auszüge aus dem Output sind in allen Beispielen des Abschnittes 7 dargestellt und interpretiert. Diesen Darstellungen wurde allerdings eine im Original-Output und in Tab. 5 nicht enthaltene Spalte trt_1_ED zugefügt, die anzeigt, welches Prüfglied laut Tabellenreiter *Prüfglieder* die kritischste Verteilung aufzuweisen scheint. Nachvollziebarkeit und Diskussion vereinfachen sich aber durch diese manuell ergänzte Spalte.

Das Ranking der Runs / Designs / Pläne wird 2018 verbessert und wird schlägt sich in der Variablen DesignID und entsprechender Sortierung der Runs nieder.

Tabelle 5: Kurzdefinition der Variablen im Tabellenreiter ’Maßzahlen‘ (Stand 2017)

Variablen-Name	Bedeutung
RunID	Nummer des Runs
DesignID	Nummer des Ranking
zentrale Maßzahl für gleichmäßige Verteilung der Wiederholungen je Prüfglied (ED - *even distribution*)	
ED_Design	gewichtetes Abstandsmaß = 0.5 * MSTL_1 + 0.2 * NNLD_1 + 0.2 * MST_1 + 0.1 * NND_1
Maßzahlen für Nachbarschaftsbalancierung (NB)	
max_row_neighbor	max. Anzahl der paarweisen Nachbarschaft zweier Prüfglieder unter allen Prüfglied-Paaren (Nachbarschaft nur in Zeilen berücksichtigt) (z.B.: 5 und 7 sowie 3 und 9 sind je 5 mal direkt benachbart)
n_max_row_neighbor	Anzahl Prüfglied-Paare mit ‚max_row_neighbor‘ (im vorigen Beispiel kommt die max. Nachbarschaft 2 mal vor)
Maßzahlen für Güte der zeilen-Spalten-Balancierung	
f_bar_A	mittlerer Effizienzfaktor gegenüber einem Zeilen-Spalten-Plan mit festen Zeilen- und Spalteneffekten; s.a. *baseline efficiency* nach Piepho et al. 2016
zusätzliche Informationen für ED und kritischste Prüfglieder	
NND_1	NND - *nearest neighbour distance* = Mittelwert der euklidischen Distanzen der (w) *nearest neighbour plots* eines Prüfgliedes; Default: w=2; NND_1 - minimale NND unter allen Prüfgliedern
trt_1_NND_1	Nummer des Prüfglied mit NND_1
NNLD_1	wie NND, aber geometrisches Mittel der Distanzen (log-transformierte Distanzen & Rücktransformation); Ziel: große Abstände gedämpft wichten
trt_1_NNLD_1	Nummer des Prüfgliedes mit NNLD_1
MST_1	MST - *minimum spanning tree* (nach Moser 1992) = Mittelwert der Längen der Linien eines *minimium spanning tree*, welcher alle Wiederholungen eines Prüfgliedes verbindet; MST_1 - minimaler MST unter allen Prüfgliedern
trt_1_MST_1	Nummer des Prüfgliedes mit MST_1
MSTL_1	wie MST, aber geometrisches Mittel der Distanzen
trt_1_MSTL_1	Nummer des Prüfgliedes mit MSTL_1
mean_MSTL	Mittelwert der MSTL über alle Prüfgliedes
var_MSTL	Varianz der MSTL über alle Prüfglieder
seed	Anfangswert des Zufallsgenerators des konkreten Runs
Kalkulation für das Ranking / Rating der Runs	
Mean_n_max_row_neighbor	Mittelwert von n_max_row_neighbor je max_row_neighbor
Mean_ED_weighted	Mittelwert der gewichteten Abstandsmaße je max_row_neighbor
NB	relative Abweichung = -1 * (n_max_row_neighbor - Mean_n_max_row_neighbor) * 100 / Mean_n_max_row_neighbor
ED	relative Abweichung = (ED_weighted - Mean_ED_weighted) * 100 / Mean_ED_weighted
NB_weight	vorgegebenes Gewicht für die Nachbarschaften
Rating	Bewertung = NB_weight * NB + ED

Zur Bewertung von Designs sind diese Maßzahlen sinnvollerweise nur zum Vergleich von Anlagen mit gleicher v_r_k_s - Kombination geeignet (siehe Abkürzungsverzeichnis).

Tabellenreiter ‚Dataset'

In diesem Tabellenreiter sind alle Zuordnungen von Lageplan-Informationen IT- bzw. Datenbank gerecht abgelegt. Diese Struktur ist die Grundlage für die nutzerfreundlicheren Darstellungen in vorliegenden Outputs. Sie kann ebenso herangezogen werden, wenn individuelle Nutzer-Aufbereitungen gewünscht sind.

In der Spalte ‚DesignTyp' ist für jeden Plan ausgewiesen, ob die Randomisation original durch eine Run erzeigt wurde ($DesignTyp = RUN$), oder entsprechend der Vorgabe in B6B aus einer Permutation des Best-Runs ($DesignTyp = PER$) oder eine spezielle Permutation, die in der ersten Wiederholung eine systematische Reihenfolge der Prüfglied-Nummern erzeugt ($DesignTyp = SYS$) (s.a. Option O64 in 6.9.4).

Tabellenreiter ‚Prüfglieder'

Tabelle Prüfglieder für Design-Typ RUN

PlanID	RunID	DesignID	trt	NND	NNLD	MST	MSTL	ED	MAX_NO_ROWNEIGHBOR	no_of_reps
1	5	1	1	3,0	2,9	2,7	2,7	2,8	1	4
1	5	1	2	6,2	5,7	4,9	4,7	5,1	1	4
...										
3	9	3	16	5,5	4,4	4,3	3,5	4,1	1	4

Abbildung 17: Inhalt und Struktur der Tabelle ‚Prüfglieder' in der Output-Excel-Datei (Stand 2017)

In Tab. 17 werden je Versuchsplan (PlanID 1 bis x) Maßzahlen je Prüfglied ausgewiesen. Zu jedem Plan ist die Verknüpfung zur DesignID zu ersehen - ob es durchweg die DesignID(1) ist oder ob jeder Plan eine unterschiedliche (aufsteigende) DesignID hat, hängt von den Einstellungen in B6B ab. Letztlich ist also die gesamte Verknüpfungskette RunID → DesignID → PlanID nachvollziehbar. Dann folgen prüfgliedbezogene Maßzahlen für die ED (in der Version ab 2018 als ‚ED_Pruefglied' bezeichnet).

Die Maßzahlen werden nur für Pläne ausgewiesen, die nicht durch Permutationen erzeugt wurden (also nur DesignTyp='RUN', nicht ‚PER' oder ‚SYS' (s.o. bei Tabellenreiter ‚*Prüfglieder*').

Der Parameter ED wird aus den vier davorstehenden Parametern entsprechend der Formel für $ED\_weighted$ in Tab. 5 abgeleitet. Während ED_Design in Tab. 5 aber ein Design bezogener Parameter ist, bei welchem diese vier Ausgangswerte von unterschiedlichen Prüfgliedern stammen können (jeweils von dem Prüfglied mit dem je Maßzahl geringsten Wert), bezieht sich ED_Pruefglied auf genau ein Prüfglied. Insofern sind ED_Design und ED_Pruefglied nicht in jedem Fall absolut identisch.

Danach wird je Prüfglied ausgewiesen, wie viele wiederholte Nachbarschaften es maximal zu mindestens einem anderen Prüfglied gibt.

Und letztlich werden je Prüfglied die Anzahl Wiederholungen ausgewiesen. Dies ist besonders dann von Interesse, wenn die Wiederholungszahl uneinheitlich ist (s.a. 7.3.7).

Aus diesem Gesamtkomplex lässt sich ersehen, welche Prüfglieder ggf. sensibler auf Störgrößen reagieren könnten (geringstes ED, geringste Anzahl Wiederholungen, höhere Zahl wiederholter Nachbarschaften). Sollte es z.B. Prüfglieder geben, die nur als Zusatzprüfglied mit geringerem Interesse aufgenommen wurden o.ä., so kann der Nutzer ggf. in der Weise eingreifen, dass er Prüfgliednummern tauscht (quasi eine eigene Permutation erzeugt). Es ist dann dafür Sorge zu tragen, dass so ein Tausch an allen Positionen im Lageplan fehlerfrei bearbeitet wird und dass dies auch in der Datenbank (PIAF) sauber berücksichtigt wird.

6.9.2 Unteroption O62: PIAF - Lageplan

Durch diese Option werden Schnittstellen-Dateien gebildet, welche in PIAF mit der üblichen Nutzeroberfläche zum Einlesen von Lageplänen angesprochen werden können. Damit kann das Ergebnis von DESIGN hinsichtlich der Zuodrnung der Parzellen (Prüfglied/Wiederholung) zu Zeilen-Spalten-Kombinatonen effizient in PIAF eingelesen und genutzt werden.

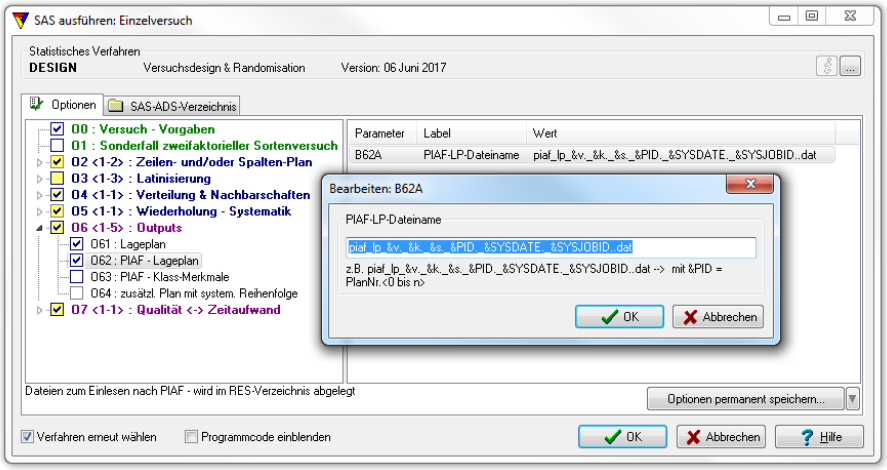


Abbildung 18: Erzeugung einer Schnittstellendatei zum Import des Lageplanes in PIAF

B62A PIAF-LP-Dateiname

In Abb. 18 ist zu ersehen, dass die Vorgaben für den Dateinamen im Default parametrisiert sind, d.h., sie werden dynamisch aus den konkreten Designvorgaben gebildet. Es könnten aber auch nicht parametrisierte individuelle Dateinamen, andere Parametrisierungen und Reihenfolgen gebildet werden und als neuer Default permanent gespeichert werden.

Der Pfad ist automatisch das in der PIAFStat-Grundeinstellung eingestellte RES-Verzeichnis. Zur systematischen Ablage bietet es sich an, diese Dateien an anderem Speicherort in nutzerspezifisch strukturierten Verzeichnissen abzulegen, am günstigsten im Kontext zu den Output-Excel-Dateien.

Die Anzahl der Schnittstellen-Dateien, die gebildet werden, hängt von der Anzahl der erzeugten Pläne ab. Das Verwenden der Makrovariable „&PID“ (= Plan-Nummer) im Namen der Schnittstellen-Datei ermöglicht das Abspeichern von allen erzeugten Plänen 1 bis n. Wird zudem über die Option O64 ein zusätzlicher Plan mit systematischer Reihenfolge gefordert (s. 6.9.4), erhält dieser Plan und die entsprechende Schnittstellen-Datei die Nummer 0.

Unter 9.1 sind die Schnittstellendatei sowie die Nutzeroberfläche zum Einlesen des Lageplans in PIAF beschrieben.

6.9.3 Unteroption O63: PIAF - Klass-Merkmale

Durch diese Option werden Schnittstellen-Dateien gebildet, welche in PIAF mit der üblichen Nutzeroberfläche zum Einlesen von Merkmalen angesprochen werden. Damit kann das Ergebnis von DESIGN hinsichtlich der durch Latinisierung gebildeten Blockungsfaktoren effizient in PIAF eingelesen werden. Damit stehen diese Blockungsfaktoren als sog. KLASS-Merkmale für die Einzelversuchsauswertung in PIAFStat (EVA, ZVA) zur Verfügung. Das ist Voraussetzung für die notwendige Übereinstimmung von Versuchsdesign und Auswertungsmodell (s. Abschnitt 10).

Zeilen und Spalten als Blockungsfaktoren im auswertungsmodell werden unabhängig davon automatisch durch den Lageplan-Import in PIAF gebildet und stehen in den Auswertungsverfahren EVA und ZVA automatisch zur Verfügung.

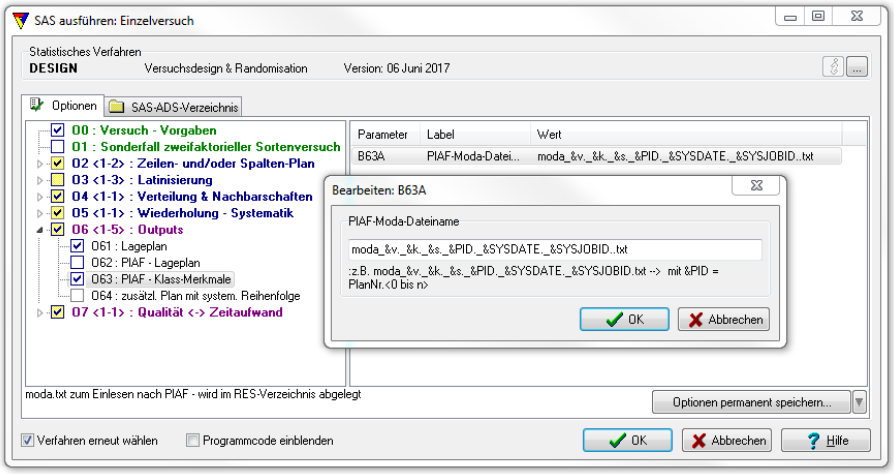


Abbildung 19: Erzeugung einer Schnittstellendatei zum Import der Klassifikationsmerkmale (Blockung durch Latinisierung) in PIAF

B63A PIAF-MODA-Dateiname

In Abb. 19 ist zu ersehen, dass die Vorgaben für den Dateinamen im Default parametrisiert sind, d.h., sie werden dynamisch aus den konkreten Designvorgaben gebildet. Es könnten aber auch nicht parametrisierte individuelle Dateinamen, andere Parametrisierungen und Reihenfolgen gebildet werden und als neuer Default permanent gespeichert werden.

Der Pfad ist automatisch das in der PIAFStat-Grundeinstellung eingestellte RES-Verzeichnis. Zur systematischen Ablage bietet es sich an, diese Dateien an anderem Speicherort in nutzerspezifisch strukturierten Verzeichnissen abzulegen, am günstigsten im Kontext zu den Output-Excel-Dateien.

Die Anzahl der Schnittstellen-Dateien, die gebildet werden, hängt von der Anzahl der erzeugten Pläne ab. Das Verwenden der Makrovariable „&PID“ (= Plan-Nummer) im Namen der Schnittstellen-Datei ermöglicht das Abspeichern der KLASS-Merkmale von allen erzeugten Plänen 1 bis n. Wird zudem über die Option O64 ein zusätzlicher Plan mit systematischer Reihenfolge gefordert (s. 6.9.4), erhält dieser Plan und die entsprechende Schnittstellen-Datei die Nummer 0.

Unter 9.2 sind die Schnittstellendatei sowie die Nutzeroberfläche zum Einlesen der Blockungsstrukturen (als Klassifikationsmerkmale) in PIAF beschrieben.

6.9.4 Unteroption O64: zusätzlicher Plan mit systematischer Reihenfolge

Grundsätzlich sollen alle Wiederholungen randomisiert sein. Eine systematische Anordnung in einer Wiederholung steht dazu im Widerspruch. Insbesondere könnten systematische Anordnungen zu Verzerrungen führen, wenn Versuchsserien angelegt und ausgewertet werden und also in jedem Versuch zumindest in der ersten Zeile die gleichen Nachbarschaften entstehen. Systematische Nachbarschaftseffekte führen dann zur übersteigerten Verzerrung von Prüfgliedvergleichen. Das Angebot dieser Option ist ein Zugeständnis an noch immer verbreitete Gepflogenheiten und Erwartungshaltungen, die aber ausdrücklich fachlich nicht unterstützt werden.

Als aus praktischer Sicht halbwegs unproblematisch könnte dieser Ansatz nur dann angesehen werden, wenn es keine Serie sondern nur einen Einzelversuch gibt bzw. wenn nur ein einziger Versuch der Serie diese systematische Anordnung in der ersten Wiederholung aufweist, sowohl

unter mehrortiger als auch mehrjähriger Sicht. Dann stellt dieser Plan in vorliegendem Algorithmus eine von vielen möglichen Permutationen einer Ausgangs-Randomisation dar. Es wird daher auch grundsätzlich nur ein einziger systematischer Plan vom Verfahren zugelassen. Dieser ist eine Permutation des ‚Bestplanes' $DesigID(1) = PlanID(1)$ (s.a. 6.10).

Zur Veranschaulichung wird die Konstellation $\mathrm{v\_r\_k\_s} = 7\_5\_5\_7$ (O0) mit Zeilen-Spalten-Balancierung (O2), ohne Latinisierung (O3) erzeugt. Es werden beide Randomisationsvarianten illustriert (O6, B6B). In beiden Situationen werden zwei ‚normale' Pläne (Plan_1 und Plan_2) sowie ein zusätzlicher Plan mit systematischer Anordnung der Prüfglieder in der ersten Wiederholung (Plan_0) angefordert.

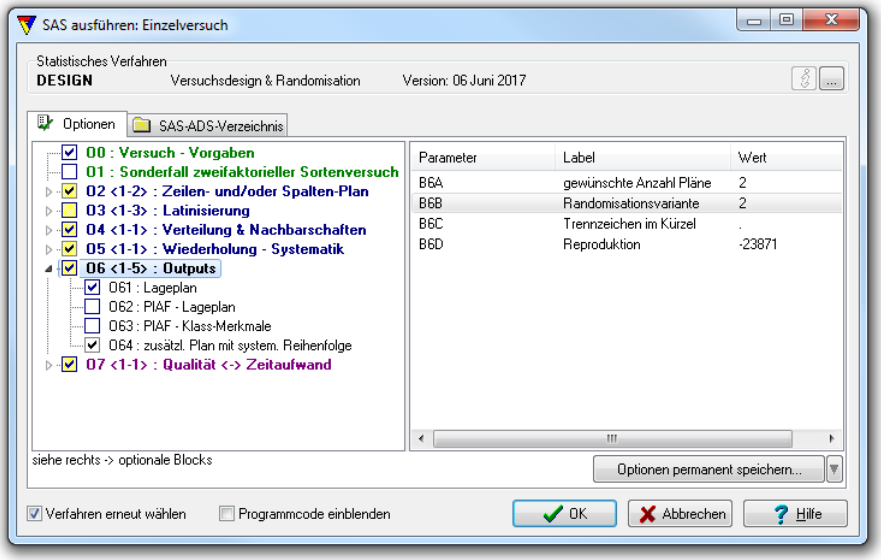


Abbildung 20: Anforderung eines zusätzlichen Planes mit systematischer Reihenfolge

Plan 0 mit Prüfglied-Kürzel

Z	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6	Sp7
5	2	3	1	6	4	7	5
4	5	4	2	7	3	1	6
3	7	1	4	3	6	5	2
2	3	5	6	2	7	4	1
1	1	2	3	4	5	6	7

RunID = 2
DesignID = 1
PlanID = 0
DesignTyp = SYS

Plan 1 mit Prüfglied-Kürzel

Z	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6	Sp7
5	7	4	1	3	5	6	2
4	2	5	7	6	4	1	3
3	6	1	5	4	3	2	7
2	4	2	3	7	6	5	1
1	1	7	4	5	2	3	6

RunID = 2
DesignID = 1
PlanID = 1
DesignTyp = RUN

Plan 2 mit Prüfglied-Kürzel

Z	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6	Sp7
5	1	5	7	3	6	4	2
4	2	6	1	4	5	7	3
3	4	7	6	5	3	2	1
2	5	2	3	1	4	6	7
1	7	1	5	6	2	3	4

RunID = 2
DesignID = 1
PlanID = 2
DesignTyp = PER

Abbildung 21: Randomisationsergebnisse mit / ohne systematische Reihenfolge bei $\mathbf{B6B} = \mathbf{2}$ (nachbearbeitet aus unterschiedlichen Tabellenreitern)

Plan 0 mit Prüfglied-Kürzel

Z	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6	Sp7
5	4	1	5	3	7	2	6
4	2	7	4	1	6	5	3
3	6	3	2	5	4	7	1
2	7	5	1	6	3	4	2
1	1	2	3	4	5	6	7

RunID = 4
DesignID = 1
PlanID = 0
DesignTyp = SYS

Plan 1 mit Prüfglied-Kürzel

Z	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6	Sp7
5	4	3	5	6	2	7	1
4	7	2	4	3	1	5	6
3	1	6	7	5	4	2	3
2	2	5	3	1	6	4	7
1	3	7	6	4	5	1	2

RunID = 4
DesignID = 1
PlanID = 1
DesignTyp = RUN

Plan 2 mit Prüfglied-Kürzel

Z	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6	Sp7
5	2	3	1	7	5	4	6
4	1	5	4	3	6	2	7
3	3	7	2	1	4	6	5
2	5	2	3	6	7	1	4
1	6	1	7	4	3	5	2

RunID = 1
DesignID = 2
PlanID = 2
DesignTyp = RUN

Abbildung 22: Randomisationsergebnisse mit / ohne systematische Reihenfolge bei $\mathbf{B6B} = \mathbf{1}$ (nachbearbeitet aus unterschiedlichen Tabellenreitern)

Enthält die gewählte Blockstruktur für die erste Wiederholung nicht jedes Prüfglied genau einmal, wird ein systematischer Plan abgelehnt.

Auch wenn man den systematischen Plan nicht verwenden möchte, ist die standardmäßige Anforderung dieses Planes eine gute Kontrollmöglichkeit, ob Orthogonalität erzielt wurde.

6.10 Option O7: Qualität ←→ Zeitaufwand

Unabhängig von der Komplexität der Designvorgaben und damit von gezielten Einschränkungen einer völlig freien Randomisation findet bei jedem Ablauf des Verfahrens (Run) ein Randomisationsprozess statt. Je zielführender die Designvorgaben vom Nutzer getroffen wurden, desto stabiler werden die Zielkriterien optimiert und desto stabiler sind die erzielten Werte für diese Kriterien, wenn man das Verfahren mehrmals ablaufen lässt (mehrere Runs).

Gleichwohl erbringen verschiedene Runs i.d.R. nicht völlig identische Masszahlen. Insofern kann es lohnend sein, mehrere Runs für einen Plan ablaufen zu lassen, sie der Güte nach zu bewerten und den ‚Bestplan' / die ‚Bestpläne' zu nutzen. Ist im Optionalen Block B6B ‚1' gewählt, so werden für x geforderte Pläne die x besten Runs von $Anzahl_{Runs} \geq x$ genutzt. Ist im Optionalen Block B6B ‚2' gewählt, so wird für x geforderte Pläne nur der eine Best-Run genutzt und die Pläne mit $PlanID$>1 werden Permutationen aus diesem Bestplan.

Natürlich kostet jeder zusätzliche Run zusätzliche Laufzeit des Verfahrens - Qualitätsgewinn wird quasi mit Zeit erkauft. Hier sollte also eine sinnvolle Abwägung getroffen werden. Für diese Abwägung sollte der Nutzer folgende Zusammenhänge und Zielkonflikte kennen:

- Je größer der Versuch (Parzellenzahl), desto größer ist die Variation in den Maßzahlen über die Runs - insofern wäre eine steigende Anzahl Runs mit steigendem Versuchsumfang lohnend.
- Mit steigendem Versuchsumfang steigt allerdings im Gegenzug der Zeitbedarf je Run stark progressiv, was leider einer sehr großen Anzahl Runs gerade bei sehr großen Versuchen entgegensteht.
- Erwartungsgemäß gilt auch hier das Gesetz vom abnehmenden Zuwachs - die Qualität der Bestpläne steigt zwar mit steigender Anzahl Runs, aber nicht linear, sondern degressiv gegen eine Asymptote.

Bei großen Versuchen kann es sinnvoll sein, zielgerichtet lange Pausen oder sogar die Rechenzeit nach Feierabend zu nutzen bzw. das Verfahren im Hintergrund laufen zu lassen und andere Aufgaben zu erledigen. So werden Qualitätsreserven genutzt, ohne am Arbeitsplatz auf das Ende des Programmablaufes warten zu müssen.

Unteroptionen O71 - O73: schnell bis lang

In den Unteroptionen sind zunächst drei vordefinierte Entscheidungen für das Verhältnis Qualität ←→Zeitaufwand angeboten:

	Bezeichnung	wenn $B6B$ = ‚2'	wenn $B6B$ = ‚1'
O71:	schnelle	$y = 1$	$y \geq 1 \cup y \geq x$
O72:	mittel - Kurzpause	$y = 4$	$y \geq 4 \cup y \geq x$
O73:	lang - große Pause	$y = 10$	$y \geq 10 \cup y \geq x$

wobei gilt: $\boldsymbol{y}$ = Anzahl Runs *und* $\boldsymbol{x}$ = Anzahl geforderte Pläne

Nur einen Run auszuführen (O71) und nur einen Plan abzufordern kann zur Beschleunigung bei Testläufen oder Demonstrationen genutzt werden.

Für reale Versuchsplanungen sollten ca. vier Runs (O72) das Minimum sein.

Mit weiterer Verbesserung der Auswahl von Best-Runs wird die Vorteilswirkung vieler Runs weiter steigen.

Bei sehr vielen Runs steigt die Chance auf das Finden eines Planes, der die Zielkriterien besonders gut und simultan erzielt. Die in 2018 und 2019 anstehenden Verfeinerungen der Bewertung von Plänen sind Anlass, die Anzahl Runs künftig noch stärker als bisher zu erhöhen.

Unteroption O74: Vorgabe

Die Option O74 wurde angeboten, um dem Nutzer eine individuelle Vorgabe für die Anzahl Runs zu ermöglichen, insbesondere um eine deutlich größere Anzahl Mindest-Runs je Plan zu definieren. Als Default ist zunächst im optionalen Block der Wert ‚25' eingegeben, diese Zahl lässt sich aber beliebig anders wählen.

Sehr große Werte könnten bei besonders großen Versuchen lohnend werden - vorausgesetzt, man kann aufgrund sehr langer Pausen, Beratungen, nach Feierabend etc. den Rechner lange laufen lassen - oder das Verfahren läuft auf sehr leistungsfähigen Rechnern im Hintergrund neben anderen simpleren Anwendungen.

Anzahl Iterationen (Iter)

Im Optionalen Block zur Ober-Option O7 kann die Anzahl Iterationen je Run editiert werden. Mit der ‚auto'-Funktion werden aus empirischen Untersuchen als geeignet gefundene Werte eingesetzt: je mehr Parzellen ein Versuch hat, desto höher ist die Anzahl Iterationen; je weniger Runs je angefordertes vollrandomisiertes Design angewiesen wurden, desto größer die Anzahl Iterationen je Run.

Derzeit gilt:

$$Iter = Zeilenzahl \times Spaltenzahl \times x \;;\quad mit\ x = 8$$

Eine Erhöhung der Anzahl Iterationen über die Größenordnung $(\approx 8 \times Parzellnzahl)$ führt nur selten zur Verbesserung der Zielkriterien. Bei Abwägung von Zeit und Qualität lohnt es sich dann eher, die Anzahl Runs zu erhöhen, als die Anzahl der Iterationen je Run zu erhöhen.

6.11 Optionen permanent speichern

Einstellungen in Optionen, Unteroptionen und in deren optionalen Blocks können temporär erfolgen, wobei bei jedem erneuten Verfahrens-Aufruf wieder die Grundeinstellungen (Defaults) angeboten werden. Oft hat ein Nutzer bzw. eine Institution aber für Designvorgaben bestimmte spezifische Präferenzen. Dann lohnt es sich, dies zu verstetigen und zum eigenen Default zu machen. Dazu dient die PIAFStat-Grundfunktionalität, die in allen Verfahren verfügbar ist: ‚Optionen permanent speichern' (Abb. 23).

Auch wenn für bestimmte Grundkonstellationen, insbesondere in den Versuchsvorgaben (O0 und O1), nacheinander verschiedene Szenarien ablaufen sollen, um z.B. im Nachherein einen Strategievergleich und eine Auswahl vorzunehmen, kann diese Funktion nützlich und vereinfachend sein. Man gewährleistet dadurch weitgehend fehlerfrei, dass die zu vergleichenden Szenarien in den Grundeinstellungehn identisch sind und sich nur in bewusster Variation einzelner Designvorgaben unterscheiden. Ggf. können im Nachgang wieder die Ausgangsdefaults eingestellt werden.

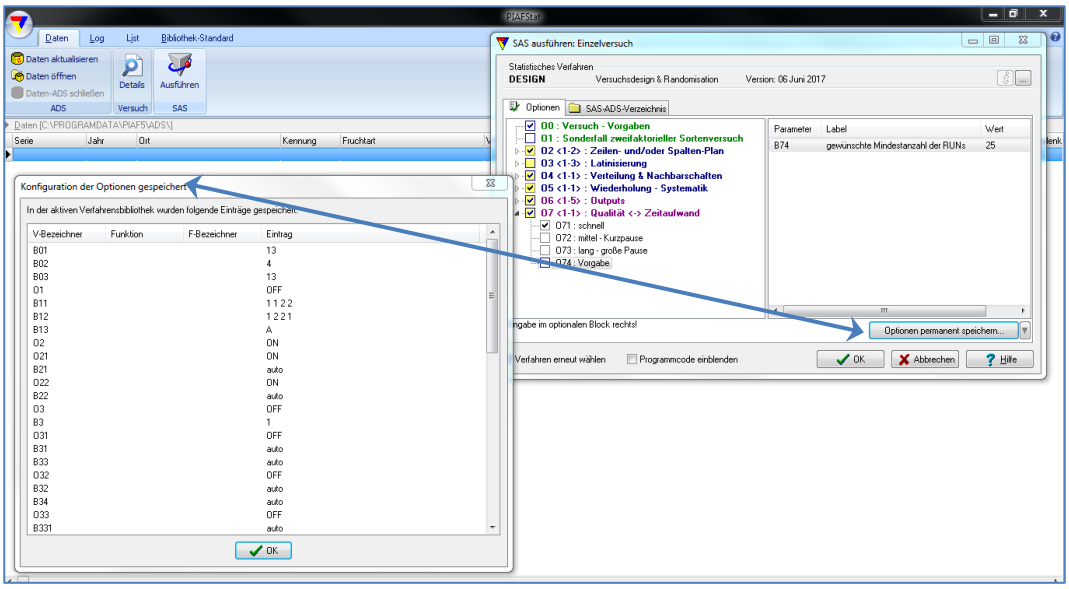


Abbildung 23: PIAFStat-Funktionalität ‚Optionen permanent speichern'

In diesem Zusammenhang ist im Rahmen der Pflege und Weiterentwicklung von PIAF in 2018 vorgesehen, mehrere unterschieliche Options-Konstellationen zu speichern und abrufen zu können (z.B. ‚Raps-LSV', ‚zweifaktorielle Getreide-LSV'; ‚Versuche mit 5 Zeilen' u.v.m).

7 Versuchsdesigns mit unterschiedlich komplexe Vorgaben

7.1 Zielkriterien und deren Maßzahlen

Wichtigste Zielkriterien sind Zeilen-Spalten-Balancierung, gleichmäßige Verteilung und Nachbarschaftsbalancierung. Je Randomisationsplan werden für diese Zielkriterien folgende Maßzahlen zur Bewertung eines Randomisationsergebnisses ausgewiesen:

- für Zeilen- und Spaltenbalancierung: **f_bar_A**

  mittlere relative Effizienz gegenüber einem reinen Zeilen-Spalten-Plan mit festen Zeilen- und Spalteneffekten

  Ziel: Maximierung (Obergrenze im Lateinischen Quadrat ist ‚1')

- für gleichmäßige Verteilung: **ED_Design** (in älteren Versionen **ED_weighted** bezeichnet) bzw. **ED_Pruefglied** (in älteren Versionen **ED bezeichnet**)
  gewichtete Abstandsmaße für die gleichmäßige Verteilung der Wiederholungen je Prüfglied (ED - even distribution)

  Ziel: Maximierung
  (Szenarien nur bei gleichen Vorgaben in O0 und O1 vergleichbar; je mehr Prüfglieder, desto größer wird das erreichbare Maximum für ED)

- Nachbarschaftsbalancierung: (a) **max_row_neighbor** und (b) **n_max_row_neighbor**

  (a) max. Anzahl der paarweisen Nachbarschaft zweier Prüfglieder unter allen Prüfglied-Paaren (Nachbarschaft nur in Zeilen berücksichtigt)

  Ziel: Minimierung (Untergrenze ist ‚1')
  (dieser Parameter hat übergeordnete Bedeutung, quasi als Quantensprung)

  (b) Anzahl Prüfglied-Paare mit ‚max_row_neighbor'

  Ziel: Minimierung
  (nur innerhalb einer (a)-Gruppe vergleichbar)

Die Definition der Maßzahlen und ihrer Herleitungen erfolgten in den Erläuterungen zu den Tabellenreitern *Maßzahlen* und *Prüfglieder*.

Zur Bewertung von Designs sind diese Maßzahlen sinnvollerweise nur zum Vergleich von Anlagen mit gleicher v_r_k_s - Kombination geeignet (siehe Abkürzungsverzeichnis).

Nachfolgend werden alle durch die Nutzer steuerbaren Designvorgaben und jeweils beispielhaft erzielte Outputs gezeigt und vergleichend diskutiert. Die Komplexität der Designvorgaben nimmt dabei über die Abschnitte zu.

7.2 Einfaktorielle Versuche in der Konstellation: Zeile = vollständige Wiederholung

7.2.1 Uneingeschränkte Randomisation

Für die Konstellation v_r_k_s = 12_4_4_12 (siehe Abkürzungsverzeichnis) wird eine uneingeschränkte Randomisation erzeugt. In Optionen O2 sind die Varianzen auf null gesetzt, Latinisierung in O3 ist ausgeschaltet, beide räumlichen Kovarianzen in O4 sind auf ‚0' gesetzt (Abb. 24, Tab. 6).

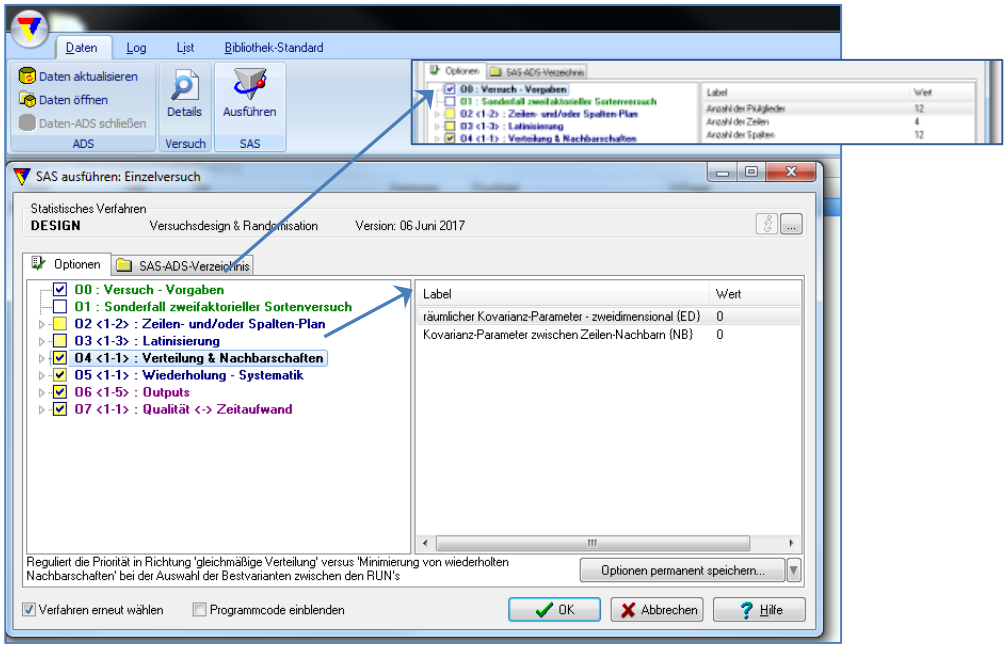


Abbildung 24: Design-Vorgaben für uneingeschränkte Randomisation

Z	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6	Sp7	Sp8	Sp9	Sp10	Sp11	Sp12
4	12	6	6	10	2	11	9	10	8	6	10	4
3	5	9	5	11	2	8	8	1	4	12	1	3
2	9	12	12	2	5	11	1	6	10	8	7	4
1	7	3	9	7	1	5	3	7	3	2	11	4

ED_weighted	max_row_neighbor	n_max_row_neighbor	f_bar_A	trt_1_ED
1,83447	3	3	,58328	8

Die Spalte ‚trt_1_ED' liefert in der ersten Zeile dasjenige Prüfglied, welches im Tabellenreiter ‚Prüfglied' den geringsten Wert in der Kennzahl ‚ED' erbringt. Der Parameter ‚trt_1_ED' wird im Tabellenreiter ‚Maßzahlen so nicht bereitgestellt, er ist hier nachträglich eingefügt.

Abbildung 25: Randomisation und Maßzahlen bei uneingeschränkter Randomisation

Aufgrund fehlender Vorgaben für Zeilen- und Spalteneffekte ist deren Balancierung erwartungsgemäß schlecht (Abb. 25 z.B. Prüfglieder 4 und 8 sowie f_bar_A). Ebenso ist auch die räumliche Verteilung und Nachbarschaftsbalancierung schlecht.

Für das praktische Versuchswesen kann die uneingeschränkte Randomisation, also der Verzicht auf weitere Designvorgaben, nicht empfohlen werden.

Tabelle 6: Komplettes Protokoll, uneingeschränkte Randomisation

Bereich	Bezeichnung	Vorgabe
Verfahren:	Versuchsdesign & Randomisation Version: 06 Juni 2017	
biometr.Grundlagen & Veröffentlichung:	Piepho (Uni Hohenheim) und Michel (LFA Gülzow)	
Konzeption & Koordinierung:	Michel (LFA Gülzow)	
SAS-Biometrie (QC):	Piepho (Uni Hohenheim)	
SAS-Datenmanagement*:	Schmidtke (BioMath GmbH)	
Umsetzung in PIAFStat:	Zenk und Michel (LFA Gülzow)	
Schnittstellen zu PIAF:	Zenk (LFA Gülzow)	
	* finanziert durch Bund-Länder-PIAF-Gemeinschaft	
---	---	---
Parametereingabe	Anzahl Prüfglieder	12
	Anzahl Zeilen	4
	Anzahl Spalten	12
	Zeilenbalancierung	ohne
	Spaltenbalancierung	ohne
	Anzahl Zeilengruppen	ohne / 4
	Anzahl Säulen	ohne / 12
	Anzahl Superzeilen	ohne / 4
	Anzahl Superspalten	ohne / 12
Varianzen	Zeilen	ohne / 0
	Spalten	ohne / 0
	Zeilengruppen	ohne / 0
	Säulen	ohne / 0
	gebrochene Blocks	ohne / 0
Priorität	'gleichmäßige Verteilung & Nachbarschaften'	
	Focus	'ND'
	räumlicher Kovarianz-Parameter - zweidimensional	0 / 0
	Kovarianz-Parameter zwischen Zeilen-Nachbarn	0 / 0
	Koeffizient für Gewichtung der Nachbarschaften	0,08
WDH - Systematik	Sortierung nach Variante	Zeile bzw. Zeilengruppe
Qualität & Zeit	gewählter Zeitaufwand:	schnell
	Mindestanzahl Runs	1
	gewünschte Anzahl Designs	2
	ausgeführte Runs	1
	Runs pro Design	1
	Anzahl Iterationen je RUN	auto / 384
	zusätzl. systematischer Plan	nein
	Randomisationsvariante	Permutation des Bestplanes
Ausgabe	Pfad:	C:\PROGRAMDATA\PIAF5\RES\
	Lageplan-Datei in Excel	12_4_12_12JUN17_4764.xlsx
	Parellenkürzel mit WDH?	nein
	Trennzeichen innerhalb des Kürzels	.
Zusatzinformationen	Anzahl WDH je Prüfglied - Mittelwert	5
	maximale WDH - für Eingabe in PIAF	6
	zufällige Blockungsfaktoren	
Startwert	für die Randomisation	-224692506
erstellt am	12JUN2017	

7.2.2 ‚Echte' Blockanlage (ohne Spaltenbalancierung)

Dieser Fall wird wie in 7.2.1 an der Konstellation v_r_k_s = 12_4_4_12 geprüft. Gegenüber 7.2.1 wird die Zeilenbalancierung in der Unteroption O21 durch die Vorgabe ‚auto' für die Zeilenvarianz bewirkt, was zum Design einer Blockanlage führt (Abb. 26).

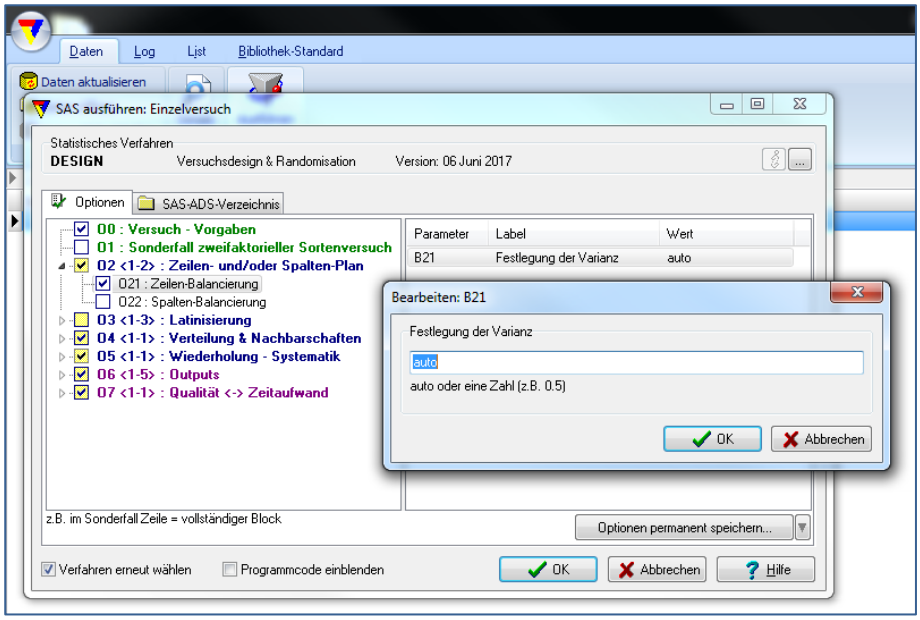


Abbildung 26: Design-Vorgaben für eine ‚echte' Blockanlage

Im optionalen Block B21 der eingeschalteten Option O21 wurde die ‚auto'-Funktion als Default für die Bestimmung einer geeigneten Varianz belassen, sie ergab eine Varianz von 0.084 (Tab. 7). Die Varianz hätte aber mit einem beliebigen Wert editiert werden können.

Tabelle 7: Protokollauszug mit Vorgaben für eine ‚echte' Blockanlage

Bereich	Bezeichnung	Vorgabe
Parametereingabe	Anzahl Prüfglieder	12
	Anzahl Zeilen	4
	Anzahl Spalten	12
	Zeilenbalancierung	ja
	Spaltenbalancierung	ohne
Varianzen	Zeilen	auto / 0.084
	Spalten	ohne / 0

Z	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6	Sp7	Sp8	Sp9	Sp10	Sp11	Sp12
4	1	5	3	12	2	11	9	7	8	6	10	4
3	10	9	6	11	2	5	8	7	4	12	1	3
2	6	12	9	2	5	11	1	3	10	8	7	4
1	6	1	10	8	9	12	11	4	5	2	7	3

ED_weighted	max_row_neighbor	n_max_row_neighbor	f_bar_A	trt_1_ED
1,6543	3	2	,6079	7

Die Spalte ‚trt_1_ED' liefert in der ersten Zeile dasjenige Prüfglied, welches im Tabellenreiter ‚Prüfglied' den geringsten Wert in der Kennzahl ‚ED' erbringt. Der Parameter ‚trt_1_ED' wird im Tabellenreiter ‚Maßzahlen so nicht bereitgestellt, er ist hier nachträglich eingefügt.

Abbildung 27: Randomisation und Maßzahlen bei einer ‚echten' Blockanlage

Die Zeilenbalancierung führt dazu, dass es keine Dopplungen eines Prüfglieds in einer Zeile mehr gibt. Die fehlende Spaltenbalancierung wirkt sich weiterhin negativ aus - es gibt Dopplungen eines Prüfglieds in einer Spalte mehr.

Da dies in realen Versuchsanstellungen allerdings kaum zu beobachten ist, kann man davon ausgehen, dass fast jeder Versuchsansteller intuitiv keine freie Randomisation innerhalb von Zeilen / Wiederholungen akzeptiert, sondern zumindest ‚händisch' Spalten balanciert. Dies wird in nachfolgendem Abschnitt berücksichtigt - damit wandelt sich das Design von einer Blocklanlage zu einem Zeilen-Spalten-Plan.

Für das praktische Versuchswesen kann der Verzicht auf eine Spaltenbalancierung i.d.R. nicht empfohlen werden. Insofern gibt es (ab 7.2.3) dann keine ‚echten' Blockanlagen mehr, sondern (ggf. erweiterte) Zeilen-Spalten-Pläne.

7.2.3 Zeilen - Spalten - Plan

Dieser Fall wird wiederum an der Konstellation $v\_r\_k\_s = 12\_4\_4\_12$ geprüft. Gegenüber 7.2.2 wird die Spaltenbalancierung in der Unteroption O21 durch die Vorgabe ‚auto' für die Spaltenvarianz bewirkt, was zum Design eines Zeilen-Spalten-Planes führt (Abb. 28).

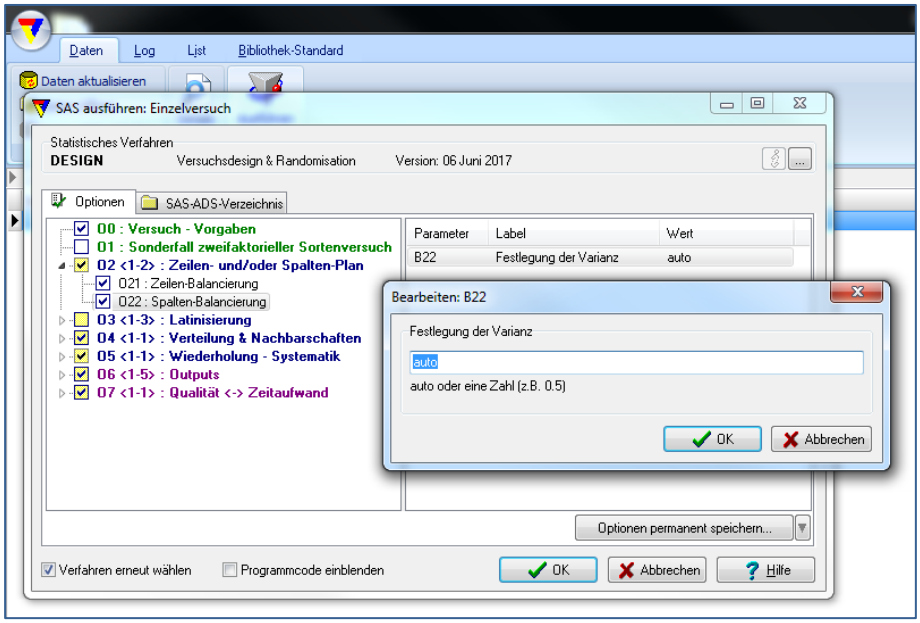


Abbildung 28: Design-Vorgaben für einen Zeilen-Spalten-Plan ohne räumliche Kovarianzen für ED und NB

Im optionalen Block B22 der eingeschalteten Option O22 wurde die ‚auto'-Funktion als Default für die Bestimmung einer geeigneten Varianz belassen, sie ergab eine Varianz von 0.006 (Tab. 8). Die Varianz könnte durch einem beliebigen Wert neu editiert werden.

Tabelle 8: Protokollauszug mit Vorgaben für einen Zeilen-Spalten-Plan

Bereich	Bezeichnung	Vorgabe
Parametereingabe	Anzahl Prüfglieder	12
	Anzahl Zeilen	4
	Anzahl Spalten	12
	Zeilenbalancierung	ja
	Spaltenbalancierung	ja
Varianzen	Zeilen	auto / 0.084
	Spalten	auto / 0.006

Z	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6	Sp7	Sp8	Sp9	Sp10	Sp11	Sp12
4	2	1	3	4	9	11	7	6	12	5	8	10
3	11	10	9	3	12	6	8	2	7	1	5	4
2	10	6	2	9	7	8	4	5	1	3	12	11
1	9	8	1	10	6	3	12	4	11	7	2	5

ED_weighted	max_row_neighbor	n_max_row_neighbor	f_bar_A	trt_1_ED
2,1207	3	1	,7984	9

Die Spalte ‚trt_1_ED' liefert in der ersten Zeile dasjenige Prüfglied, welches im Tabellenreiter ‚Prüfglied' den geringsten Wert in der Kennzahl ‚ED' erbringt. Der Parameter ‚trt_1_ED' wird im Tabellenreiter ‚Maßzahlen so nicht bereitgestellt, er ist hier nachträglich eingefügt.

Abbildung 29: Randomisation und Maßzahlen bei einem Zeilen-Spalten-Plan

Die Zeilen- und Spaltenbalancierung ist jetzt erheblich verbessert: vor allem treten jetzt keine Dopplungen eines Prüfglieds in Zeile oder Spalte auf und die Maßzahl für die Zeilen-Spalten-Effizienz f_bar_A ist erhöht.

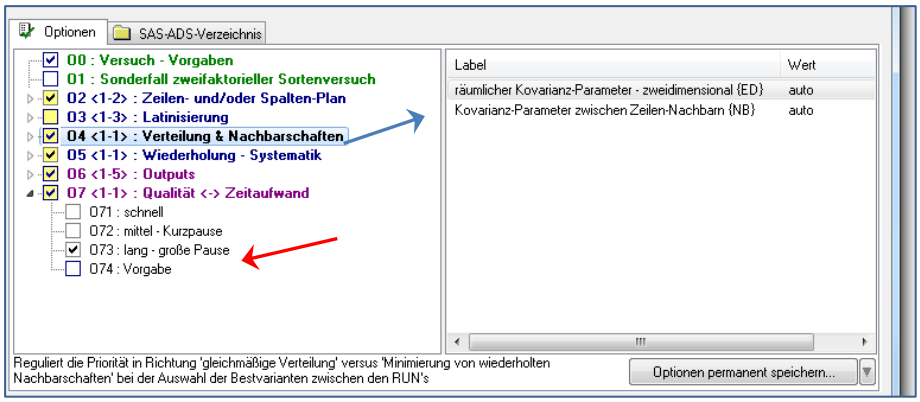
Suboptimal ist (a) die räumliche Verteilung, besonders bei Prüfglied 9 und (b) die weiterhin vielen wiederholten Nachbarschaften (Abb. 29).

Zeilen- und Spaltenbalancierung ist als Default für nahezu alle Situationen empfehlenswert.

I.d.R. sind weitere Designvorgaben empfehlenswert,

7.2.4 Erweiterter Zeilen-Spalten-Plan mit Kovarianzen für ED und NB

Dieser Fall wird wiederum an der Konstellation $v\_r\_k\_s = 12\_4\_4\_12$ geprüft. Gegenüber 7.2.3 wird die Nutzung der Kovarianz-Parameter für NB und ED im optionalen Block der Option O4 ergänzt. Für die konkreten Parameter-Werte wird die interne auto-Funktion genutzt, Editierung ist ebenso möglich. Des Weiteren wird für die Verbesserung von NB und ED die Option O73 eingeschaltet, sodass 10 eigenständige Randomisationen (Runs) erzeugt werden, von denen die ‚beste' genutzt wird (Abb. 30).



Die Spalte ‚trt_1_ED' liefert in der ersten Zeile dasjenige Prüfglied, welches im Tabellenreiter ‚Prüfglied' den geringsten Wert in der Kennzahl ‚ED' erbringt. Der Parameter ‚trt_1_ED' wird im Tabellenreiter ‚Maßzahlen so nicht bereitgestellt, er ist hier nachträglich eingefügt.

Abbildung 30: Design-Vorgaben für einen Zeilen-Spalten-Plan mit räumlichen Kovarianzen für ED und NB

Tabelle 8: Protokollauszug mit Vorgaben für einen Zeilen-Spalten-Plan mit räumlichen Kovarianzen für ED und NB

Bereich	Bezeichnung	Vorgabe
Parametereingabe	Anzahl Prüfglieder	12
	Anzahl Zeilen	4
	Anzahl Spalten	12
	Zeilenbalancierung	ja
	Spaltenbalancierung	ja
Varianzen	Zeilen	auto / 0.084
	Spalten	auto / 0.006
Priorität	'gleichmäßige Verteilung & Nachbarschaften'	
	räumlicher Kovarianz-Parameter - zweidimensional	auto / 0.0375
	Kovarianz-Parameter zwischen Zeilen-Nachbarn	auto / 0.12
Qualität & Zeit	gewählter Zeitaufwand:	länger
	ausgeführte Runs	10

Z	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6	Sp7	Sp8	Sp9	Sp10	Sp11	Sp12
4	12	7	1	9	3	8	5	2	10	4	6	11
3	1	3	5	10	7	6	12	4	9	11	8	2
2	10	12	9	8	4	3	2	11	7	5	1	6
1	11	1	4	5	9	7	8	12	2	6	10	3

Run ID	Design ID	ED_ weighted	max_row_ neighbor	n_max_row_ neighbor	f_bar_A	trt_1_ED
4	1	2,6102	1	44	,7897	2
9	2	2,5281	1	44	,7407	3
... **DesignID** 3 bis 9 aus Darstellung herausgenommen.						
3	10	2,6959	2	1	,7826	8

Die Spalte ‚trt_1_ED' liefert in der ersten Zeile dasjenige Prüfglied, welches im Tabellenreiter ‚Prüfglied' den geringsten Wert in der Kennzahl ‚ED' erbringt. Der Parameter ‚trt_1_ED' wird im Tabellenreiter ‚Maßzahlen so nicht bereitgestellt, er ist hier nachträglich eingefügt.

Abbildung 31: Randomisation und Maßzahlen bei einem Zeilen-Spalten-Plan mit Kovarianzen für ED und NB

Die Nachbarschaftsbalancierung hat sich nun erheblich verbessert. Während bis zum Punkt 7.2.3 immer Prüfgliedpaare enthalten waren, die dreifach benachbart waren, konnten nun sogar Doppel-Nachbarschaften im Best-Run vermieden werden (Abb. 31).

Die Verteilung (ED) konnte etwas verbessert werden - und zwar eher im Nahbereich (insbesondere bei Diagonalberührungen innerhalb eines Prüfgliedes). Allerdings liegt z.B. Prüfglied 2 immer noch ‚geklumpt. Hier sollte Latinisierung in Querrichtung entscheidende Verbesserungen bewirken (s. 7.2.5 - vor allem für die gleichmäßige Verteilung über die Gesamtfläche, also im Fernbereich).

Wenn die Spaltenzahl größer als die Prüfgliedzahl ist, werden beide Parameter in der auto-Funktion auf ‚Null' gesetzt, da es erfahrungsgemäß sonst manchmal zu systematischen ungünstigen Anordnungen in der Randomisation kam.

Die Nutzung der räumlichen Parameter für ED und NB ist als Default für die meisten Situationen zu empfehlen.

Werden bei kleinen Versuchen ungewollt systematische Spaltenzusammensetzungen beobachtet, liegt eine Konkurrenz zur Nachbarschaftsbalancierung vor - es wird auf Abschnitt 8 verwiesen, ggf. muss im Einzelfall auf diese Kovarianzen verzichtet werden.

7.2.5 Erweiterter Zeilen-Spalten-Plan als Lateinisches Rechteck

Dieser Fall wird wiederum an der Konstellation $v\_r\_k\_s = 12\_4\_4\_12$ geprüft. Gegenüber 7.2.4 wird die Nutzung von ‚Säulen' im optionalen Block der Option O32 ergänzt. Für die Anzahl Säulen wird die interne auto-Funktion genutzt - das führt hier zu vier Säulen entsprechend vier Wiederholungen. Für die Säulen-Varianz wird ebenso der auto-Default genutzt (Abb. 32).

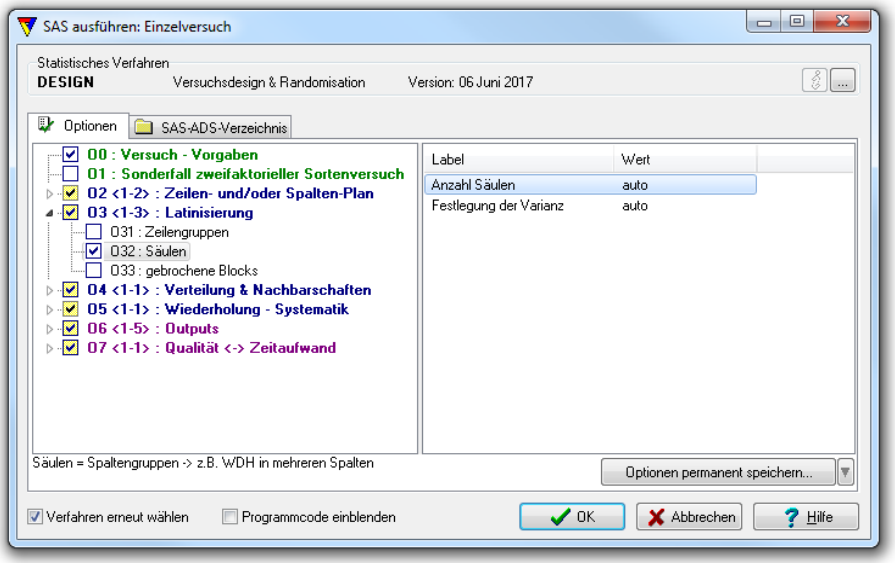


Abbildung 32: Design-Vorgaben für einen erweiterten Zeilen-Spalten-Plan als Lateinisches Rechteck

Tabelle 9: Protokollauszug mit Vorgaben für einen erweiterten Zeilen-Spalten-Plan als Lateinisches Rechteck

Bereich	Bezeichnung	Vorgabe
Parametereingabe	Anzahl Prüfglieder	12
	Anzahl Zeilen	4
	Anzahl Spalten	12
	Zeilenbalancierung	ja
	Spaltenbalancierung	ja
	Anzahl Säulen	auto / 4
Varianzen	Säulen	auto / 1
Priorität	'gleichmäßige Verteilung & Nachbarschaften'	
	räumlicher Kovarianz-Parameter - zweidimensional	auto / 0.0375
	Kovarianz-Parameter zwischen Zeilen-Nachbarn	auto / 0.12
Qualität & Zeit	gewählter Zeitaufwand:	länger
	ausgeführte Runs	10

Durch die Nutzung der Säulen wird automatisch ein zusätzlicher Tabellenreiter ‚Säulen' in der Excel-Tabelle ausgegeben (Abb. 33).

Abbildung 33: Säulen in einem Zeilen-Spalten-Plan als Lateinisches Rechteck

Z	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6	Sp7	Sp8	Sp9	Sp10	Sp11	Sp12
4	12	7	9	10	6	4	2	8	1	5	11	3
3	4	1	11	8	3	9	5	6	7	10	12	2
2	8	10	2	5	7	11	4	12	3	1	9	6
1	5	3	6	12	1	2	9	11	10	4	8	7

Run ID	Design ID	ED_ weighted	max_row_ neighbor	n_max_row_ neighbor	f_bar_A	trt_1_ED
4	1	3,232	1	44	,6910	9
9	2	3,132	1	44	,4835	10
... DesignID 3 bis 9 aus Darstellung herausgenommen.						
6	10	2,991	2	1	,7557	1

Die Spalte ‚trt_1_ED' liefert in der ersten Zeile dasjenige Prüfglied, welches im Tabellenreiter ‚Prüfglied' den geringsten Wert in der Kennzahl ‚ED' erbringt. Der Parameter ‚trt_1_ED' wird im Tabellenreiter ‚Maßzahlen so nicht bereitgestellt, er ist hier nachträglich eingefügt.

Abbildung 34: Randomisation und Maßzahlen bei einem Zeilen-Spalten-Plan als Lateinisches Rechteck

Die gleichmäßige Verteilung (ED) konnte erheblich verbessert werden - und zwar nun auch im Fernbereich. Auch konnten Doppel-Nachbarschaften im Best-Run vermieden werden. Die Säulenbildung führt zu etwas verschlechterter Zeilen- und Spaltenbalancierung, insbesondere ist sie zwischen Runs instabiler - es lohnen sich also mehrere Runs.

In besonders geeigneten Versuchsstrukturen sind die latinisierten Blöcke sowohl rechteckig als auch balanciert (enthalten jedes Prüfglied genau einmal). Dies erfüllt die Bedingungen des klassischen lateinischen Rechteckes.

Ist diese günstige Kombination nicht gegeben, erhält eines der beiden Ziele Vorrang. Die Blockform ‚krumm' zielt vorrangig auf Vollständigkeit ab (7.2.6), die Blockform ‚rechteckig' erzwingt Rechteckigkeit unabhängig von dem Erreichen einer Vollständigkeit (7.2.7).

Säulen führen bei ‚breiten' Versuchen zu einer deutlich verbesserten Querverteilung aller Prüfglieder.

7.2.6 Unorthogonales Lateinisches Rechteck mit ‚krummen' Blocks

Dieser Fall wird an der veränderten Konstellation v_r_k_s = 14_4_4_14 geprüft. Wie in 7.2.5 wird die Nutzung von ‚Säulen' im optionalen Block der Option O32 ergänzt. Für die Anzahl Säulen wird die interne auto-Funktion genutzt - das führt auch hier zu vier Säulen entsprechend vier Wiederholungen. Für die Säulen-Varianz wird ebenso der auto-Default genutzt (Abb. 35).

Allerdings kann in dieser Konstellation mit 14 Prüfgliedern in 4 Zeilen eine Säule nicht sowohl vollständig besetzt als auch rechteckig sein. In dieser Szenarium wird im optionalem Block der Option O3 die Blockform ‚krumm' gewählt. Diese Blockform gibt der vollständigen Besetzung Vorrang vor Rechteckigkeit des Blocks (hier also der Säule).

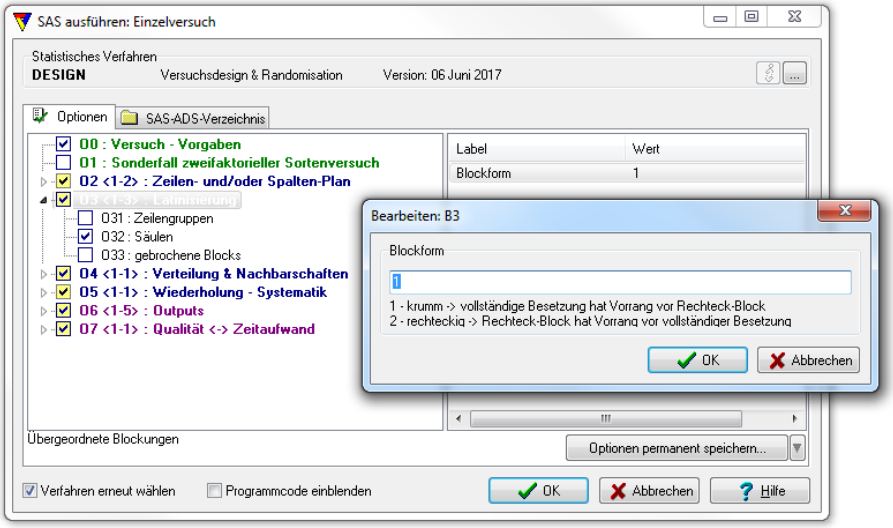


Abbildung 35: Design-Vorgaben für ein Lateinisches Rechteck mit ‚krummen' Blocks

Tabelle 10: Protokollauszug mit Vorgaben für ein Lateinisches Rechteck mit ‚krummen' Blocks

Bereich	Bezeichnung	Vorgabe
Parametereingabe	Anzahl Prüfglieder	14
	Anzahl Zeilen	4
	Anzahl Spalten	14
	Zeilenbalancierung	ja
	Spaltenbalancierung	ja
	Anzahl Säulen	auto / 4
	Blockform	'krumm'
Qualität & Zeit	ausgeführte Runs	10

Die krumme Blockform ist in Abb. 36 visualisiert. In den Spalten 4 und 11 wechselt die Codierung der Säulen. Wenn es Probleme mit der Spaltenbalancierung geben sollte (insbesondere Dopplung eines Prüfgliedes), dann sind solche Spalten die riskanten.

‘krumme' Säulen im Plan

Z	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6	Sp7	Sp8	Sp9	Sp10	Sp11	Sp12	Sp13	Sp14
4	1	1	1	2	2	2	2	3	3	3	4	4	4	4
3	1	1	1	2	2	2	2	3	3	3	4	4	4	4
2	1	1	1	1	2	2	2	3	3	3	3	4	4	4
1	1	1	1	1	2	2	2	3	3	3	3	4	4	4

Plan 1 mit Prüfglied-Kürzel

Z	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6	Sp7	Sp8	Sp9	Sp10	Sp11	Sp12	Sp13	Sp14
4	1	4	12	3	10	6	11	9	7	2	8	5	14	13
3	8	3	7	5	2	14	4	6	1	13	10	9	12	11
2	10	5	6	13	12	1	9	8	14	3	11	2	4	7
1	2	9	14	11	8	7	13	4	10	12	5	1	3	6

RunID	DesignID	ED_ weighted	max_row_ neighbor	n_max_row_ neighbor	f_bar_A	trt_1_ED
9	1	3,777	1	52	,7434	9
5	2	3,4986	1	52	,7143	3
... **DesignID** 3 bis 9 aus Darstellung herausgenommen.						
2	10	2,9422	1	52	,71710	5

Die Spalte ‚trt_1_ED‘ liefert in der ersten Zeile dasjenige Prüfglied, welches im Tabellenreiter ‚Prüfglied‘ den geringsten Wert in der Kennzahl ‚ED‘ erbringt. Der Parameter ‚trt_1_ED‘ wird im Tabellenreiter ‚Maßzahlen so nicht bereitgestellt, er ist hier nachträglich eingefügt.

Abbildung 36: Blockstruktur, Randomisation und Maßzahlen bei einem Lateinischen Rechteck mit ‚krummen‘ Blocks

Die Blockform ‚krumm‘ ist ein geeigneter Default für die meisten einfach strukturierten Designs und für Situationen mit vielen Prüfgliedern.

Bei mittlerer Prüfgliedzahl (ca. 8 bis 16) lohnt ein Vergleich beider Blockformen.

Bei komplexen Design-Vorgaben kann die Blockform ‚krumm‘ für die Nachbarschafts-balancierung kritischer als ‚rechteckig‘ sein.

Nachteilig kann diese Blockform bei Mehrfach-Latinisierung in unbalancierten Strukturen sein.

7.2.7 Unorthogonales Lateinisches Rechteck mit rechteckigen Blocks

Dieser Fall wird wie in 7.2.6 an der Konstellation v_r_k_s = 14_4_4_14 geprüft. Auch in dieser Konstellation mit 14 Prüfgliedern in 4 Zeilen kann eine Säule nicht sowohl vollständig besetzt als auch rechteckig sein. Im Unterschied zu 7.2.6 wird im optionalem Block der Option O3 die Blockform ‚rechteckig' gewählt. Diese Blockform gibt der Rechteckigkeit des Blocks (hier also der Säule) Vorrang vor vollständiger Besetzung. Die Säulen sind also unterschiedlich besetzt, z.T. unter- und z.T. überbesetzt und sind daher verschieden groß (Abb. 37).

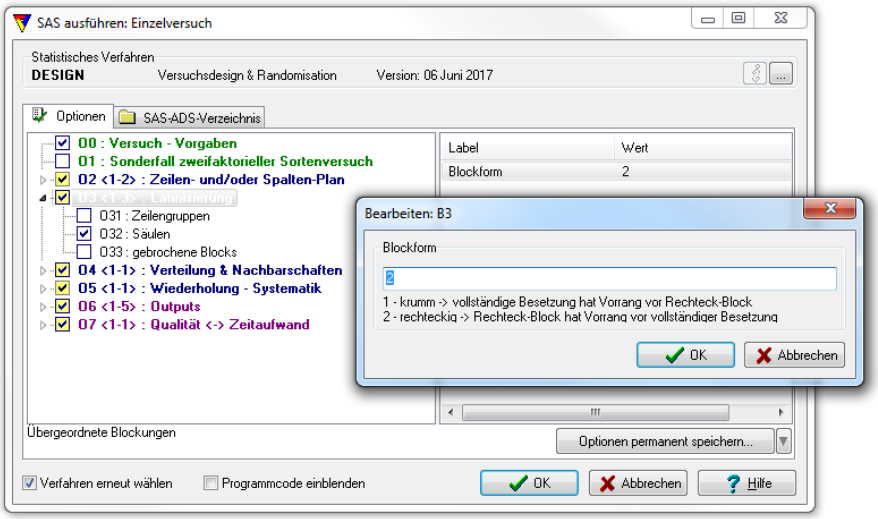


Abbildung 37: Design-Vorgaben für ein Lateinisches Rechteck mit rechteckigen Blocks

Tabelle 11: Protokollauszug mit Vorgaben für ein Lateinisches Rechteck mit rechteckigen Blocks

Bereich	Bezeichnung	Vorgabe
Parametereingabe	Anzahl Prüfglieder	14
	Anzahl Zeilen	4
	Anzahl Spalten	14
	Zeilenbalancierung	ja
	Spaltenbalancierung	ja
	Anzahl Säulen	auto / 4
	Blockform	'rechteckig'

Säulen im Plan

Z	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6	Sp7	Sp8	Sp9	Sp10	Sp11	Sp12	Sp13	Sp14
4	1	1	1	1	2	2	2	3	3	3	4	4	4	4
3	1	1	1	1	2	2	2	3	3	3	4	4	4	4
2	1	1	1	1	2	2	2	3	3	3	4	4	4	4
1	1	1	1	1	2	2	2	3	3	3	4	4	4	4

Plan 1 mit Prüfglied-Kürzel

Z	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6	Sp7	Sp8	Sp9	Sp10	Sp11	Sp12	Sp13	Sp14
4	13	11	7	12	10	1	2	5	8	6	3	9	14	4
3	1	5	4	3	8	7	6	14	13	9	10	11	12	2
2	2	14	11	6	4	13	5	10	7	3	1	12	9	8
1	9	4	10	8	12	3	14	1	11	2	7	5	6	13

RunID	DesignID	ED_ weighted	max_row_ neighbor	n_max_row_ neighbor	f_bar_A	trt_1_ED
10	1	3,0157	1	52	,7560	7
3	2	2,9422	1	52	,7438	8

... DesignID 3 bis 9 aus Darstellung herausgenommen.

6	10	3,1385	2	2	,7242	1

Die Spalte ‚trt_1_ED' liefert in der ersten Zeile dasjenige Prüfglied, welches im Tabellenreiter ‚Prüfglied' den geringsten Wert in der Kennzahl ‚ED' erbringt. Der Parameter ‚trt_1_ED' wird im Tabellenreiter ‚Maßzahlen so nicht bereitgestellt, er ist hier nachträglich eingefügt.

Abbildung 38: Blockstruktur, Randomisation und Maßzahlen bei einem Lateinischen Rechteck mit rechteckigen Blocks

Bezüglich ED ist diese Blockform in gegebener Konstellation unterlegen. Außerdem erreichen nur 2 von 10 Runs bestmögliche NB mit $(\max\_row\_neighbor = 1)$. Allerdings ist die Zeilen-Spalten-Balancierung etwas besser, da die Säulen/Spalten-Struktur im Gegensatz zur Blockform ‚krumm' eindeutig hierarchisch definiert ist (Abb. 38).

Im Verfahren DESIGN ist vorrangig die linke Außen-Säule überbesetzt, danach die rechte Außensäule und erst danach innen liegende Säulen. Anlass dafür ist, dass in unterbesetzten Außen-Säulen eine dort fehlendes Püfglied zudem in weiteren x angrenzenden Spalten der benachbarten inneren Spaltengruppe fehlen könnte und damit keine ausgewogene räumliche Verteilung erzielt werden kann. Blockform ‚krumm' löst dieses Problem auf.

Die Blockform ‚rechteckig' kann bei geringer Prüfgliedzahl von Vorteil für die gleichmäßige Verteilung und bei sehr komplexen Design-Vorgaben für die Nachbarschaftsbalancierung sein.

Von der Blockform ‚rechteckig' ist abzuraten, wenn mindestens eine der beiden Außen-Säulen unterbesetzt sein würde ($v > N_{Parzellen\ je\ Gruppe}$).

7.2.8 Erweiterter Zeilen-Spalten-Plan mit Säulen und Gebrochenen Blocks

Dieser Fall wird wie in 7.2.6 an der Konstellation v_r_k_s = 14_4_4_14 mit ‚krummen' Säulen geprüft. Wie in 7.2.6 können Säulen nicht sowohl vollständig besetzt als auch rechteckig sein. Allerdings gelingt dies mit Blocks aus $2\ Zeilen \times 7\ Spalten$, also in Form von Quadranten. In Ergänzung zu 7.2.6 werden in der Option O33 ‚Gebrochene Blocks' angewiesen, wobei für Blockzahl und Blockraster der auto-Default genutzt wird (Abb. 39).

Damit entstehen als gebrochene Blocks die Quadranten: links-unten, rechts-unten, links-oben und rechts-oben Abb. 40, oben).

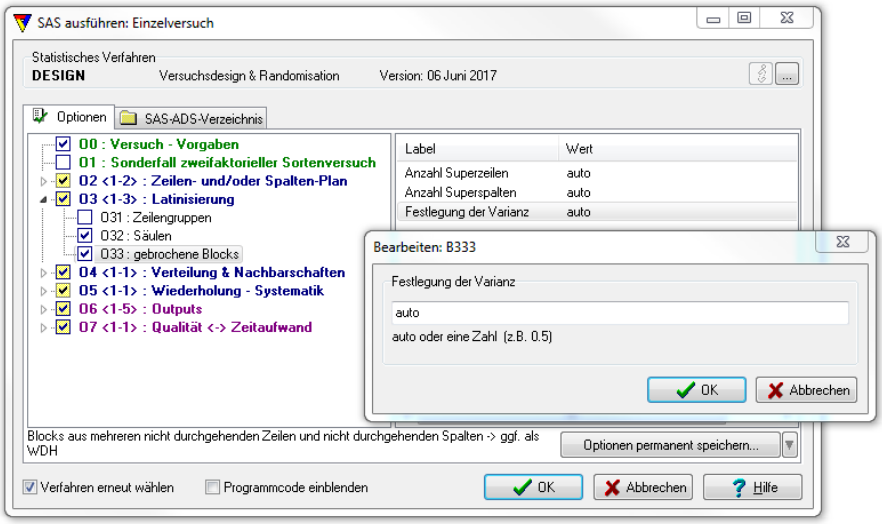
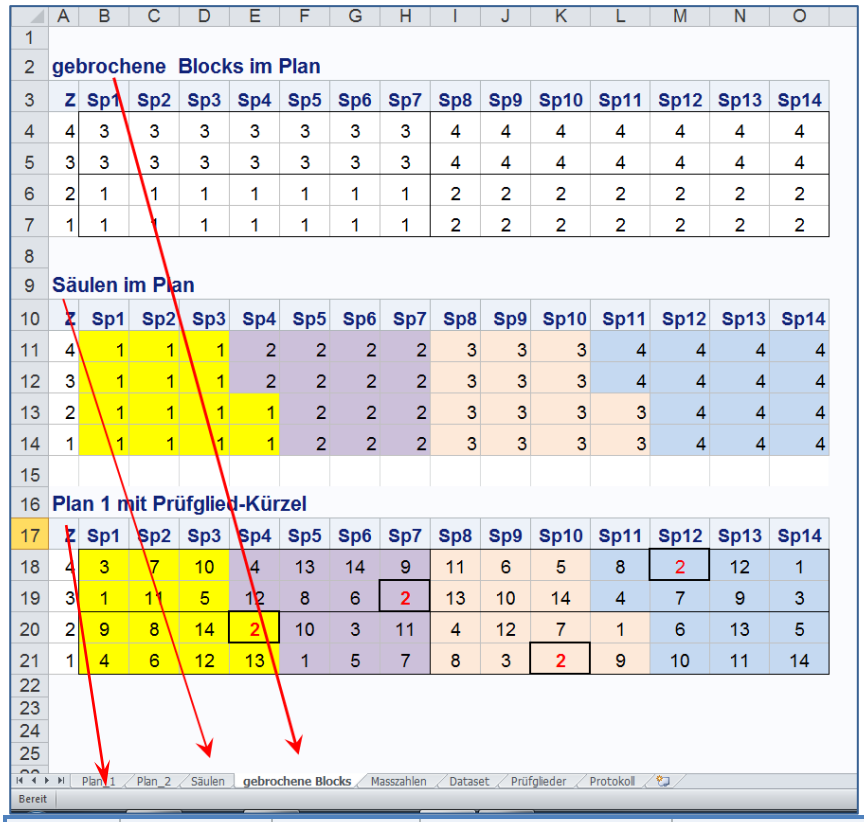


Abbildung 39: Design-Vorgaben für einen Zeilen-Spalten-Plan mit Säulen und Gebrochenen Blocks

Tabelle 12: Protokollauszug mit Vorgaben für einen Zeilen-Spalten-Plan mit Säulen und Gebrochenen Blocks

Bereich	Bezeichnung	Vorgabe
Parametereingabe	Anzahl Prüfglieder	14
	Anzahl Zeilen	4
	Anzahl Spalten	14
	Zeilenbalancierung	ja
	Spaltenbalancierung	ja
	Anzahl Säulen	auto / 4
	Anzahl Superzeilen	auto / 2
	Anzahl Superspalten	auto / 2
	Blockform	'krumm'
Varianzen	Zeilen	auto / 0.0862
	Spalten	auto / 0.0046
	Säulen	auto / 1
	gebrochene Blocks	auto / 1



gebrochene Blocks im Plan

Z	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6	Sp7	Sp8	Sp9	Sp10	Sp11	Sp12	Sp13	Sp14
4	3	3	3	3	3	3	3	4	4	4	4	4	4	4
3	3	3	3	3	3	3	3	4	4	4	4	4	4	4
2	1	1	1	1	1	1	1	2	2	2	2	2	2	2
1	1	1	1	1	1	1	1	2	2	2	2	2	2	2

Säulen im Plan

Z	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6	Sp7	Sp8	Sp9	Sp10	Sp11	Sp12	Sp13	Sp14
4	1	1	1	2	2	2	2	3	3	3	4	4	4	4
3	1	1	1	2	2	2	2	3	3	3	4	4	4	4
2	1	1	1	1	2	2	2	3	3	3	3	4	4	4
1	1	1	1	1	2	2	2	3	3	3	3	4	4	4

Plan 1 mit Prüfglied-Kürzel

Z	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6	Sp7	Sp8	Sp9	Sp10	Sp11	Sp12	Sp13	Sp14
4	3	7	10	4	13	14	9	11	6	5	8	2	12	1
3	1	11	5	12	8	6	2	13	10	14	4	7	9	3
2	9	8	14	2	10	3	11	4	12	7	1	6	13	5
1	4	6	12	13	1	5	7	8	3	2	9	10	11	14

Plan_1 | Plan_2 | Säulen | gebrochene Blocks | Masszahlen | Dataset | Prüfglieder | Protokoll

Bereit

RunID	DesignID	ED_ weighted	max_row_ neighbor	n_max_row_ neighbor	f_bar_A	trt_1_ED
6	1	3,5943	1	52	,7429	**2**
4	2	3,4899	1	52	,7224	5
... **DesignID** 3 bis 9 aus Darstellung herausgenommen.						
5	10	3,2068	2	1	,7301	13

Die Spalte ‚trt_1_ED' liefert in der ersten Zeile dasjenige Prüfglied, welches im Tabellenreiter ‚Prüfglied' den geringsten Wert in der Kennzahl ‚ED' erbringt. Der Parameter ‚trt_1_ED' wird im Tabellenreiter ‚Maßzahlen so nicht bereitgestellt, er ist hier nachträglich eingefügt.

Abbildung 40: Blockstruktur, Randomisation und Maßzahlen bei einem Zeilen-Spalten-Plan mit Säulen und Gebrochenen Blocks

Bezüglich ED ist diese zusätzliche Latinisierung in gegebener Konstellation zur Variante ohne Gebrochene Blocks in 7.2.6 praktisch gleichwertig. Es erreichen allerdings nur 7 von 10 Runs bestmögliche NB mit $(\max\_row\_neighbor = 1)$ - ohne zusätzliche Gebrochene Blocks erreichten dies alle 10 Runs.

Für die sehr häufige Konstellation mit 4 Wiederholungen=Zeilen: $v\_r\_k\_s = v\_4\_4\_s$ werden für $v = s = 3 \cdots 30$ nachfolgende Empfehlungen gegeben.

Säulenbildung kann bei geradem v erst ab ca. $v \geq 6$ Vorteile erbringen. Dies sollte aber nicht pauschal angenommen, sondern in zwei Szenarien geprüft werden.

Bei ungeradem v ist die Aufnahme Gebrochener Blocks meist ungünstig.

Auch für andere Konstellation mit $r = k \neq 4$ ist die Nutzung der Latinisierung in Gebrochenen Blocks nicht durchweg als Default zu empfehlen. Es sollte ggf. vielmehr ein Szenarienvergleich erfolgen (mit vs. ohne Gebrochene Blocks) und aufgrund der Zielparameter entschieden werden.

7.2.9 Lateinisches Rechteck mit Gitterstruktur

Dieser Fall wird wie in 7.2.5 an der Konstellation v_r_k_s = 12_4_4_12 geprüft. Ebenso wird die Nutzung von ‚Säulen' genutzt und wie in 7.2.8 werden zusätzlich in der Option O33 ‚Gebrochene Blocks' angewiesen. In diesem Fall wird für Blockzahl und Blockraster nicht der auto-Default genutzt, welcher zu vollständigen Wiederholungen führen würde. Stattdessen werden Teilblocks innerhalb von Zeilen erzeugt (Abb. 41). Das ergibt also de facto eine Gitteranlage (Abb. 42, oben).

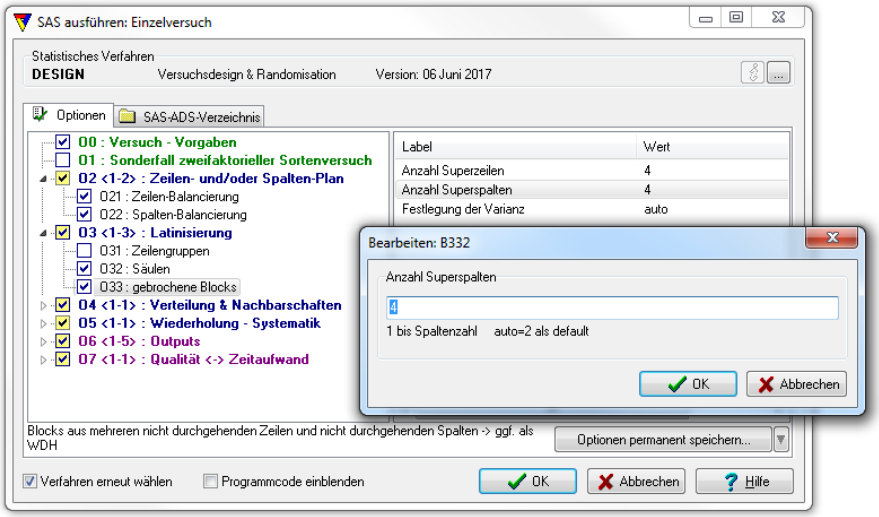
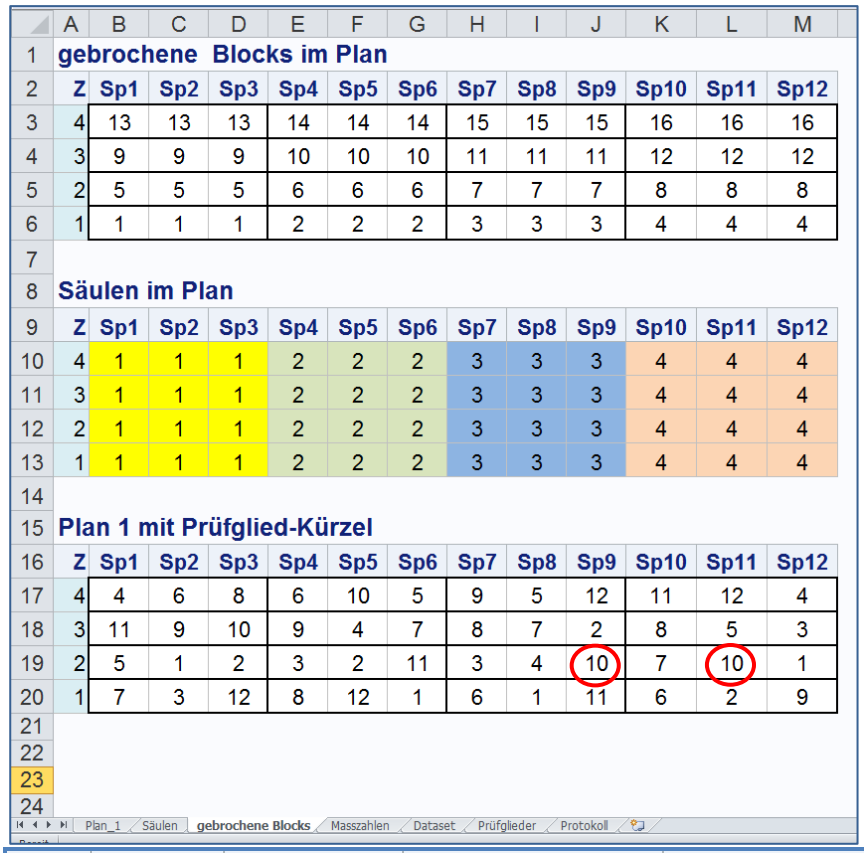


Abbildung 41: Design-Vorgaben für ein Lateinisches Rechteck mit Gitterstruktur

Tabelle 13: Protokollauszug mit Vorgaben für ein Lateinisches Rechteck mit Gitterstruktur

Bereich	Bezeichnung	Vorgabe
Parametereingabe	Anzahl Prüfglieder	12
	Anzahl Zeilen	4
	Anzahl Spalten	12
	Zeilenbalancierung	ja
	Spaltenbalancierung	ja
	Anzahl Säulen	auto / 4
	Anzahl Superzeilen	4 / 4
	Anzahl Superspalten	4 / 4



gebrochene Blocks im Plan

Z	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6	Sp7	Sp8	Sp9	Sp10	Sp11	Sp12
4	13	13	13	14	14	14	15	15	15	16	16	16
3	9	9	9	10	10	10	11	11	11	12	12	12
2	5	5	5	6	6	6	7	7	7	8	8	8
1	1	1	1	2	2	2	3	3	3	4	4	4

Säulen im Plan

Z	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6	Sp7	Sp8	Sp9	Sp10	Sp11	Sp12
4	1	1	1	2	2	2	3	3	3	4	4	4
3	1	1	1	2	2	2	3	3	3	4	4	4
2	1	1	1	2	2	2	3	3	3	4	4	4
1	1	1	1	2	2	2	3	3	3	4	4	4

Plan 1 mit Prüfglied-Kürzel

Z	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6	Sp7	Sp8	Sp9	Sp10	Sp11	Sp12
4	4	6	8	6	10	5	9	5	12	11	12	4
3	11	9	10	9	4	7	8	7	2	8	5	3
2	5	1	2	3	2	11	3	4	10	7	10	1
1	7	3	12	8	12	1	6	1	11	6	2	9

Plan_1 | Säulen | gebrochene Blocks | Masszahlen | Dataset | Prüfglieder | Protokoll

Run ID	Design ID	ED_ weighted	max_row_ neighbor	n_max_row_ neighbor	f_bar_A	trt_1_ED
5	1	2,9606	2	9	,6011	10
7	2	2,8863	2	11	,6950	2
... **DesignID** 3 bis 9 aus Darstellung herausgenommen.						
6	10	2,5268	2	10	,6455	3

Die Spalte ‚trt_1_ED' liefert in der ersten Zeile dasjenige Prüfglied, welches im Tabellenreiter ‚Prüfglied' den geringsten Wert in der Kennzahl ‚ED' erbringt. Der Parameter ‚trt_1_ED' wird im Tabellenreiter ‚Maßzahlen so nicht bereitgestellt, er ist hier nachträglich eingefügt.

Abbildung 42: Blockstruktur, Randomisation und Maßzahlen bei einem Lateinischen Rechteck mit Gitterstruktur (3×1 - Teilblocks)

Es zeigt sich, dass die Gitterstruktur zu Verschlechterungen in allen Parametern führt (im Vergleich zum Lateinischen Rechteck nach 3.2.5). Insbesondere erkennt man die Verschlechterung der Zeilen-Balancierung augenfällig daran, dass die Zeilen nicht mehr orthogonal besetzt sind - so ist u.a. das kritische Prüfglied 10 zweimal in Zeile 2. (Anmerkung: derart schwerwiegende Mängel sind in der Version ab 2018 sehr unwahrscheinlich geworden.)

Insofern favorisieren wir Gitterstrukturen nicht, sondern halten Anlagen mit besserer ED und NB (durch die auto-Defaults + Latinisierung zu annähernd vollständigen Blockeinheiten) für aussichtsreicher. Gegenüber ggf. eintretender Bodenheterogenität ist diese Strategie erstens robuster und kann statt durch Schätzung von Teilblockeffekten durch die Aufnahme einer räumlichen Kovarianz im Auswertungsmodell adäquater berücksichtigt werden:

Erläuterungen dazu (LATINISIERUNG in großen Blocks + ggf. räumliches Modell vs. GITTERANLAGEN):

LATINISIERUNG (1-3-fach) mit dem Ansatz, jeweils annähernd vollst. Blocks zu bilden (das ist ‚auto'-Default im Verfahren DESIGN), führt zu ganz erheblicher Verbesserung der ED. Das sollte bereits die MEANS deutlich robuster gegenüber räumlicher Varianz machen. Wird dann Bodenheterogenität mit einem räumlichen Ansatz erkannt und ist im Modell effizient (AICC), dann kann Adjustierung durch latinisierte Blocks + räumliches Modell zu noch weiter 'verbesserten' LSMEANS führen. Ein räumliches (stetiges) Modell passt sich zumindest potenziell dem ebenfalls stetigen Verlauf von Heterogenität (mehr oder weniger gut) an und ist im Ansatz adäquat.

GITTER-Strukturen (kleine Teilblocks) werden als Raster der Fläche 'übergestülpt', ohne Vorkenntnis der konkreten stetigen Heterogenität, nur in der Hoffnung, dass die geschätzten Effekte kleiner Teilblocks die Heterogenität hinreichend fassen. Trotz des Widerspruchs zwischen stetiger Heterogenität und distinkten Blocks (die Annahme, dass innerhalb von Teilblocks

räumliche Effekte konstant sind und dass sie an den Blockgrenzen springen, ist a priori nicht adäquat). Insbesondere führt aber eine Balancierung nach Gitterstrukturen hinsichtlich ED zum Gegenteil von Latinisierung --> zu einer Verschlechterung. Die MEANS werden also zunächst statt robuster sogar anfälliger. Alle Hoffnung, das auszugleichen und möglichst sogar zu überkompensieren wird auf die Adjustierung gelegt, also auf eine präzise Schätzung der Teilblockeffekte und darauf, dass diese den Großteil aller Fehlerwirkungen verursachen.

Wie sieht es aber aus, wenn der Varianzanteil zwar bedeutsam, aber trotzdem vglw. klein ist, also z.B. <30% der ehemalige Restvarianz? Das halte ich für realistischer/häufiger als z.B. >70%. Dann greift SCHRUMPFUNG spürbar. Primär werden die räumlichen stetigen bzw. bei Gittern die distinkten Teilblockeffekte geschrumpft, sekundär verhalten sich dadurch die LSMEANS konservativ in Richtung MEANS. Und das bedeutet:

Bei einem räumlichen Modell + Latinisierung werden die LSMEANS zu bereits robusteren MEANS 'gezogen'.

Bei der Gitteranlage werden die LSMEANS zu den viel sensibleren MEANS gezogen.

Alles in allem halte ich die Gitteranlagen für weniger aussichtsreich, seit wir in der Lage sind, (a) Heterogenität auch durch ein räumliches Modell zu berücksichtigen und zudem (b) durch Latinisierung (b1) eine gute räumliche Verteilung erzielen und (b2) diese Latinisierung im Modell berücksichtigen können.

In der Auswertung ist $‚Gebrochener\ Block' = Teilblock$ identisch zur Codierung $Zeile(Säule)$ (sprich: Zeile in Säule) bzw. im zufälligen Ansatz im rechnerischen Ergebnis auch identisch zu $Zeile \times Säule$. Auf diese Weise kann eine adäquate Auswertung in PIAFStat/ EVA oder ZVA sichergestellt werden.

7.2.10 Lateinisches Quadrat als idealer Zeilen-Spalten-Plan

Dieser Fall wird an der Konstellation $v\_r\_k\_s = 6\_6\_6\_6$ geprüft. In solchen Fällen wird jede Zeile wie auch jede Spalte zur vollständigen Wiederholung. Hierfür reicht beim Lateinischen Quadrat die Zeilen- und Spaltenbalancierung durch Option O2, Latinisierung ist dafür nicht erforderlich (Abb. 43).

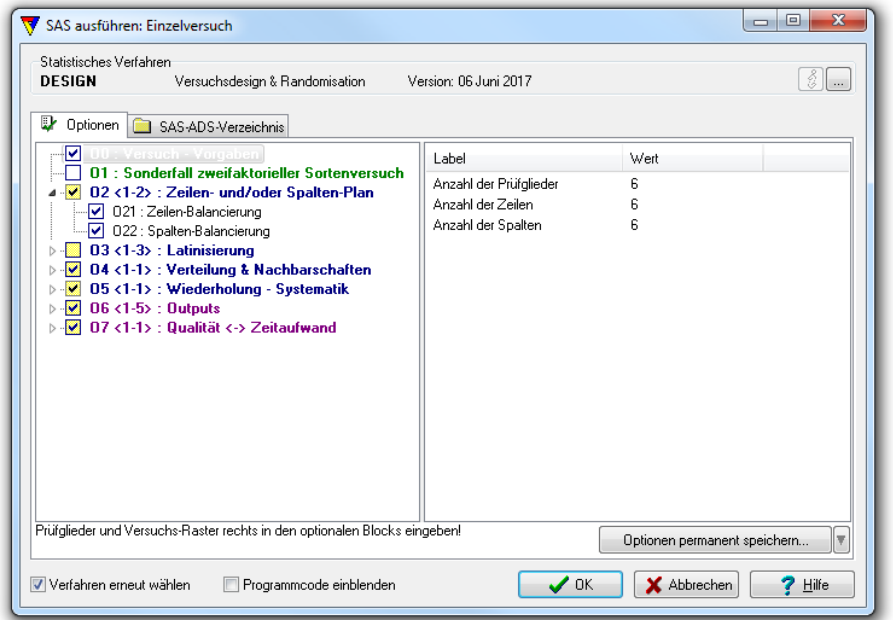


Abbildung 43: Design-Vorgaben für ein Lateinisches Quadrat

Tabelle 14: Protokollauszug mit Vorgaben für ein Lateinisches Quadrat

Bereich	Bezeichnung	Vorgabe
Parametereingabe	Anzahl Prüfglieder	6
	Anzahl Zeilen	6
	Anzahl Spalten	6
	Zeilenbalancierung	ja
	Spaltenbalancierung	ja
Varianzen	Zeilen	auto / 0.069
	Spalten	auto / 0.069

Z	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6
6	1	5	3	2	6	4
5	4	2	6	5	1	3
4	5	3	4	1	2	6
3	6	1	2	3	4	5
2	2	4	5	6	3	1
1	3	6	1	4	5	2

Run ID	Design ID	ED_ weighted	max_row_ neighbor	n_max_ row_neighbor	f_bar_A	trt_1_ED
6	1	2,3439	3	2	1	5
10	2	2,2689	3	2	1	5
... DesignID 3 bis 9 aus Darstellung herausgenommen.						
8	10	2,2020	4	1	1	5

Die Spalte ‚trt_1_ED' liefert in der ersten Zeile dasjenige Prüfglied, welches im Tabellenreiter ‚Prüfglied' den geringsten Wert in der Kennzahl ‚ED' erbringt. Der Parameter ‚trt_1_ED' wird im Tabellenreiter ‚Maßzahlen so nicht bereitgestellt, er ist hier nachträglich eingefügt.

Abbildung 44: Randomisation und Maßzahlen bei einem Lateinischen Quadrat

Der Parameter für NB zeigt mit dem $f\_bar\_A = 1$ an, dass die Zeilen- und Spaltenbalancierung durch das lateinische Quadrat in idealer (mit ‚1' bzw. zu 100%) Weise erfüllt ist (Abb. 44).

Dies ergibt sich aus folgenden Aspekten:

- Jedes Püfglied steht in jeder Zeile und Spalte genau einmal.
- Alle Paar von Prüfgliedern stehen damit logischerweise auch gleich oft gemeinsam in Zeilen und in Spalten.

7.2.11 Dreifach-gerechtes Lateinisches Quadrat

Dieser Fall wird wie in 7.2.10 an der Konstellation $v\_r\_k\_s = 6\_6\_6\_6$ geprüft. In solchen Fällen wird jede Zeile wie auch jede Spalte zur vollständigen Wiederholung. Zusätzlich wird nun die Möglichkeit genutzt, eine dritte Form vollständiger Wiederholungen zu konstruieren: Gebrochene Blocks im 2×3 oder im 3×2 - Raster. Im Folgenden wird in O33 ein Raster aus $3\ Superzeilen \times 2\ Superspalten$ - wegen der meist schmal-langen Parzellenform - bevorzugt (Abb. 45).

Derartige dreifach orthogonalen Konstrukte wurden als *Gerechte Pläne* bzw. *Gerechte Designs* publiziert.

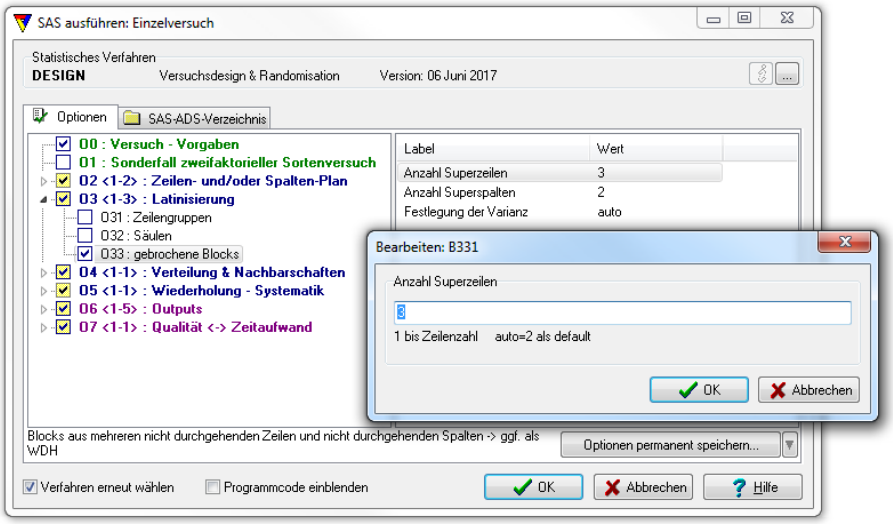


Abbildung 45: Design-Vorgaben für ein Dreifach-gerechtes Lateinische Quadrat

Tabelle 15: Protokollauszug mit Vorgaben für ein Dreifach-gerechtes Lateinische Quadrat

Bereich	Bezeichnung	Vorgabe
Parametereingabe	Anzahl Prüfglieder	6
	Anzahl Zeilen	6
	Anzahl Spalten	6
	Zeilenbalancierung	ja
	Spaltenbalancierung	ja
	Anzahl Superzeilen	3 / 3
	Anzahl Superspalten	2 / 2

Gebrochene Blocks im Plan

Z	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6
6	5	5	5	6	6	6
5	5	5	5	6	6	6
4	3	3	3	4	4	4
3	3	3	3	4	4	4
2	1	1	1	2	2	2
1	1	1	1	2	2	2

Plan 1 mit Prüfglied-Kürzel

Z	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6
6	5	2	1	3	4	6
5	4	3	6	5	1	2
4	6	1	4	2	3	5
3	2	5	3	1	6	4
2	3	6	2	4	5	1
1	1	4	5	6	2	3

Run ID	Design ID	ED_ weighted	max_row_ neighbor	n_max_ row_neighbor	f_bar_A	trt_1_ED
1	1	2,3805	2	15	1	1
2	2	2,3805	3	2	1	1
... DesignID 3 bis 9 aus Darstellung herausgenommen.						
5	10	2,2172	3	4	1	1

Die Spalte ‚trt_1_ED' liefert in der ersten Zeile dasjenige Prüfglied, welches im Tabellenreiter ‚Prüfglied' den geringsten Wert in der Kennzahl ‚ED' erbringt. Der Parameter ‚trt_1_ED' wird im Tabellenreiter ‚Maßzahlen so nicht bereitgestellt, er ist hier nachträglich eingefügt.

Abbildung 46: Blockstruktur, Randomisation und Maßzahlen bei einem Dreifach-gerechten Lateinischen Quadrat

Gegenüber 7.2.10 (ohne Gebrochene Blocks) hat sich ED marginal verbessert. NB hat sich aber sprunghaft von $max\_row\_neighbor = 3\ auf \ldots = 2$ verbessert. Die Zeilen-Spalten-Balan–cierung ist weiterhin ideal (Abb. 46).

In anderen Konstellationen Lateinischer Quadrate bieten sich für Gebrochene Blocks Raster von $Superzeilen \times Superspalten = 2 \times 2\ bei\ v = 4;\ 2 \times 4\ oder\ 4 \times 2\ bei\ v = 8;\ 3 \times 3\ bei\ v = 9$ etc. an. Bei der Konstellation $\mathrm{v\_r\_k\_s} = 9\_9\_9\_9$ konstruiert man mit dem 3×3 - Raster neun Gebrochene Blocks (Aufgabenstellung beim Sudoku-Rätsel) (Abb. 47).

Z	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6	Sp7	Sp8	Sp9
9	1	5	2	9	3	4	6	8	7
8	6	8	4	5	7	1	9	2	3
7	3	7	9	8	6	2	1	4	5
6	4	2	6	1	8	5	3	7	9
5	8	3	5	7	2	9	4	6	1
4	9	1	7	6	4	3	2	5	8
3	5	6	3	2	1	7	8	9	4
2	7	4	8	3	9	6	5	1	2
1	2	9	1	4	5	8	7	3	6

Abbildung 47: Randomisation in einem Lateinischen Quadrat mit $3 \times 3 = 9$ Gebrochen Blocks (Sudoku)

7.3 Komplexe Latinisierung in der Konstellation: Zeile ≠ vollständige Wiederholung

7.3.1 Balancierte Zeilengruppen

Der Fall einer Latinisierung ausschließlich in Zeilengruppen wird an der Konstellation v_r_k_s = 22_3_6_11 vorgestellt. Nur Zeilengruppen, bestehend aus je zwei Zeilen, sind in dieser Konstellation vollständig besetzt und gleichzeitig rechteckig. Sie werden durch die Option O31 gewählt. Für die Anzahl Zeilengruppen wird die interne auto-Funktion genutzt - das führt hier zu drei Zeilengruppen entsprechend drei Wiederholungen. Für die Zeilengruppen-Varianz wird ebenso der auto-Default genutzt (Abb. 48).

Für die Wiederholungs-Codierung kann entsprechen dem Default die Option O51 angeschaltet bleiben.

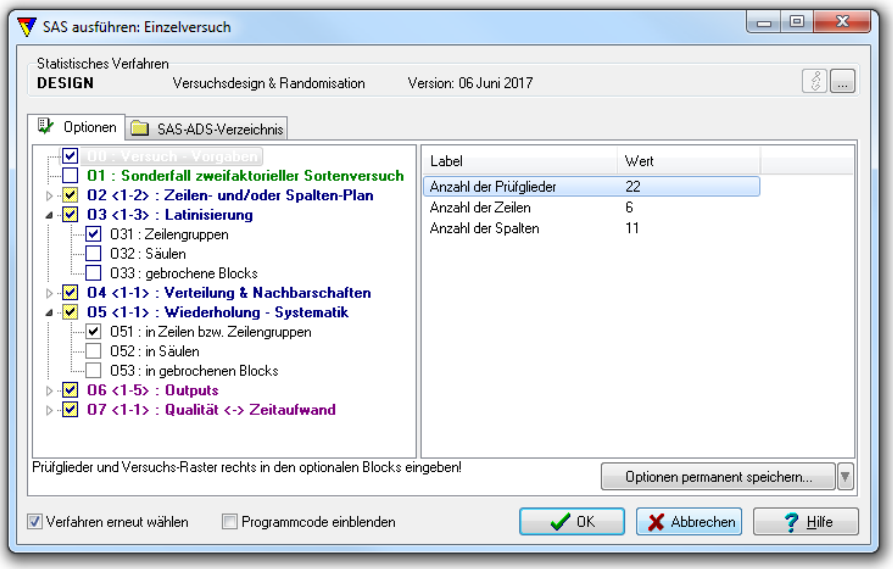


Abbildung 48: Design-Vorgaben für einen Zeilen-Spalten-Plan mit Zeilengruppen

Tabelle 16: Protokollauszug mit Vorgaben für einen Zeilen-Spalten-Plan mit Zeilengruppen

Bereich	Bezeichnung	Vorgabe
Parametereingabe	Anzahl Prüfglieder	22
	Anzahl Zeilen	6
	Anzahl Spalten	11
	Zeilenbalancierung	ja
	Spaltenbalancierung	ja
	Anzahl Zeilengruppen	auto / 3
Varianzen	Zeilengruppen	auto / 1
WDH - Systematik	Sortierung nach Variante	Zeile bzw. Zeilengruppe

Zeilengruppen im Plan

Z	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6	Sp7	Sp8	Sp9	Sp10	Sp11
6	3	3	3	3	3	3	3	3	3	3	3
5	3	3	3	3	3	3	3	3	3	3	3
4	2	2	2	2	2	2	2	2	2	2	2
3	2	2	2	2	2	2	2	2	2	2	2
2	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1

Plan 1 mit Prüfglied-Kürzel

Z	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6	Sp7	Sp8	Sp9	Sp10	Sp11
6	9.3	12.3	14.3	15.3	13.3	22.3	16.3	2.3	6.3	5.3	11.3
5	4.3	8.3	20.3	3.3	10.3	17.3	21.3	19.3	7.3	18.3	1.3
4	1.2	6.2	21.2	9.2	14.2	7.2	13.2	8.2	16.2	20.2	10.2
3	12.2	19.2	15.2	4.2	22.2	3.2	2.2	5.2	17.2	11.2	18.2
2	8.1	11.1	12.1	10.1	5.1	21.1	22.1	1.1	2.1	7.1	15.1
1	17.1	14.1	4.1	6.1	3.1	13.1	19.1	16.1	18.1	9.1	20.1

Plan_1 | Plan_2 | Plan_3 | Zeilengruppen | Masszahlen | Dataset | Prüfglieder | Protokoll

Run ID	Design ID	ED_ weighted	max_row_ neighbor	n_max_row_ neighbor	f_bar_A	trt_1_ED
18	1	2,6632	1	60	,7735	3
25	2	2,6632	1	60	,7638	4

... DesignID 3 bis 24 aus Darstellung herausgenommen.

13	25	2,3217	1	60	,7526	2

Die Spalte ‚trt_1_ED' liefert in der ersten Zeile dasjenige Prüfglied, welches im Tabellenreiter ‚Prüfglied' den geringsten Wert in der Kennzahl ‚ED' erbringt. Der Parameter ‚trt_1_ED' wird im Tabellenreiter ‚Maßzahlen so nicht bereitgestellt, er ist hier nachträglich eingefügt.

Abbildung 49: Blockstruktur, Randomisation und Maßzahlen bei einem Zeilen-Spalten-Plan mit Zeilengruppen

Die Querverteilung von Prüfglied 3 erscheint aus prophylaktischer Sicht unbefriedigend. Eine zusätzliche Latinisierung durch Säulen in Querausrichtung wäre förderlich (siehe 7.3.4), und zwar unabhängig davon, dass Säulen in diesem Falle nicht zugleich rechteckig und vollständig besetzt sein können (Abb. 49).

7.3.2 Balancierte Säulen / Spaltengruppen

Der Fall einer Latinisierung ausschließlich in Säulen wird an der Konstellation $v\_r\_k\_s = 15\_3\_5\_9$ vorgestellt. Nur Säulen sind in dieser Konstellation vollständig besetzt und gleichzeitig rechteckig. Sie werden durch die Option O32 gewählt. Für die Anzahl Säulen wird die interne auto-Funktion genutzt - das führt hier zu drei Säulen entsprechend drei Wiederholungen. Für die Säulen-Varianz wird ebenso der auto-Default genutzt (Abb. 50).

Für die Wiederholungs-Codierung sollte auf Option O52 umgeschaltet werden.

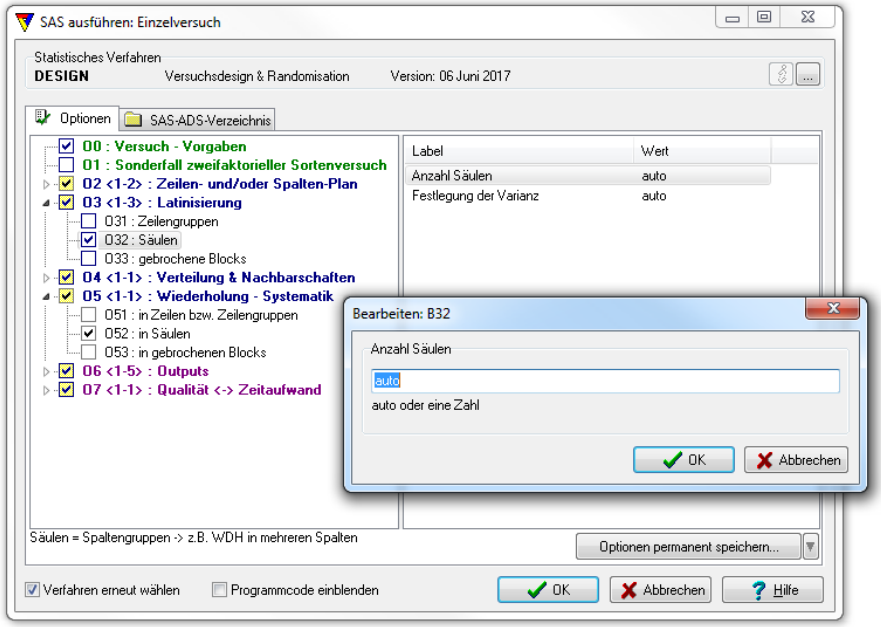


Abbildung 50: Design-Vorgaben für einen Zeilen-Spalten-Plan mit Säulen

Tabelle 17: Protokollauszug mit Vorgaben für einen Zeilen-Spalten-Plan mit Säulen

Bereich	Bezeichnung	Vorgabe
Parametereingabe	Anzahl Prüfglieder	15
	Anzahl Zeilen	5
	Anzahl Spalten	9
	Zeilenbalancierung	ja
	Spaltenbalancierung	ja
	Anzahl Säulen	auto / 3
Varianzen	Säulen	auto / 1
WDH - Systematik	Sortierung nach Variante	Säule

Säulen im Plan

Z	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6	Sp7	Sp8	Sp9
5	1	1	1	2	2	2	3	3	3
4	1	1	1	2	2	2	3	3	3
3	1	1	1	2	2	2	3	3	3
2	1	1	1	2	2	2	3	3	3
1	1	1	1	2	2	2	3	3	3

Plan 1 mit Prüfglied-Kürzel

Z	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6	Sp7	Sp8	Sp9
5	3.1	6.1	8.1	2.2	14.2	11.2	10.3	4.3	1.3
4	9.1	11.1	4.1	15.2	10.2	12.2	3.3	2.3	6.3
3	12.1	1.1	2.1	13.2	6.2	7.2	11.3	5.3	15.3
2	5.1	7.1	15.1	9.2	1.2	8.2	12.3	13.3	14.3
1	14.1	10.1	13.1	4.2	5.2	3.2	7.3	9.3	8.3

Plan_1 | Plan_2 | Plan_3 | **Säulen** | Masszahlen | Dataset | Prüfglieder | Protokoll

Run ID	Design ID	ED_ weighted	max_row_ neighbor	n_max_ row_neighbor	f_bar_A	trt_1_ED
1	1	3,2579	1	40	,6663	2
4	2	3,2579	1	40	,5626	6
... DesignID 3 bis 9 aus Darstellung herausgenommen.						
9	10	2,8023	1	40	,6506	10

Die Spalte ‚trt_1_ED' liefert in der ersten Zeile dasjenige Prüfglied, welches im Tabellenreiter ‚Prüfglied' den geringsten Wert in der Kennzahl ‚ED' erbringt. Der Parameter ‚trt_1_ED' wird im Tabellenreiter ‚Maßzahlen so nicht bereitgestellt, er ist hier nachträglich eingefügt.

Abbildung 51: Blockstruktur, Randomisation und Maßzahlen bei einem Zeilen-Spalten-Plan mit Säulen

Die vertikale Verteilung von Prüfglied 2 erscheint aus prophylaktischer Sicht unbefriedigend. Eine zusätzliche Latinisierung z.B. durch Zeilengruppen könnte förderlich sein (siehe 7.3.4), und zwar unabhängig davon, dass Zeilengruppen in diesem Falle nicht zugleich rechteckig und vollständig besetzt sein können (Abb. 51).

Die beiden Best-Designs unterscheiden sich nicht in ED und in der Nachbarschafts-Balancierung. Dann greift nach aktuellem Gewichtungsprinzip die Sortierung nach f_bar_A, also der Zeilen-Spalten-Balancierung. In diesem Parameter ist Design_1 deutlich besser als Design_2. In einer für 2018 geplanten Verfeinerung der Gewichtungsfunktionen würde diese Rangfolge auch dann erhalten bleiben, wenn es vergleichsweise nur marginale Vorteile bei ED von Run_4 (jetzt Design_2) gegenüber Run_1 (jetzt Design_1) gäbe.

7.3.3 Balancierte Gebrochene Blocks

Der Fall einer Latinisierung ausschließlich in Gebrochenen Blocks wird an der Konstellation v_r_k_s = 15_4_6_10 vorgestellt. Nur Gebrochene Blocks sind in dieser Konstellation vollständig besetzt und gleichzeitig rechteckig. Sie werden durch die Option O33 gewählt. Für die Anzahl Gebrochener Blocks wird die interne auto-Funktion genutzt - das führt hier zu vier Blocks entsprechend vier Wiederholungen und ergibt vier Quadranten (Abb. 52).

Für die Block-Varianz wird ebenso der auto-Default genutzt. Für die Wiederholungs-Codierung sollte auf Option O53 umgeschaltet werden.

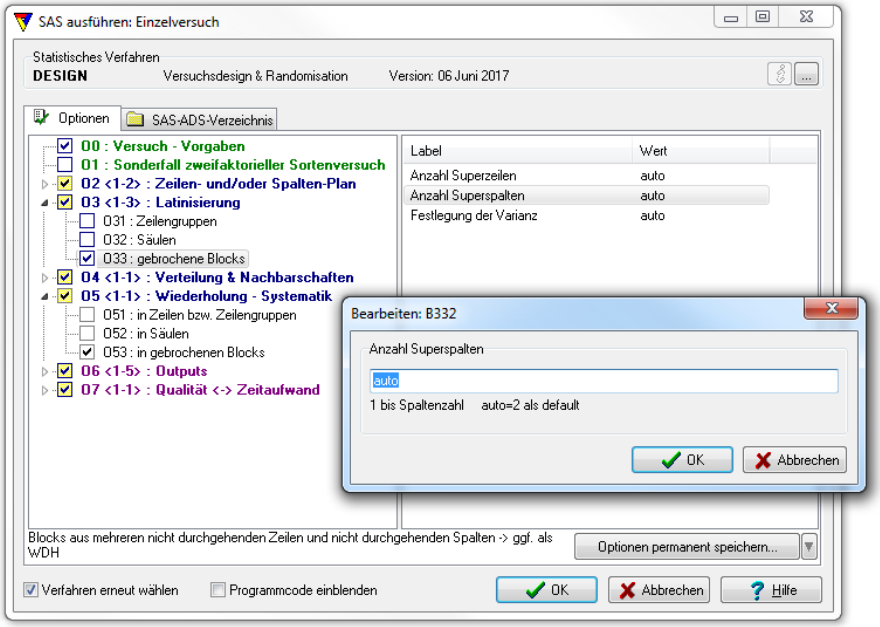
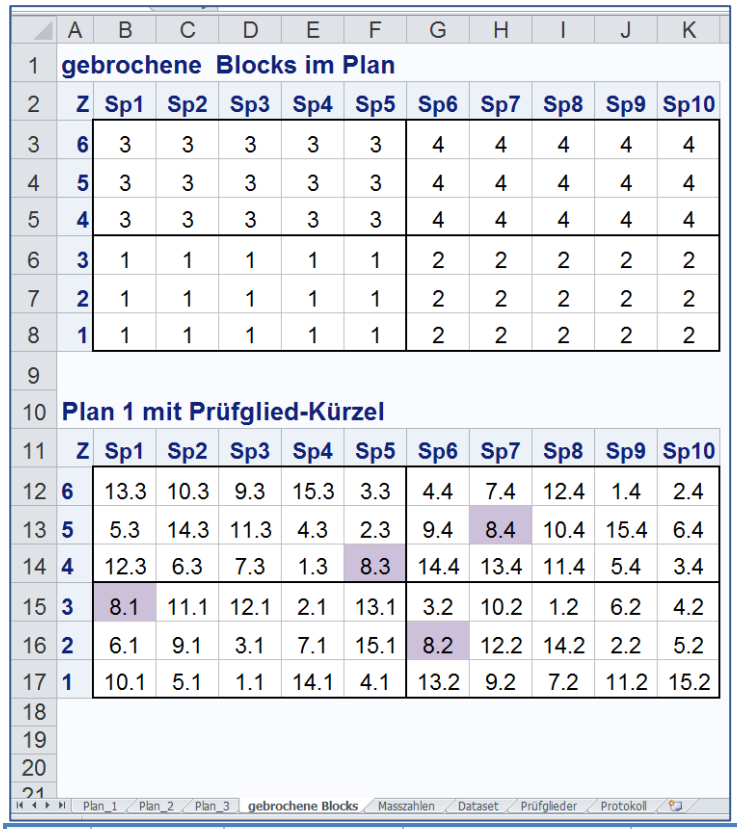


Abbildung 52: Design-Vorgaben für einen Zeilen-Spalten-Plan mit Gebrochenen Blocks

Tabelle 18: Protokollauszug mit Vorgaben für einen Zeilen-Spalten-Plan mit Gebrochenen Blocks

Bereich	Bezeichnung	Vorgabe
Parametereingabe	Anzahl Prüfglieder	15
	Anzahl Zeilen	6
	Anzahl Spalten	10
	Zeilenbalancierung	ja
	Spaltenbalancierung	ja
	Anzahl Superzeilen	auto / 2
	Anzahl Superspalten	auto / 2
Varianzen	gebrochene Blocks	auto / 1
WDH - Systematik	Sortierung nach Variante	gebrochener Block



gebrochene Blocks im Plan

Z	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6	Sp7	Sp8	Sp9	Sp10
6	3	3	3	3	3	4	4	4	4	4
5	3	3	3	3	3	4	4	4	4	4
4	3	3	3	3	3	4	4	4	4	4
3	1	1	1	1	1	2	2	2	2	2
2	1	1	1	1	1	2	2	2	2	2
1	1	1	1	1	1	2	2	2	2	2

Plan 1 mit Prüfglied-Kürzel

Z	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6	Sp7	Sp8	Sp9	Sp10
6	13.3	10.3	9.3	15.3	3.3	4.4	7.4	12.4	1.4	2.4
5	5.3	14.3	11.3	4.3	2.3	9.4	8.4	10.4	15.4	6.4
4	12.3	6.3	7.3	1.3	8.3	14.4	13.4	11.4	5.4	3.4
3	8.1	11.1	12.1	2.1	13.1	3.2	10.2	1.2	6.2	4.2
2	6.1	9.1	3.1	7.1	15.1	8.2	12.2	14.2	2.2	5.2
1	10.1	5.1	1.1	14.1	4.1	13.2	9.2	7.2	11.2	15.2

Plan_1 | Plan_2 | Plan_3 | gebrochene Blocks | Masszahlen | Dataset | Prüfglieder | Protokoll

Run ID	Design ID	ED_ weighted	max_row_ neighbor	n_max_ row_neighbor	f_bar_A	trt_1_ED
4	1	2,8337	1	54	,8223	8
1	2	2,7231	1	54	,8294	2
... DesignID 3 bis 9 aus Darstellung herausgenommen.						
7	10	2,3396	2	1	,8254	9

Die Spalte ‚trt_1_ED' liefert in der ersten Zeile dasjenige Prüfglied, welches im Tabellenreiter ‚Prüfglied' den geringsten Wert in der Kennzahl ‚ED' erbringt. Der Parameter ‚trt_1_ED' wird im Tabellenreiter ‚Maßzahlen so nicht bereitgestellt, er ist hier nachträglich eingefügt.

Abbildung 53: Blockstruktur, Randomisation und Maßzahlen bei einem Zeilen-Spalten-Plan mit Gebrochenen Blocks

Die zweidimensionale Verteilung von Prüfglied 8 erscheint aus prophylaktischer Sicht etwas unbefriedigend. Eine zusätzliche Latinisierung durch Zeilen- und/oder Spaltengruppen wäre förderlich (s. 7.3.5) - und zwar unabhängig davon, dass diese beiden Blockungen in diesem Falle nicht zugleich rechteckig und vollständig besetzt sein können. Mit ausreichender Anzahl Runs gelingt es, doppelte Nachbarschaften von Prüfgliedpaaren zu vermeiden (Abb. 53).

Lassen sich keine rechteckigen balancierten Gebrochenen Blocks erzeugen, so ist i.d.R. der Verzicht auf diese Latinisierung oder - trotz Unbalanciertheit - die Nutzung der ‚rechteckigen' Blockform zu bevorzugen.

Wenn bei krummen Gebrochenen Blocks in einzelnen Zeilen oder Spalten mehr als 2 Blockungsstufen entstehen (siehe Tabellenreiter ‚Gebrochene Blocks', ist prinzipiell von gebrochenen Blocks abzuraten.

7.3.4 Gerechter balancierter Plan

Es gibt Konstellationen, bei denen zwei oder sogar drei Latinisierungen gleichzeitig erfolgen können und bei denen diese jeweils vollständig besetzte und gleichzeitig rechteckige Blocks ergeben. Derartige Konstrukte wurden als *Gerechte Pläne* bzw. *Gerechte Designs* publiziert.

So ein Fall wird an der Konstellation $v\_r\_k\_s = 24\_4\_8\_12$ vorgestellt. Hier werden die Optionen O31, O32 und O33 gemeinsam gewählt. Für die Anzahl Blocks wird jeweils die interne auto-Funktion genutzt - das führt hier zu jeweils vier Blocks entsprechend vier Wiederholungen. Für die Block-Varianzen werden ebenso die auto-Defaults genutzt (Abb. 47). Für die Wiederholungs-Codierung kann entsprechen dem Default die Option O51 angeschaltet bleiben oder in diesem Fall auch nach persönlicher Präferenz beliebig geändert werden.

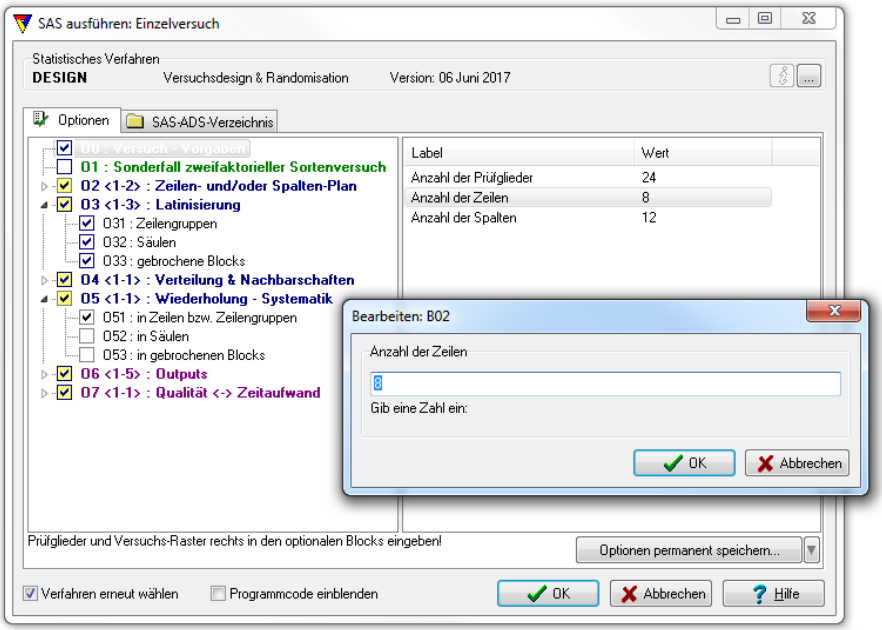
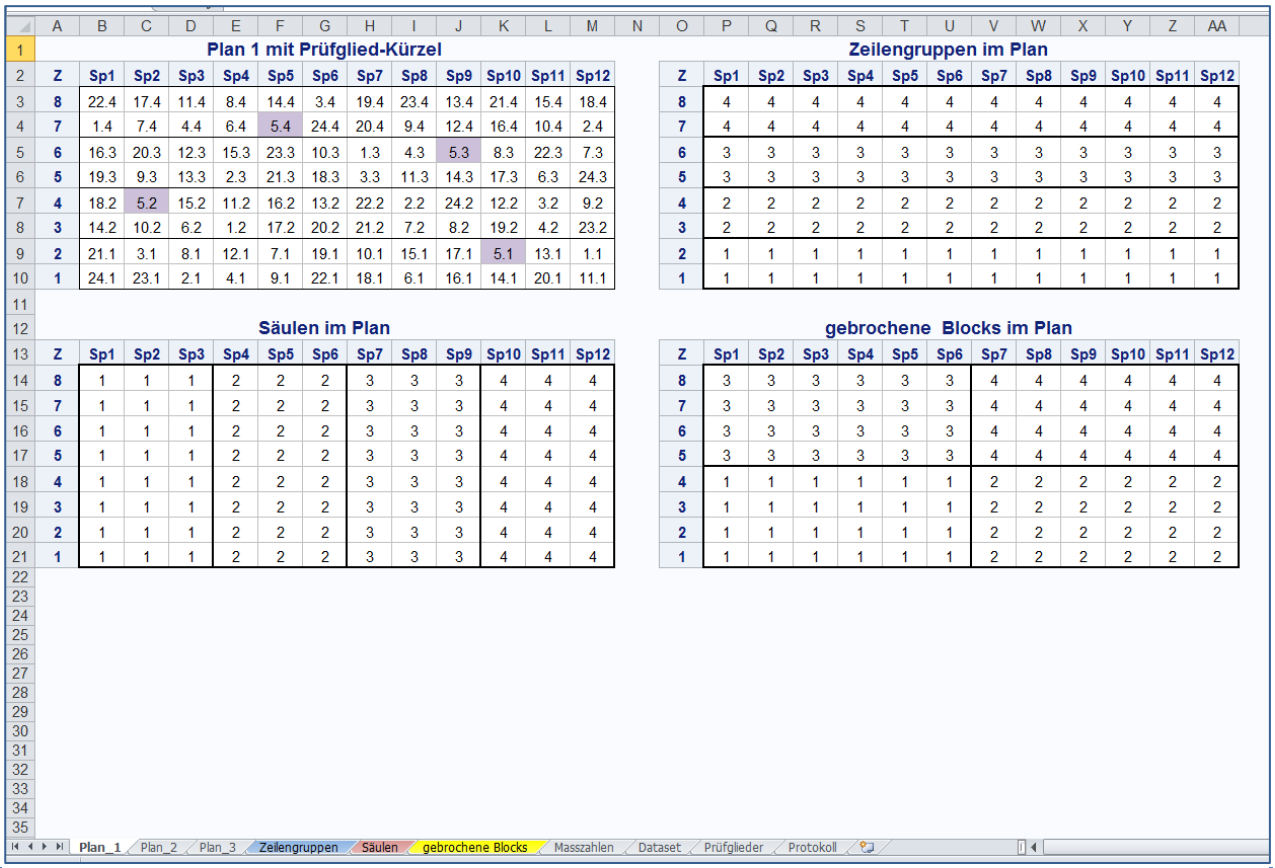


Abbildung 54: Design-Vorgaben für Gerechte balancierte Pläne

Tabelle 19: Protokollauszug mit Vorgaben für einen Gerechten balancierten Plan

Bereich	Bezeichnung	Vorgabe
Parametereingabe	Anzahl Prüfglieder	24
	Anzahl Zeilen	8
	Anzahl Spalten	12
	Zeilenbalancierung	ja
	Spaltenbalancierung	ja
	Anzahl Zeilengruppen	auto / 4
	Anzahl Säulen	auto / 4
	Anzahl Superzeilen	auto / 2
	Anzahl Superspalten	auto / 2
Varianzen	Zeilen	auto / 0.021
	Spalten	auto / 0.008
	Zeilengruppen	auto / 1
	Säulen	auto / 1
	gebrochene Blocks	auto / 1
	räumlicher Kovarianz-Parameter - zweidimensional	auto / 0.0375
	Kovarianz-Parameter zwischen Zeilen-Nachbarn	auto / 0.12
WDH - Systematik	Sortierung nach Variante	Zeile bzw. Zeilengruppe



Plan 1 mit Prüfglied-Kürzel

Z	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6	Sp7	Sp8	Sp9	Sp10	Sp11	Sp12
8	22.4	17.4	11.4	8.4	14.4	3.4	19.4	23.4	13.4	21.4	15.4	18.4
7	1.4	7.4	4.4	6.4	5.4	24.4	20.4	9.4	12.4	16.4	10.4	2.4
6	16.3	20.3	12.3	15.3	23.3	10.3	1.3	4.3	5.3	8.3	22.3	7.3
5	19.3	9.3	13.3	2.3	21.3	18.3	3.3	11.3	14.3	17.3	6.3	24.3
4	18.2	5.2	15.2	11.2	16.2	13.2	22.2	2.2	24.2	12.2	3.2	9.2
3	14.2	10.2	6.2	1.2	17.2	20.2	21.2	7.2	8.2	19.2	4.2	23.2
2	21.1	3.1	8.1	12.1	7.1	19.1	10.1	15.1	17.1	5.1	13.1	1.1
1	24.1	23.1	2.1	4.1	9.1	22.1	18.1	6.1	16.1	14.1	20.1	11.1

Zeilengruppen im Plan

Z	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6	Sp7	Sp8	Sp9	Sp10	Sp11	Sp12
8	4	4	4	4	4	4	4	4	4	4	4	4
7	4	4	4	4	4	4	4	4	4	4	4	4
6	3	3	3	3	3	3	3	3	3	3	3	3
5	3	3	3	3	3	3	3	3	3	3	3	3
4	2	2	2	2	2	2	2	2	2	2	2	2
3	2	2	2	2	2	2	2	2	2	2	2	2
2	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1

Säulen im Plan

Z	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6	Sp7	Sp8	Sp9	Sp10	Sp11	Sp12
8	1	1	1	2	2	2	3	3	3	4	4	4
7	1	1	1	2	2	2	3	3	3	4	4	4
6	1	1	1	2	2	2	3	3	3	4	4	4
5	1	1	1	2	2	2	3	3	3	4	4	4
4	1	1	1	2	2	2	3	3	3	4	4	4
3	1	1	1	2	2	2	3	3	3	4	4	4
2	1	1	1	2	2	2	3	3	3	4	4	4
1	1	1	1	2	2	2	3	3	3	4	4	4

gebrochene Blocks im Plan

Z	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6	Sp7	Sp8	Sp9	Sp10	Sp11	Sp12
8	3	3	3	3	3	3	4	4	4	4	4	4
7	3	3	3	3	3	3	4	4	4	4	4	4
6	3	3	3	3	3	3	4	4	4	4	4	4
5	3	3	3	3	3	3	4	4	4	4	4	4
4	1	1	1	1	1	1	2	2	2	2	2	2
3	1	1	1	1	1	1	2	2	2	2	2	2
2	1	1	1	1	1	1	2	2	2	2	2	2
1	1	1	1	1	1	1	2	2	2	2	2	2

Plan_1 | Plan_2 | Plan_3 | Zeilengruppen | Säulen | gebrochene Blocks | Masszahlen | Dataset | Prüfglieder | Protokoll

Run ID	Design ID	ED_ weighted	max_row_neighbor	n_max_row_neighbor	f_bar_A	trt_1_ED
3	1	4,3385	1	88	,8097	5
13	2	4,2809	1	88	,8178	10
... DesignID 3 bis 24 aus Darstellung herausgenommen.						
25	25	3,7470	1	88	,8029	18

Die Spalte ‚trt_1_ED' liefert in der ersten Zeile dasjenige Prüfglied, welches im Tabellenreiter ‚Prüfglied' den geringsten Wert in der Kennzahl ‚ED' erbringt. Der Parameter ‚trt_1_ED' wird im Tabellenreiter ‚Maßzahlen so nicht bereitgestellt, er ist hier nachträglich eingefügt.

Abbildung 55: Blockstruktur, Randomisation und Maßzahlen bei einem Gerechten balancierten Plan

Prüfglied 5 hat rein numerisch die ‚schlechteste' Verteilung. Dieses Prüfglied fehlt in 3 Außenzeilen und in den beiden Außenspalten. Es sind aber keine tatsächlich relevanten Schwächen mehr erkennbar. Die Dreifach-Latinisierung zuzüglich Nutzung des Kovarianzparameters für die räumliche Verteilung führt in diesem Fall zu besonders gleichmäßigen Verteilungen aller Prüfglieder. Allerdings gibt es noch gewisse Unterschiede zwischen dem besten (4,3) und dem schlechtesten (3,7) von 25 Runs. Auch die Nachbarschaftsbalancierung ist optimal (Abb. 55).

Bei Balanciertheit mehrerer Latinisierungen bestehen gute Aussichten, dass sich diese positiv auf die gleichmäßige Verteilung auswirken.

Aber auch bei Balanciertheit kann Mehrfach-Latinisierung andere Designziele so beeinträchtigen, dass eine Gesamtbewertung negativ ausfällt. Ein Szenarienvergleich zu weniger komplexen Latinisierungen ist anzuraten. Insbesondere sollte geprüft werden, ob die Minimierung wiederholter Nachbarschaften hinreichend erzielt wird und ob bei kleinen Versuchen die Spaltenbalance gut ist.

7.3.5 Gerechter unbalancierter Plan

Der Fall einer Dreifach-Latinisierung trotz Nichterfüllung der Bedingung *vollständig besetzt* und gleichzeitig *rechteckig* wird wie in 7.3.3 an der Konstellation $v\_r\_k\_s = 15\_4\_6\_10$ vorgestellt. Nur Gebrochene Blocks sind in dieser Konstellation vollständig besetzt und gleichzeitig rechteckig, für Säulen und Zeilengruppen entstehen bei der hier vorgestellten Wahl in Option O3 'krumme' Blocks (Abb. 56).

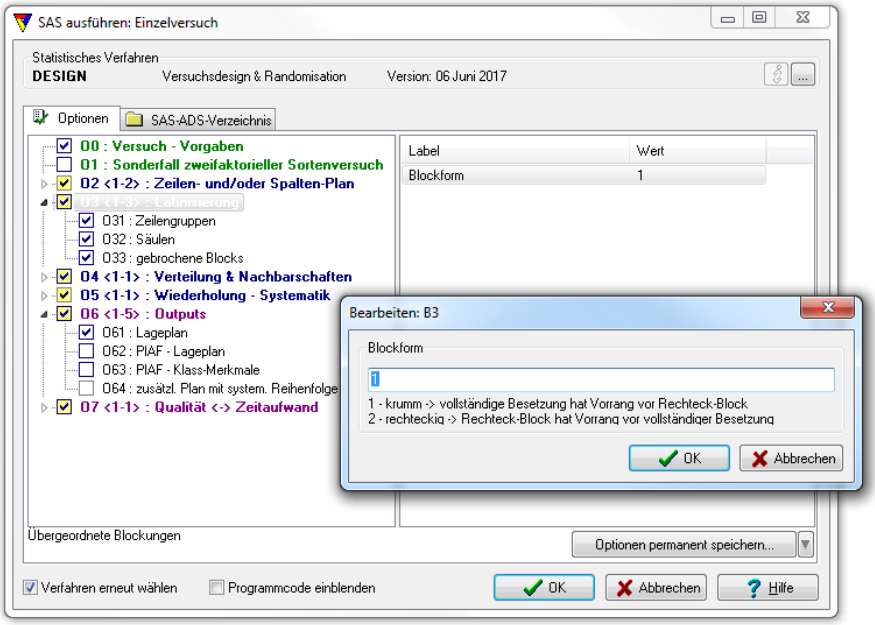


Abbildung 56: Design-Vorgaben für Gerechte unbalancierte Pläne

Tabelle 20: Protokollauszug mit Vorgaben für einen Gerechten unbalancierten Plan

Bereich	Bezeichnung	Vorgabe
Parametereingabe	Anzahl Prüfglieder	15
	Anzahl Zeilen	6
	Anzahl Spalten	10
	Zeilenbalancierung	ja
	Spaltenbalancierung	ja
	Anzahl Zeilengruppen	auto / 4
	Anzahl Säulen	auto / 4
	Anzahl Superzeilen	auto / 2
	Anzahl Superspalten	auto / 2
	Blockform	'krumm'
Varianzen	Zeilen	auto / 0.036
	Spalten	auto / 0.011
	Zeilengruppen	auto / 1
	Säulen	auto / 1
	gebrochene Blocks	auto / 1
WDH - Systematik	Sortierung nach Variante	Zeile bzw. Zeilengruppe

Plan 1 mit Prüfglied-Kürzel

Z	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6	Sp7	Sp8	Sp9	Sp10
6	7.4	3.4	2.4	4.4	5.4	14.4	15.4	9.4	10.4	12.4
5	12.3	9.3	14.3	10.3	15.3	6.4	13.4	11.4	8.4	1.4
4	13.3	1.3	11.3	6.3	8.3	2.3	7.3	4.3	3.3	5.3
3	8.2	14.2	4.2	13.2	9.2	5.2	1.2	10.2	7.2	6.2
2	5.1	10.1	6.1	1.1	7.1	12.2	11.2	3.2	15.2	2.2
1	2.1	11.1	15.1	12.1	3.1	9.1	4.1	8.1	13.1	14.1

Zeilengruppen im Plan

Z	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6	Sp7	Sp8	Sp9	Sp10
6	4	4	4	4	4	4	4	4	4	4
5	3	3	3	3	3	4	4	4	4	4
4	3	3	3	3	3	3	3	3	3	3
3	2	2	2	2	2	2	2	2	2	2
2	1	1	1	1	1	2	2	2	2	2
1	1	1	1	1	1	1	1	1	1	1

Säulen im Plan

Z	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6	Sp7	Sp8	Sp9	Sp10
6	1	1	2	2	2	3	3	4	4	4
5	1	1	2	2	2	3	3	4	4	4
4	1	1	2	2	2	3	3	4	4	4
3	1	1	1	2	2	3	3	3	4	4
2	1	1	1	2	2	3	3	3	4	4
1	1	1	1	2	2	3	3	3	4	4

gebrochene Blocks im Plan

Z	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6	Sp7	Sp8	Sp9	Sp10
6	3	3	3	3	3	4	4	4	4	4
5	3	3	3	3	3	4	4	4	4	4
4	3	3	3	3	3	4	4	4	4	4
3	1	1	1	1	1	2	2	2	2	2
2	1	1	1	1	1	2	2	2	2	2
1	1	1	1	1	1	2	2	2	2	2

Run ID	Design ID	ED_ weighted	max_row _neighbor	n_max_ row_neighbor	f_bar_A	trt_1_ED
6	1	3,0034	1	54	,8280	6
2	2	3,3372	2	4	,8142	3
... DesignID 3 bis 9 aus Darstellung herausgenommen.						
10	10	2,87158	2	4	,7739	2

Die Spalte ‚trt_1_ED' liefert in der ersten Zeile dasjenige Prüfglied, welches im Tabellenreiter ‚Prüfglied' den geringsten Wert in der Kennzahl ‚ED' erbringt. Der Parameter ‚trt_1_ED' wird im Tabellenreiter ‚Maßzahlen so nicht bereitgestellt, er ist hier nachträglich eingefügt.

Abbildung 57: Blockstruktur, Randomisation und Maßzahlen bei einem Gerechten unbalancierten Plan

Prüfglied 6 hat rein numerisch die ‚schlechteste' Verteilung. Dieses Prüfglied fehlt in 2 zusammenhängenden Außenspalten und in den beiden Außenzeilen. Es sind aber keine hochrelevanten Schwächen mehr erkennbar. Die Dreifach-Latinisierung zuzüglich Nutzung des Kovarianzparameters für die räumliche Verteilung führt zu verbesserter und stabilerer Verteilungen der Prüfglieder im Vergleich zum Szenarium in 7.3.3, in dem ausschließlich Gebrochene Blocks genutzt wurden. Allerdings gibt es noch gewisse Unterschiede zwischen dem besten und dem schlechtesten von 10 Runs. Nur der Best-Run erreicht, dass keine doppelten Nachbarschaften auftreten. Das bedeutet, dass Mehrfach-Latinisierung die NB verschlechtern kann, und zwar umso mehr, je geringer die Prüfgliedzahl ist. Bereits der zweitbeste Run führt zu 4 doppelten Nachbarschaften, dafür aber zu einer besseren ED. Bei Verzicht auf Vorgaben zur Nachbarschaftsbalancierung könnte hier die ED um ca. 10% verbessert werden (DesignID=2) (Abb. 57).

In den Spalten 3 und 8 wechselt die Codierung der Säulen. Wenn es Probleme mit der Spaltenbalancierung geben sollte (insbesondere Dopplung eines Prüfgliedes), dann sind solche Spalten die riskanten.

Die Nutzung einer dreifachen Latinisierung ist besonders dann vorsichtig anzugehen, wenn nicht alle Blocktypen balanciert sind. Insbesondere unbalancierte gebrochene Blocks können im Einzelfall verschlechtern statt zu verbessern. Wenn trotzdem Dreifachlatinisierung in Erwägung gezogen wird, ist ein Szenarienvergleich zu weniger komplexen Latinisierungen unter Ausschluss der unbalancierten Blockungen anzuraten.

7.3.6 Gerechter unbalancierter Plan mit individuell angepasster Blockzahl

Das Design in 7.3.5 an der Konstellation $v\_r\_k\_s = 15\_4\_6\_10$ wird nun in der Weise abgewandelt, dass die rechteckige Blockform gewählt wird und dass die Anzahl Zeilengruppen per Hand abgewandelt - auf ‚3' gesetzt wird (Abb. 58).

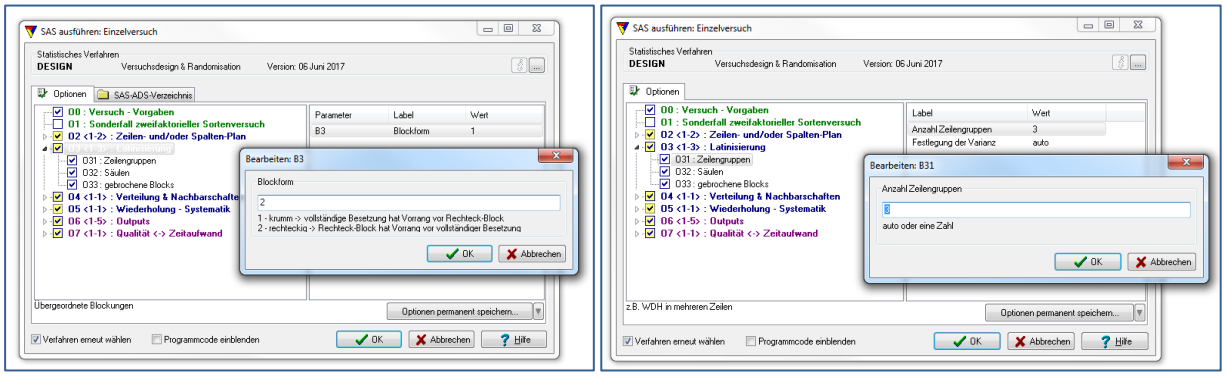


Abbildung 58: Design-Vorgaben für individuell angepasste Blockzahl

Tabelle 21: Protokollauszug mit Vorgaben für einen Plan mit individuell angepasster Blockzahl

Bereich	Bezeichnung	Vorgabe
Parametereingabe	Anzahl Prüfglieder	15
	Anzahl Zeilen	6
	Anzahl Spalten	10
	Zeilenbalancierung	ja
	Spaltenbalancierung	ja
	Anzahl Zeilengruppen	3 / 3
	Anzahl Säulen	auto / 4
	Anzahl Superzeilen	auto / 2
	Anzahl Superspalten	auto / 2
	Blockform	'rechteckig'
Varianzen	Zeilen	auto / 0.036
	Spalten	auto / 0.011
	Zeilengruppen	auto / 1
	Säulen	auto / 1
	gebrochene Blocks	auto / 1
	räumlicher Kovarianz-Parameter - zweidimensional	auto / 0.0375
	Kovarianz-Parameter zwischen Zeilen-Nachbarn	auto / 0.12
	Koeffizient für Gewichtung der Nachbarschaften	0,08
WDH - Systematik	Sortierung nach Variante	gebrochener Block
Qualität & Zeit	gewählter Zeitaufwand:	lang
	ausgeführte Runs	25

Die gleichen drei Zeilengruppen würden auch mit der Einstellung $Blockform = krumm$ erzeugt werden, da 3 Zeilengruppen in diesem Fall keine vollständige Blöcke ergeben können. Dann würden im Unterschied zum vorgestellten Plan nur die Säulen krumm und vollständig besetzt sein.

Plan 1 mit Prüfglied-Kürzel

Z	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6	Sp7	Sp8	Sp9	Sp10
6	1.3	11.3	12.3	9.3	13.3	4.4	8.4	3.4	14.4	5.4
5	6.3	14.3	10.3	8.3	7.3	13.4	11.4	15.4	9.4	2.4
4	3.3	2.3	15.3	5.3	4.3	10.4	12.4	7.4	1.4	6.4
3	11.1	4.1	9.1	1.1	12.1	14.2	2.2	8.2	5.2	13.2
2	13.1	3.1	5.1	10.1	2.1	7.2	15.2	4.2	6.2	12.2
1	7.1	6.1	8.1	15.1	14.1	1.2	3.2	11.2	10.2	9.2

Zeilengruppen im Plan

Z	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6	Sp7	Sp8	Sp9	Sp10
6	3	3	3	3	3	3	3	3	3	3
5	3	3	3	3	3	3	3	3	3	3
4	2	2	2	2	2	2	2	2	2	2
3	2	2	2	2	2	2	2	2	2	2
2	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1

Säulen im Plan

Z	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6	Sp7	Sp8	Sp9	Sp10
6	1	1	1	2	2	3	3	4	4	4
5	1	1	1	2	2	3	3	4	4	4
4	1	1	1	2	2	3	3	4	4	4
3	1	1	1	2	2	3	3	4	4	4
2	1	1	1	2	2	3	3	4	4	4
1	1	1	1	2	2	3	3	4	4	4

gebrochene Blocks im Plan

Z	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6	Sp7	Sp8	Sp9	Sp10
6	3	3	3	3	3	4	4	4	4	4
5	3	3	3	3	3	4	4	4	4	4
4	3	3	3	3	3	4	4	4	4	4
3	1	1	1	1	1	2	2	2	2	2
2	1	1	1	1	1	2	2	2	2	2
1	1	1	1	1	1	2	2	2	2	2

Plan_1 | Plan_2 | Plan_3 | Zeilengruppen | Säulen | gebrochene Blocks | Masszahlen | Dataset | Prüfglieder | Protokoll

Run ID	Design ID	ED_ weighted	max_row_ neighbor	n_max_ row_neighbor	f_bar_A	trt_1_ED
14	1	3,09479	1	54	,82255	4
22	2	2,93849	1	54	,82741	14
... DesignID 3 bis 24 aus Darstellung herausgenommen.						
8	25	2,62615	2	2	,81351	12

Die Spalte ‚trt_1_ED' liefert in der ersten Zeile dasjenige Prüfglied, welches im Tabellenreiter ‚Prüfglied' den geringsten Wert in der Kennzahl ‚ED' erbringt. Der Parameter ‚trt_1_ED' wird im Tabellenreiter ‚Maßzahlen so nicht bereitgestellt, er ist hier nachträglich eingefügt.

Abbildung 59: Blockstruktur, Randomisation und Maßzahlen bei einem Plan mit individuell angepasster Blockzahl

Die Best-Runs in 7.3.5 und 7.3.6 unterscheiden sich nur marginal. Die Ausbeute an bester NB ist in 7.3.6 etwas erhöht - ein Beispiel für den möglichen Vorteil individueller Blockanzahl (Abb. 59).

Potenziell aussichtsreiche Konstellationen für individuell angepasste Blockzahl:

a) wenn die Außenblocks dann überbesetzt sind, also mehr Parzellen als Prüfglieder enthalten; dann ist die Blockanzahl kleiner als der Default, also $\leq r$. **und**

b) die Blöcke dadurch rechteckig statt krumm werden; dann ist die Zeilenzahl ein ganzzahliges Vielfaches der Zeilengruppenzahl bzw. die Spaltenzahl ein ganzzahliges Vielfaches der Spaltengruppenzahl (Säulen) **und**

c) wenn dadurch gleichzeitig vermieden wird, dass einzelne Zeilen- bzw. Spaltengruppen aus nur einer Zeile bzw. Spalte bestehen;

Im gezeigten Beispiel ist weiterhin von Vorteil, dass alle drei Zeilengruppen dann folgende Bedingungen simultan erfüllen: gleichgroß, rechteckig, nicht unterbesetzt und >1 Zeile hoch.

7.3.7 Ungleiche Wiederholungszahl

Es sei zunächst das Design entsprechend 7.3.6 (v_r_k_s = ~~15~~_4_6_10) geplant worden, mit der Abwandlung, dass $Blockform = krumm$ gewählt sei. Angenommen, kurz vor der Aussaat stellt sich heraus, dass ein Prüfglied nicht angelegt werden kann, die Feldplanung aber abgeschlossen ist und Raster und Design nicht mehr geändert werden sollen. Dann kann in Option O0 / B01 die Anzahl Prüfglieder mit 14 überschrieben werden: (v_r_k_s = **14**_4_6_10) (Abb. 60).

Es ergibt sich allerdings bei $\frac{6\times10\ Parzellen}{14\ Prüfglieder} \approx 4{,}3\ Wiederholungen$ keine ganzzahlige Wiederholungsanzahl. Die Prüfglieder können also keine einheitliche Wiederholungszahl erhalten. Letztere wird vom Verfahren ausbalanciert, so dass sie sich um maximal 1 unterscheidet.

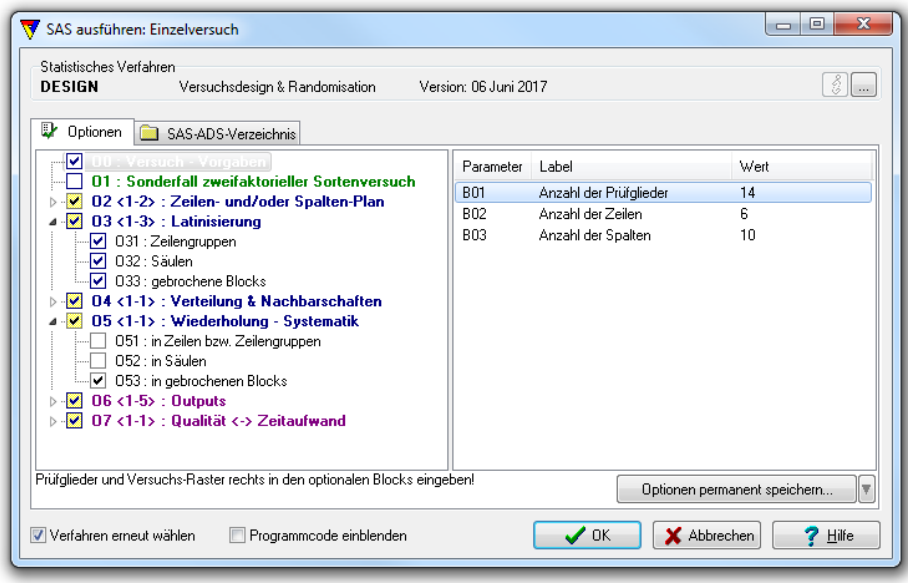


Abbildung 60: Design-Vorgaben bei ungleicher Wiederholungszahl

Tabelle 22: Protokollauszug mit Vorgaben bei ungleicher Wiederholungszahl

Bereich	Bezeichnung	Vorgabe
Parametereingabe	Anzahl Prüfglieder	14
	Anzahl Zeilen	6
	Anzahl Spalten	10
	Zeilenbalancierung	ja
	Spaltenbalancierung	ja
	Anzahl Zeilengruppen	3 / 3
	Anzahl Säulen	auto / 4
	Anzahl Superzeilen	auto / 2
	Anzahl Superspalten	auto / 2
	Blockform	'krumm'
Varianzen	Zeilen	auto / 0.041
	Spalten	auto / 0.013
	Zeilengruppen	auto / 1
	Säulen	auto / 1
	gebrochene Blocks	auto / 1
Priorität	'gleichmäßige Verteilung & Nachbarschaften'	
	räumlicher Kovarianz-Parameter - zweidimensional	auto / 0.0375
	Kovarianz-Parameter zwischen Zeilen-Nachbarn	auto / 0.12

	Koeffizient für Gewichtung der Nachbarschaften	0,08
WDH - Systematik	Sortierung nach Variante	gebrochener Block
	ausgeführte Runs	25

Plan 1 mit Prüfglied-Kürzel

Z	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6	Sp7	Sp8	Sp9	Sp10
6	5.3	7.3	2.3	11.3	10.3	13.4	1.4	12.4	9.5	6.4
5	14.3	1.3	3.3	13.5	9.3	7.4	4.4	2.4	10.4	8.4
4	6.3	13.3	12.3	4.3	8.3	11.4	3.4	5.4	14.4	9.4
3	4.1	6.5	10.1	7.1	13.1	5.5	2.2	9.2	11.2	1.2
2	2.1	12.1	11.1	6.1	14.1	8.2	5.2	10.2	3.2	7.2
1	9.1	3.1	8.1	1.1	5.1	6.2	12.2	14.2	4.2	13.2

Zeilengruppen im Plan

Z	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6	Sp7	Sp8	Sp9	Sp10
6	3	3	3	3	3	3	3	3	3	3
5	3	3	3	3	3	3	3	3	3	3
4	2	2	2	2	2	2	2	2	2	2
3	2	2	2	2	2	2	2	2	2	2
2	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1

Säulen im Plan

Z	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6	Sp7	Sp8	Sp9	Sp10
6	1	1	2	2	2	3	3	4	4	4
5	1	1	2	2	2	3	3	4	4	4
4	1	1	2	2	2	3	3	4	4	4
3	1	1	1	2	2	3	3	3	4	4
2	1	1	1	2	2	3	3	3	4	4
1	1	1	1	2	2	3	3	3	4	4

gebrochene Blocks im Plan

Z	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6	Sp7	Sp8	Sp9	Sp10
6	3	3	3	3	3	4	4	4	4	4
5	3	3	3	3	3	4	4	4	4	4
4	3	3	3	3	3	4	4	4	4	4
3	1	1	1	1	1	2	2	2	2	2
2	1	1	1	1	1	2	2	2	2	2
1	1	1	1	1	1	2	2	2	2	2

Run ID	Design ID	ED_ weighted	max_row_ neighbor	n_max_ row_neighbor	f_bar_A	trt_1_ED
8	1	2,6174	1	54	,8385	5
13	2	3,0927	2	2	,8429	12
12	3	3,0308	2	3	,8377	9
6	4	2,8520	2	1	,8461	12
7	5	3,1483	2	6	,8370	1
... **DesignID** 6 bis 24 aus Darstellung herausgenommen.						
11	25	2,6174	2	5	,8400	4

Die Spalte ‚trt_1_ED' liefert in der ersten Zeile dasjenige Prüfglied, welches im Tabellenreiter ‚Prüfglied' den geringsten Wert in der Kennzahl ‚ED' erbringt. Der Parameter ‚trt_1_ED' wird im Tabellenreiter ‚Maßzahlen so nicht bereitgestellt, er ist hier nachträglich eingefügt.

Abbildung 61: Blockstruktur, Randomisation und Maßzahlen bei ungleicher Wiederholungszahl

Der im Verfahren integrierte Algorithmus vergibt als Wiederholungsnummer zunächst entsprechend der hier gewählten Wdh-Systematik zunächst innerhalb jedes Gebrochenen Blocks für jedes Prüfglied von unten links beginnend Wdh_Nr. = gebrochener_Block_Nr. Danach verbleibt in jedem Gebrochenen Block eine Parzelle, für deren Prüfglied bereits eine Wdh-Nr. vergeben wurde. Diese insgesamt vier Parzellen erhalten als Wdh-Nr. die ‚5'. Dieses Vorgehen ist geeignet für eine übersichtliche Codierung und Datenerfassung in PIAF. Sie ist aber ungeeignet für die Blockung im Auswertungsmodell (insbesondere die fünfte Wdh. stellt keine Einheit dar, für die korrelierte Effekte denkbar sind). Das bedeutet: die Wdh-Nr. ist eine reine Ordnungsnummer, die Blockung erfolgt aufgrund der Latinisierungen zuzüglich Zeilen und Spalten. In diesem Fall ist konkret der Gebrochene Block die bessere/richtige Blockung gegenüber der Wdh (Abb. 61).

Wer zusätzliche Wiederholungen nur für einen Teil der Prüfglieder nicht wünscht, kann diese 4 Parzellen auch als Füllparzellen anlegen. Werden Füllparzellen in einem definierten (Eck-) Bereich gewünscht, so kann dies über die in 2018 vorgesehen Maßnahme ‚Teilsortimente' umsetzen, in welcher definierte Lücken vorpositioniert eingebaut werden können.

Diese vier zusätzlichen Parzellen liefern zum einen Beitrag zur Verminderung des Schätzfehlers dieser vier Prüfglieder. Gleichzeitig verbessern sie aber auch alle weiteren Varianz- und Effektschätzungen und somit sämtliche Prüfgliedvergleiche. Früher wurden in derartigen Situationen häufig Füllparzellen eingebunden, die nicht weiter genutzt wurden, aber Aufwand erzeugten. Die hier vorgestellte Möglichkeit, diese Parzellen trotz Unbalanciertheit voll zu

nutzen, erhöht die Effizienz - ein adäquates Auswertungsmodell vorausgesetzt (EVA und ZVA; s.a. Abschnitt 5).

Nur einer von 25 Runs (RunID(8); DesignID(1)) konnte doppelte Nachbarschaften vermeiden. Dieser ‚Quantensprung' zu $max\_row\_neighbor = 1$ wird p.d. als Best-Run nach oben sortiert, trotz schlechterem ED. Die weiteren 24 Runs erreichen nur $max\_row\_neighbor = 2$. Innerhalb von $max\_row\_neighbor = 2$ erfolgt entsprechend der Vorgabe in Option O4 eine gewichtete Abwägung zwischen $n\_max\_row - neighbor$ und $ED\_weighted$. Unter ihnen sind auch andere sinnvolle Rang-Sortierungen in der DesignID, je nach vorgegebener Priorität zwischen NB und ED, denkbar. Vorgesehen ist, ab 2018 auch die Variation in NB (f_bar_A) in die Wichtungsfunktion einzubeziehen. Die Parameter sind in 3.9.1 (Reiter ‚Maßzahlen', Tab. 1) definiert.

Erwartungsgemäß sind die Prüfglieder mit dem geringsten ED diejenigen, welche eine zusätzliche Wiederholungszahl aufweisen - sie haben zueinander notwendigerweise geringere mittlere Abstände (Tab. 23). Insofern sollten nur die ED-Werte zwischen Prüfgliedern mit gleicher Wdh-Anzahl verglichen und bewertet werden.

Tabelle 23: Maßzahlen im Tabellenreiter ‚Prüfglieder' bei ungleicher Wiederholungszahl

Tabelle Prüfglieder nur für Design-Typ RUN

Plan ID	Run ID	Design ID	trt	NND	NNLD	MST	MSTL	ED	N_MAX_ROW NEIGHBOR	no_of_ reps
1	8	1	1	4,97289	4,92329	4,60460	4,59070	4,6982	1	4
1	8	1	2	4,32477	4,08499	3,90273	3,68403	3,8720	1	4
1	8	1	3	4,33605	4,16430	3,69155	3,63634	3,8229	1	4
1	8	1	4	4,21861	4,05952	3,59890	3,54954	3,7283	1	4
1	8	1	5	2,74592	2,44955	2,92933	2,53400	2,6174	1	5
1	8	1	6	3,34646	2,86474	3,07237	2,59400	2,8191	1	5
1	8	1	7	4,02462	3,90229	3,81133	3,70819	3,7993	1	4
1	8	1	8	3,51335	3,36838	3,46612	3,28210	3,3593	1	4
1	8	1	9	3,59344	3,29978	3,43364	3,17782	3,29450	1	5
1	8	1	10	3,98511	3,92515	3,63031	3,60910	3,7141	1	4
1	8	1	11	3,70802	3,60605	3,19875	3,18302	3,3233	1	4
1	8	1	12	4,47260	4,25278	4,11170	3,84870	4,0445	1	4
1	8	1	13	3,15448	2,94810	3,02334	2,78557	2,9025	1	5
1	8	1	14	4,74643	4,42465	3,77485	3,68403	3,9566	1	4

7.4 Zweifaktorielle Spaltanlage im Spezialfall: Zeile = Großteilstück (LSV, WP ...)

Es sollen beispielhaft 16 Sorten-Stufen in 2 Intensitäts-Stufen in 2 Wiederholungen geprüft werden. Jede Zeile stelle eine Intensitätsstufe dar, in der jede Sorte genau einmal steht. Damit wird aus Sicht des Designs die Zeile zum Großteilstück, der Faktor Intensität zum übergeordneten Großteilstücksfaktor (Randomisationseinschränkung). Insofern kann Design und Randomisation des Kleinteilstücksfaktors ‚Sorte' nach genau den gleichen Prinzipien erfolgen, wie sie sie in 7.2. beschrieben wurden. Die nachfolgend erzeugten Design sind Spaltanlagen, gleichzeitig aber auch Zeilen-Spalten-Pläne, ggf. zusätzlich Lateinische Rechtecke und ggf. gibt es noch weitere Designvorgaben. Darin zeigt sich noch mal, dass es nicht sinnvoll ist, für jede Konstellation eine Bezeichnung der *Versuchsanlage* zu definieren. Vielmehr wird mit den *Designvorgaben* das *Auswertungsmodell* vorgegeben. Mit dem Modell ist das *Versuchsdesign* bzw. in anderem Sprachgebrauch die *Versuchsanlage* oder die *Versuchsmethode* definiert.

Die Zuordnung der Stufennummern des Großteilstücksfaktors und Wiederholungsnummern zu den Zeilen wird in Option O1 in den optionalen Blocks B11 und B12 editiert. Dabei muss vom Nutzer sichergestellt werden, dass in B11 und B12 so viele Codierungen erfolgen, wie in Option O0 im optionalen Block B02 Zeilen vorgegeben wurden. Im Optionalen Block B13 wird vorgegeben, ob der in PIAF benutzte Faktor A oder Faktor B der Großteilstücksfaktor sei. Da dies i.d.R. innerhalb einer Dienststelle immer einheitlich definiert ist, kann jeder Nutzer dies vordefinieren und permanent speichern (s. 3.11). Die Randomisation der Stufennummern des Großteilstücksfaktors erfolgt durch den Nutzer außerhalb dieses Verfahrens und wird dann hier per Hand vorgegeben. Die Weiterentwicklung 2018 sieht reale zweifaktorielle Design-Planungen vor.

7.4.1 Ohne Latinisierung

Das Design für den Kleinteilstücksfaktor ‚Sorte' wird nachfolgend für die oben beschriebene Konstellation v_r_k_s = 16_4_4_16 definiert (Abb. 62).

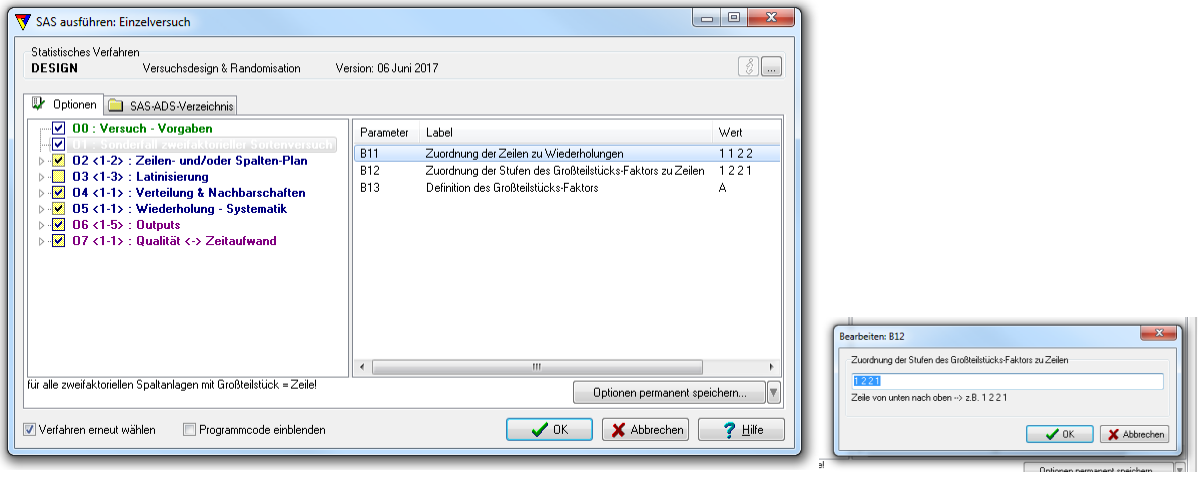


Abbildung 62: Design-Vorgaben für zweifaktorielle Sortenversuche

Tabelle 24: Protokollauszug mit Vorgaben für einen zweifaktoriellen Sortenversuch

Bereich	Bezeichnung	Vorgabe
Parametereingabe	Anzahl Prüfglieder	16
	Anzahl Zeilen	4
	Anzahl Spalten	16
	Zeilenbalancierung	ja
	Spaltenbalancierung	ja
WDH - Systematik	Sortierung nach Variante	Zeile bzw. Zeilengruppe
Sonderfall A/B bzw. B/A	Definition Großteilstück	'A'
	Zeilen zu WDH	1 1 2 2
	Großteilstücksfaktor zu Zeilen	1 2 2 1

Plan 1 mit Prüfglied-Kürzel

Z	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6	Sp7	Sp8	Sp9	Sp10	Sp11	Sp12	Sp13	Sp14	Sp15	Sp16
4	1.12.2	1.11.2	1.3.2	1.16.2	1.6.2	1.7.2	1.9.2	1.15.2	1.13.2	1.1.2	1.5.2	1.4.2	1.2.2	1.8.2	1.14.2	1.10.2
3	2.6.2	2.4.2	2.9.2	2.5.2	2.2.2	2.14.2	2.13.2	2.8.2	2.16.2	2.12.2	2.10.2	2.7.2	2.11.2	2.15.2	2.3.2	2.1.2
2	2.11.1	2.13.1	2.12.1	2.7.1	2.16.1	2.9.1	2.2.1	2.10.1	2.15.1	2.6.1	2.3.1	2.5.1	2.14.1	2.1.1	2.8.1	2.4.1
1	1.4.1	1.3.1	1.2.1	1.15.1	1.14.1	1.6.1	1.1.1	1.11.1	1.5.1	1.13.1	1.7.1	1.8.1	1.12.1	1.9.1	1.10.1	1.16.1

Run ID	Design ID	ED_ weighted	max_row_neighbor	n_max_row_neighbor	f_bar_A	trt_1_ED
2	1	3,2754	1	60	,7544	8
3	2	2,8010	1	60	,7555	2
... **DesignID** 3 bis 9 aus Darstellung herausgenommen.						
8	10	2,2361	1	60	,7542	5

Großteilstückfaktor-Stufe.Kleinteilstücksfaktor-Stufe.Wiederholungs-Nr.
Maßzahlen gültig nur für den Kleinteilstücksfaktor
Die Spalte ‚trt_1_ED' liefert in der ersten Zeile dasjenige Prüfglied, welches im Tabellenreiter ‚Prüfglied' den geringsten Wert in der Kennzahl ‚ED' erbringt. Der Parameter ‚trt_1_ED' wird im Tabellenreiter ‚Maßzahlen so nicht bereitgestellt, er ist hier nachträglich eingefügt.

Abbildung 63: Blockstruktur, Randomisation und Maßzahlen bei ungleicher Wiederholungszahl

Aufgrund fehlender Latinisierung sind einzelne Prüfglieder, insbesondere die ‚8', einseitig verteilt. Hier würde Säulen-Bildung abhelfen (Abb. 63).

7.4.2 Mit Latinisierung in Säulen

Den Designvorgaben in 7.4.1 werden ‚Säulen' hinzugefügt (Abb. 64).

Auch Gebrochene Blocks wären möglich und würden bezüglich ‚Sorte' sogar zu rechteckigen, vollständig besetzen Blocks führen. Aufgrund der übergeordneten Stufenzuordnung für ‚Intensität' führte das dazu, dass in einer Wiederholung (2 Zeilen) jede Sorte in Intensität '1' einmal links, dafür in Intensität '2' einmal rechts steht. Dies käme der Sortenschätzung im Mittel beider Intensitäten entgegen, könnte aber die Schätzung des Intensitätseffektes je Sorte sogar stören. Insofern muss sachlogisch erwogen werden, wo die Prioritäten liegen und wann die Bildung Gebrochener Blocks in Spaltanlagen mit $r \leq 3$ sachlich sogar kontraproduktiv ist.

Zeilengruppen sind in diesem Fall ungeeignet bzw. überflüssig, da bezüglich ‚Sorte' die $Zeilen = Großteilstücke$ bereits vollständig besetzt sind.

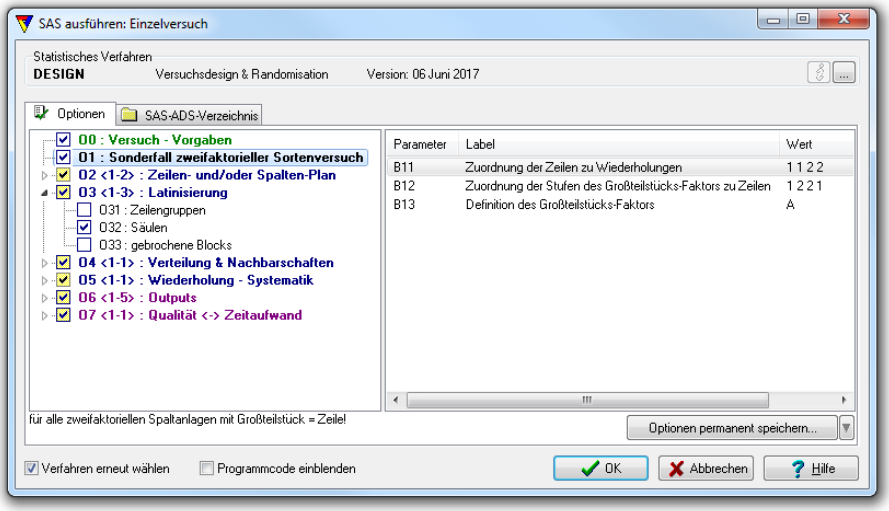


Abbildung 64: Design-Vorgaben für zweifaktorielle Sortenversuche mit Säulen

Tabelle 25: Protokollauszug mit Vorgaben für einen zweifaktoriellen Sortenversuch mit Säulen

Bereich	Bezeichnung	Vorgabe
Parametereingabe	Anzahl Prüfglieder	16
	Anzahl Zeilen	4
	Anzahl Spalten	16
	Zeilenbalancierung	ja
	Spaltenbalancierung	ja
	Anzahl Säulen	auto / 4
	gebrochene Blocks	ohne / 0
WDH - Systematik	Sortierung nach Variante	Zeile bzw. Zeilengruppe
Sonderfall A/B bzw. B/A	Definition Großteilstück	'A'
	Zeilen zu WDH	1 1 2 2
	Großteilstücksfaktor zu Zeilen	1 2 2 1

Plan 1 mit Prüfglied-Kürzel

Z	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6	Sp7	Sp8	Sp9	Sp10	Sp11	Sp12	Sp13	Sp14	Sp15	Sp16
4	1.14.2	1.9.2	1.12.2	1.3.2	1.16.2	1.4.2	1.7.2	1.10.2	1.11.2	1.13.2	1.1.2	1.15.2	1.5.2	1.8.2	1.2.2	1.6.2
3	2.11.2	2.2.2	2.10.2	2.15.2	2.14.2	2.6.2	2.12.2	2.1.2	2.5.2	2.4.2	2.8.2	2.16.2	2.9.2	2.3.2	2.13.2	2.7.2
2	2.7.1	2.5.1	2.13.1	2.16.1	2.11.1	2.9.1	2.8.1	2.15.1	2.12.1	2.2.1	2.3.1	2.6.1	2.10.1	2.4.1	2.14.1	2.1.1
1	1.8.1	1.6.1	1.4.1	1.1.1	1.3.1	1.5.1	1.2.1	1.13.1	1.10.1	1.14.1	1.7.1	1.9.1	1.15.1	1.11.1	1.12.1	1.16.1

Säulen im Plan

Z	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6	Sp7	Sp8	Sp9	Sp10	Sp11	Sp12	Sp13	Sp14	Sp15	Sp16
4	1	1	1	1	2	2	2	2	3	3	3	3	4	4	4	4
3	1	1	1	1	2	2	2	2	3	3	3	3	4	4	4	4
2	1	1	1	1	2	2	2	2	3	3	3	3	4	4	4	4
1	1	1	1	1	2	2	2	2	3	3	3	3	4	4	4	4

RunID	Design ID	ED_ weighted	max_row_ neighbor	n_max_row_neighbor	f_bar_A	trt_1_ED
8	1	4,0331	1	60	,7239	15
7	2	3,9117	1	60	,7324	9
... DesignID 3 bis 9 aus Darstellung herausgenommen.						
2	10	3,6184	1	60	,7345	10

Großteilstückfaktor-Stufe.Kleinteilstücksfaktor-Stufe.Wiederholungs-Nr.
Maßzahlen gültig nur für den Kleinteilstücksfaktor
Die Spalte ‚trt_1_ED' liefert in der ersten Zeile dasjenige Prüfglied, welches im Tabellenreiter ‚Prüfglied' den geringsten Wert in der Kennzahl ‚ED' erbringt. Der Parameter ‚trt_1_ED' wird im Tabellenreiter ‚Maßzahlen so nicht bereitgestellt, er ist hier nachträglich eingefügt.

Abbildung 65: Blockstruktur, Randomisation und Maßzahlen bei ungleicher Wiederholungszahl

Die Verteilung ist durch die Säulenbildung gegenüber 7.4.1 deutlich verbessert, was auch bei der ‚schlechtesten' ED (Prüfglied 15) zu erkennen ist (Abb. 65).

7.4.3 Differenzierte Wiederholungsanzahl für Stufen des Großteilstücksfaktors

Reales Beispiel: Am Versuchsstandort Gülzow der LFA-MV stehen für Sortenversuche fünf Zeilen zur Verfügung, die wir aus Effizienzgründen auch nutzen wollen. Das bedeutet: für eine der beiden Intensitätsstufen können drei Wiederholungen angelegt werden. Da an der ortsüblich geführten Intensitätsstufe ‚2' ein größeres fachliches Interesse in der streuungsanfälligen Ertragsauswertung liegt (Stufe ‚1' dafür eher bei Lager und Krankheiten, die aber heritabler sind und weniger Wiederholungen ‚brauchen'), soll Stufe ‚2' in drei, Stufe ‚1' in zwei Wiederholungen angelegt werden. Ein adäquates Auswertungsmodell steht mit ZVA zur Verfügung. Damit verändert sich gegenüber 7.4.1 und 7.4.2 die Konstellation für den Kleinteilstücksfaktor zu (v_r_k_s = 16_5_5_16) (Abb. 66).

Die Übereinstimmung der Anzahl Zeilen (Großteilstücke) in BO2 mit der Anzahl Einträge in B11 und B12 muss der Nutzer gewährleisten. Es ist zu ersehen, dass die Wiederholung ‚2' nur für die ortsübliche Intensität ‚2' vorgesehen wird.

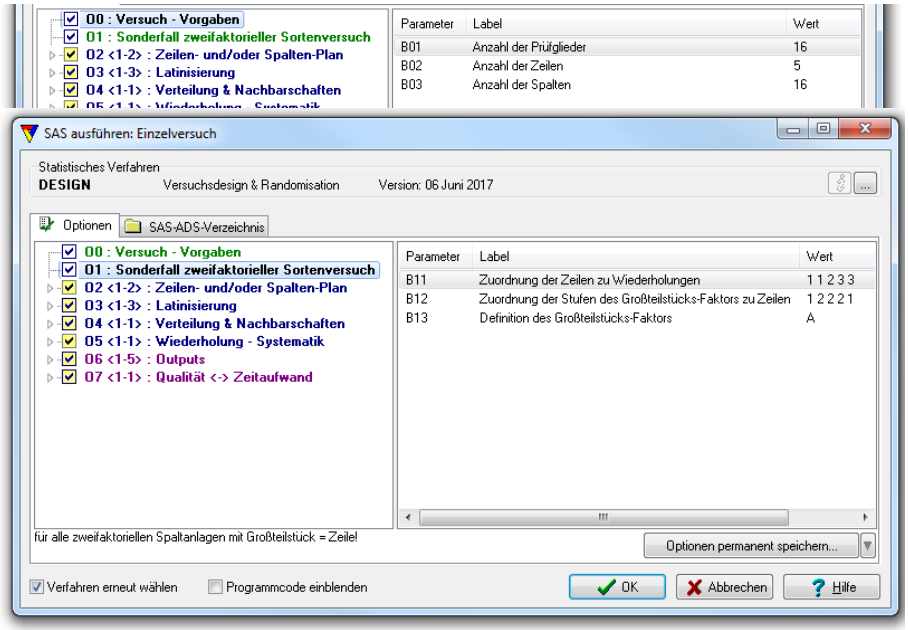


Abbildung 66: Design-Vorgaben für zweifaktorielle Sortenversuche mit ungleicher Wiederholungszahl je Intensitätsstufe

Tabelle 26: Protokollauszug mit Vorgaben für einen zweifaktoriellen Sortenversuch mit ungleicher Wiederholungszahl je Intensitätsstufe

Bereich	Bezeichnung	Vorgabe
Parametereingabe	Anzahl Prüfglieder	16
	Anzahl Zeilen	5
	Anzahl Spalten	16
	Zeilenbalancierung	ja
	Spaltenbalancierung	ja
	Anzahl Säulen	auto / 5
	Blockform	'krumm'
Sonderfall A/B bzw. B/A	Definition Großteilstück	'A'
	Zeilen zu WDH	1 1 2 3 3
	Großteilstücksfaktor zu Zeilen	1 2 2 2 1

Plan 1 mit Prüfglied-Kürzel

Z	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6	Sp7	Sp8	Sp9	Sp10	Sp11	Sp12	Sp13	Sp14	Sp15	Sp16
5	1.8.3	1.5.3	1.7.3	1.10.3	1.14.3	1.15.3	1.1.3	1.13.3	1.9.3	1.16.3	1.4.3	1.12.3	1.6.3	1.3.3	1.11.3	1.2.3
4	2.11.3	2.4.3	2.3.3	2.12.3	2.8.3	2.9.3	2.10.3	2.2.3	2.6.3	2.1.3	2.5.3	2.13.3	2.15.3	2.16.3	2.14.3	2.7.3
3	2.13.2	2.6.2	2.15.2	2.4.2	2.2.2	2.5.2	2.16.2	2.12.2	2.14.2	2.3.2	2.10.2	2.11.2	2.7.2	2.9.2	2.1.2	2.8.2
2	2.2.1	2.9.1	2.12.1	2.7.1	2.16.1	2.1.1	2.3.1	2.15.1	2.11.1	2.5.1	2.6.1	2.14.1	2.8.1	2.10.1	2.4.1	2.13.1
1	1.16.1	1.10.1	1.1.1	1.14.1	1.13.1	1.11.1	1.6.1	1.7.1	1.8.1	1.4.1	1.9.1	1.3.1	1.2.1	1.15.1	1.5.1	1.12.1

Säulen im Plan

Z	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6	Sp7	Sp8	Sp9	Sp10	Sp11	Sp12	Sp13	Sp14	Sp15	Sp16
5	1	1	1	2	2	2	3	3	3	4	4	4	5	5	5	5
4	1	1	1	2	2	2	3	3	3	4	4	4	4	5	5	5
3	1	1	1	2	2	2	3	3	3	3	4	4	4	5	5	5
2	1	1	1	2	2	2	2	3	3	3	4	4	4	5	5	5
1	1	1	1	1	2	2	2	3	3	3	4	4	4	5	5	5

Run ID	Design ID	ED_ weighted	max_row_ neighbor	n_max_row_neighbor	f_bar_A	trt_1_ED
6	1	3,6055	1	75	,8039	16
8	2	3,5630	2	1	,8216	3
... DesignID 3 bis 9 aus Darstellung herausgenommen.						
5	10	3,1727	2	4	,8085	2

Großteilstückfaktor-Stufe.Kleinteilstücksfaktor-Stufe.Wiederholungs-Nr.
Maßzahlen gültig nur für den Kleinteilstücksfaktor
Die Spalte ‚trt_1_ED' liefert in der ersten Zeile dasjenige Prüfglied, welches im Tabellenreiter ‚Prüfglied' den geringsten Wert in der Kennzahl ‚ED' erbringt. Der Parameter ‚trt_1_ED' wird im Tabellenreiter ‚Maßzahlen so nicht bereitgestellt, er ist hier nachträglich eingefügt.

Abbildung 67: Blockstruktur, Randomisation und Maßzahlen bei ungleicher Wiederholungszahl

Nur in einem von 10 Runs gelingt es, mehrfache paarweise Nachbarschaften zu vermeiden und trotzdem eine gute ED zu erzielen. Selbst die numerisch kritischste Sorte ‚16' ist unproblematisch verteilt (Abb. 67).

7.5 Zweifaktorielle Spaltanlage mit nebeneinander liegenden Wiederholungen

Nachfolgendes Beispiel zeigt ein Design, wie es für zweifaktorielle Raps-LSV in Thüringen angestrebt wird. Es stehen nur 4 Zeilen zur Verfügung, aber es sollen 3 Wiederholungen je Sorte × Intensität - Kombination entstehen (Abb. 68). Der Faktor Intensität hat 2 Stufen. Zudem sollen die Intensitäten in Zeilen als Großteilstück angelegt werden. Ein weiteres Ziel für Führungen ist, dass in jeder Zeile jede Sorte mindestens einmal vertreten ist.

Dieses Ziel kann erreicht werden, wenn (a) sechs Gebrochene Blocks aus jeweils $2\,Superzeilen \times 3\,Superspalten$ vorgebeben werden und ggf. im Nachgang eine ganze Zeile verschoben wird (Abb. 68). Dies setzt also zunächst Designvorgaben für einen einfaktoriellen Versuch mit sechs Wiederholungen voraus. Das Design wird im Nachgang durch den Nutzer in ein zweifaktorielles Design (Faktor Intensität mit zwei Stufen) mit sechs Wiederholungen überführt. Randomisationsplan und Zuordnung von Blockungen können also nicht über den normalen PIAF-Import werden, sondern müssen per Hand oder durch Eigenlösungen bewerkstelligt werden.

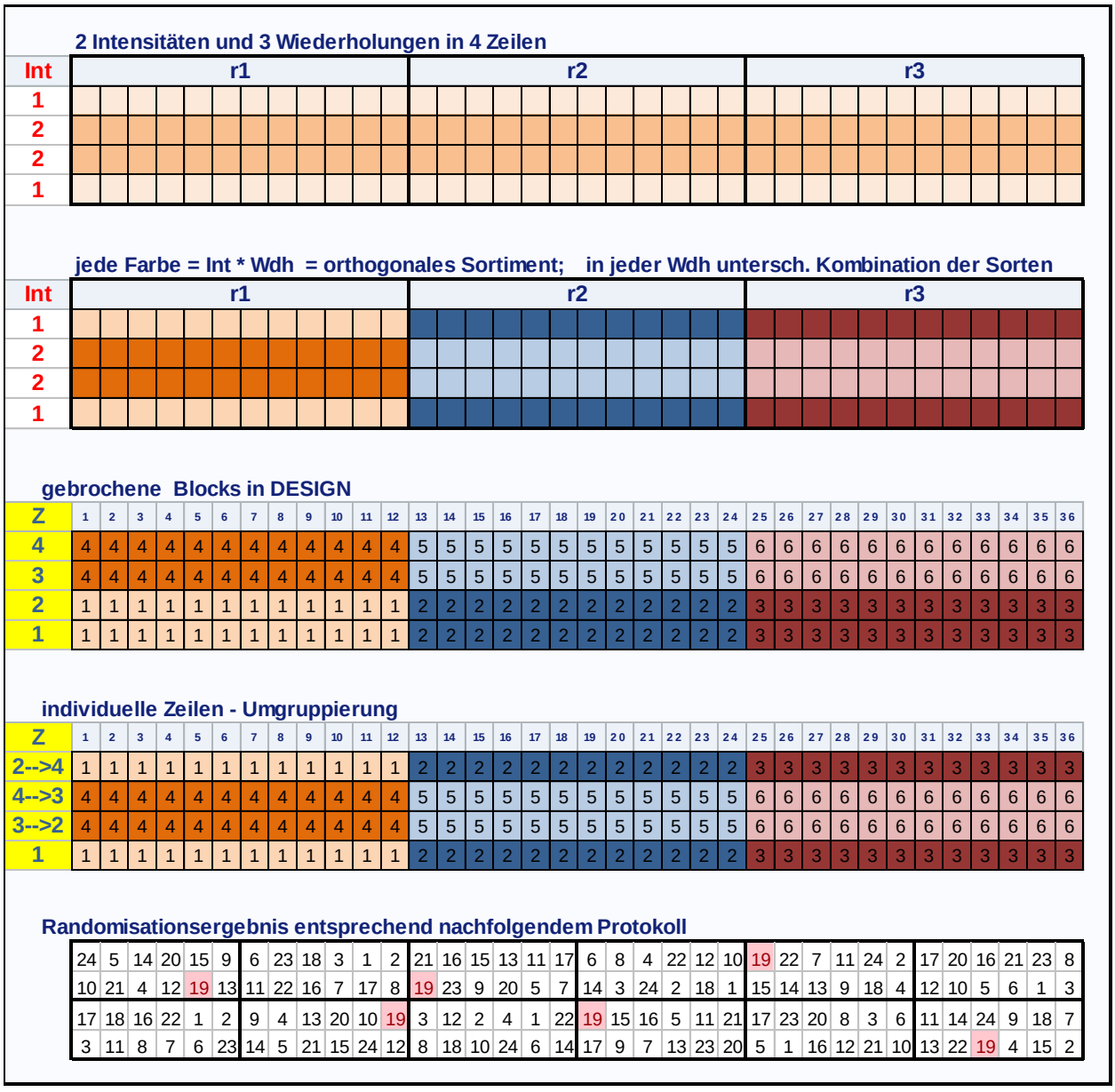


Abbildung 68: Zweifaktorielle Spaltanlage mit nebeneinander liegenden Wiederholungen und Zeilen-Umgruppierung

Tabelle 27: Designvorgaben für zweifaktorielle Spaltanlage mit nebeneinander liegenden Wiederholungen

Bereich	Bezeichnung	Vorgabe
Parametereingabe	Anzahl Prüfglieder	24
	Anzahl Zeilen	4
	Anzahl Spalten	36
	Zeilenbalancierung	ja
	Spaltenbalancierung	ja
	Anzahl Zeilengruppen	ohne / 4
	Anzahl Säulen	auto / 6
	Anzahl Superzeilen	auto / 2
	Anzahl Superspalten	3 / 3
	räumlicher Kovarianz-Parameter - zweidimensional	auto / 0
	Kovarianz-Parameter zwischen Zeilen-Nachbarn	auto / 0
	Koeffizient für Gewichtung der Nachbarschaften	0,01
Zusatzinformationen	Anzahl WDH je Prüfglied	6

In Situationen wie dieser, bei der es je Zeile mehr Parzellen als Prüfglieder gibt, werden die Kovarianzen für die räumliche Verteilung und für die Nachbarschaftsbalancierung in der ‚auto'-Funktion auf null gesetzt, da es in dieser Konstellation mitunter zu unerwünschten Randomisationsergebnissen kam.

7.6 PIAFStat-Verfahren ‚DESIGN A/B'

Dieses Vorhaben wird als Auftrag an die BioMath GmbH in 2018 abgeschlossen. Das Konzept ist durch die LFA erstellt und in einem Pflichtenheft für dieses Vorhaben abgelegt.

Das Nutzerhandbuch wird nach Abschluss aktualisiert.

7.7 Teilrandomisation

Dieses Vorhaben wird als Auftrag an die BioMath GmbH in 2018 abgeschlossen. Das Konzept ist durch die LFA erstellt und in einem Pflichtenheft für dieses Vorhaben abgelegt.

Das Nutzerhandbuch wird nach Abschluss aktualisiert.

7.7.1 Lücken wegen nicht verfügbarer Parzellenpositionen

7.7.2 Zwei außen platzierte Teilsortimente mit Rändern

7.7.3 Zwei unsystematisch platziert Teilsortimente mit Rändern

7.7.4 Mehr als zwei Teilsortimente

7.7.5 Erhöhte Wiederholungszahl für ein Standard-Prüfglied

Beispiel: $v = 9$, davon:

$v_1 = 8$ mit $r = 4$ für Rest-Prüfglieder

$v_S = 1$ mit $r = 8$ für Standard

PlanID	RunID	DesignID	trt	NND	NNLD	MST	MSTL	MAX_NO_ROWNEIGHBOR	no_of_reps
1	1	1	1	2,3	2,3	2,3	2,2	6	8
1	1	1	2	2,0	1,9	2,0	1,9	5	8
1	1	1	3	2,0	2,0	2,0	1,9	6	8
1	1	1	4	2,2	2,1	2,1	2,1	5	8
1	1	1	5	2,3	2,3	2,2	2,2	5	8

Plan 1 mit Prüfglied-Kürzel

Z	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6	Sp7	Sp8	Sp9	Sp10
4	2	5	4	1	4	3	1	3	2	5
3	3	1	2	5	3	5	4	2	4	1
2	5	4	3	4	2	1	5	1	3	2
1	1	3	1	2	5	2	3	4	5	4

Säulen im Plan

Z	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6	Sp7	Sp8	Sp9	Sp10
4	1	1	2	2	3	3	4	4	5	5
3	1	1	2	2	3	3	4	4	5	5
2	1	1	2	2	3	3	4	4	5	5
1	1	1	2	2	3	3	4	4	5	5

gebrochene Blocks im Plan

Z	Sp1	Sp2	Sp3	Sp4	Sp5	Sp6	Sp7	Sp8	Sp9	Sp10
4	3	3	3	3	3	4	4	4	4	4
3	3	3	3	3	3	4	4	4	4	4
2	1	1	1	1	1	2	2	2	2	2
1	1	1	1	1	1	2	2	2	2	2

Plan für $\boldsymbol{v, k, s = 5, 4, 10}$ *mit r=* $\boldsymbol{8, SG = 5;\ GB = 2 \times 2}$*; rechteckig*

Abbildung 69: Schritt 1: Randomisation eines überbesetzten Standard-Prüfgliedes (5)

Die Verteilung des Standards erscheint ggf. suboptimal gegenüber einer intuitiven händischen Verteilung. Sie lässt aber Platz für die geeignete Verteilung der anderen Prüfglieder um die 8 Standardparzellen und erfüllt aufgrund der Randomisation die Voraussetzungen für Wahrscheinlichkeitsrechnungen. Die Verteilung kann aber punktuell auch per Hand variiert werden. Diese Möglichkeit ist in Schritt 2 durch die blauen Pfeile angedeutet. Damit könnten die Standards von den Außenspalten ferngehalten werden, was den Vergleich aller Rest-Prüfglieder zum Standard verbessert (beidseitig Nachbarschaft und Vergleich zu Rest-Prüfgliedern).

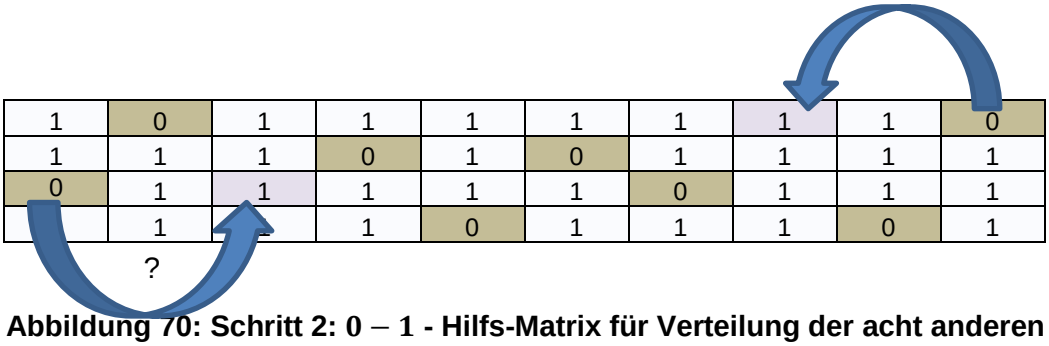


Abbildung 70: Schritt 2: $0-1$ - Hilfs-Matrix für Verteilung der acht anderen Prüfglieder

Schritt 3: Randomisation der v_1 um gegebene v_S

	COL1	COL2	COL3	COL4	COL5	COL6	COL7	COL8	COL9	COL10
ROW4	6	9	3	1	8	7	4	2	5	9
ROW3	4	2	7	9	5	9	8	6	1	3
ROW2	9	7	6	4	1	3	9	5	2	8
ROW1	5	3	2	8	9	4	7	1	9	6

Abbildung 71: Schritt 3: Randomisation der acht anderen Prüfglieder (1-8) um einen gesetzten überzähligen Standard (9)

8 Was tun bei ungewollter systematischer Spaltenzusammensetzung

Bei relativ kleinen Versuchen wie z.B. $v,k,s = 8{,}4{,}8$, $v,k,s = 12{,}4{,}12$ oder z.B. $v,k,s = 12{,}8{,}6$ ist die Spaltenbalance in der Version bis März 2018 auffällig. Und zwar in der Weise, dass Prüfgliedpaare überproportional oft in gleichen Spalten stehen. Die Freiheitsgrade reichen dann nicht für alle Ziele gleichzeitig. Konkret konkurriert Nachbarschaftsbalancierung bei kleinen Versuchen extrem mit einer ausgewogenen Spaltenzusammensetzung.

Abb. 72 zeigt ein extrem überzeichnetes Beispiel eines ‚nicht verbundenes' Designs.

	COL1	COL2	COL3	COL4	COL5	COL6	COL7	COL8	COL9	COL10	COL11	COL12
ROW1	3	1	9	7	4	5	2	12	6	11	10	8
ROW2	5	11	12	8	6	3	10	9	4	1	2	7
ROW3	2	6	7	12	1	10	5	8	11	4	3	9
ROW4	10	4	8	9	11	2	3	7	1	6	5	12

Abbildung 72: komplett nicht verbundenen Plan ohne Spaltenbalance

a) Das vorrangige Ziel der Spaltenbalancierung, dass kein Prüfglied mehrfach in einer Spalte steckt (wenn vermeidbar) wird immer unproblematisch erreicht.

b) Aber eigentlich würde man von einer guten Spaltenbalance erwarten, dass auch systematische Prüfgliedverpaarungen in Spalten minimiert werden - genau das ist in Abb. 72 aber ins Gegenteil verkehrt - 100%ige Systematik.

Offensichtlich erreicht das System die Minimierung wiederholter Zeilennachbarschaften nur genau über diese Achillesferse, wenn die Designvorgaben nicht passend sind.

Frau Günther: Verpaarungen in den Spalten können theoretisch überhaupt nur vermieden werden, wenn gilt: $v > 1 + (k-1) \times r;\ \ mit\ r = k \times s$. Also bei 4 Wiederholungen in 4 Zeilen ab 13 Prüfglieder. Und das ist erstens nur in der Spaltendimension gedacht und unterstellt zudem, dass keine weiteren Design-Vorgaben erfolgten (Latinisierung, NB, ED)

vorgeschlagene mögliche Reaktionen (einzeln oder kombiniert):

- Wenn Nachbarschaftsbalancierung keine Rolle spielt (z.B. Mais mit Mantelreihen oder Raps mit plot-in-plot - Parzellen), kann unter O4 in B42 die Kovarianz für Zeilennachbarn statt auf ‚auto' auf ‚0' gesetzt werden. Bei sehr kleinen Versuchen wie $v,k,s = 8{,}4{,}8$ kann es sinnvoll sein, zusätzlich (!) auch die räumliche Kovarianz unter B41 auf ‚0' zu setzen; keinesfalls aber umgekehrt nur die räumliche Kovarianz unter B41 auf ‚0' zu setzen.
- Für die Anzahl Runs sollte dann unter O74 eine Zahl ≥ 25 vorgegeben werden.
- Wenn eine automatische Auswahl des (vermeintlichen) Best-Runs nicht die eigenen Ansprüche voll befriedigt, können in O61 unter B6A mehr Pläne als gebraucht abgefordert werden. Eine individuelle Selektion unter den Outputs ist aber nur möglich, wenn unter B6B ‚1' = jeder Plan vollrandomisiert gewählt ist.
- Wenn die Nachbarschaftsbalancierung keine vordergründige Bedeutung hat, sollten bei anfälligen Konstellationen die Option in Richtung O41, nicht aber in Richtung O44 gewählt werden (Effizienzverbessert ab Version März 2018).
- Ab Version März 2018 werden Designs mit nicht verbundenen Spalten ausselektiert.

- Ab Version Sommer 2018 wird auch der Parameter für die Zeilen-Spalten-Balancierung in die Auswahl der Best-Runs einbezogen. Optionale Steuerungsmöglichkeiten werden unter O4 eingerichtet:

Diese Hinweise können auch bei Versuchen mit $Spaltenzahl > Prüfgliedzahl$ $(s > v)$ relevant sein, ggf. auch bei anbautechnischen Spaltanlagen - hierzu erfolgen weitere Untersuchungen zur Stabilisierung der künftigen Anwendungen.

9 Import der Design-Informationen in einen PIAF-Versuch

9.1 Lageplan

Bei aktivierter Option O62 ‚PIAF-Lageplan' können die n erzeugten Lagepläne in n dat-Dateien so abgespeichert werden, dass sie komfortabel in PIAF importiert werden können (s. 6.9.2). Abb. 73 zeigt eine dieser Schnittstellen-Dateien, die mit dem Editor geöffnet wurde.

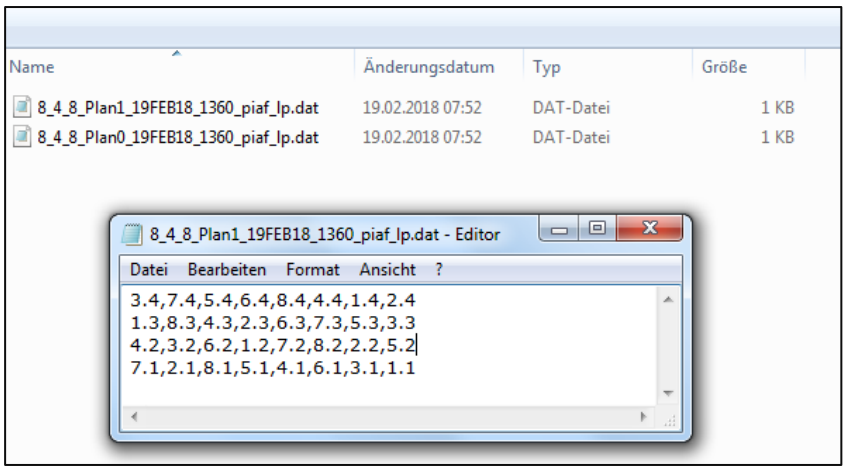


Abbildung 73: Beispiel einer gebildeten Schnittstellen-Datei des Planes Nr.1

Die Verwendung der Makrovariable „&PID" im Namen der Schnittstellen-Dateien (optionaler Block B62A, s. Abb.11) ermöglicht dem Anwender sehr leicht die Zuordnung der Schnittstellen-Datei zum gewünschten Plan (Plan*0*, Plan*1*,… , Plan*n*).

Der Import des Lageplanes in den geplanten und generierten PIAF-Versuch erfolgt in PIAF über Menüpunkt ‚Erfassung/Lageplan' (Abb. 74). Die Buttons ‚Im-/Export' sowie im dann geöffneten zweiten Fenster ‚ASCII-Datei öffnen' führen zum Aufruf der Schnittstellen-Datei. Zu finden sind die Schnittstellen-Dateien standardmäßig in dem in der PIAFStat-Grundeinstellung festgelegten RES-Verzeichnis (s. 6.9.2), wenn nicht vom Anwender zur systematischen Ablage der Ergebnisdateien andere Verzeichnisstrukturen geschaffen wurden.

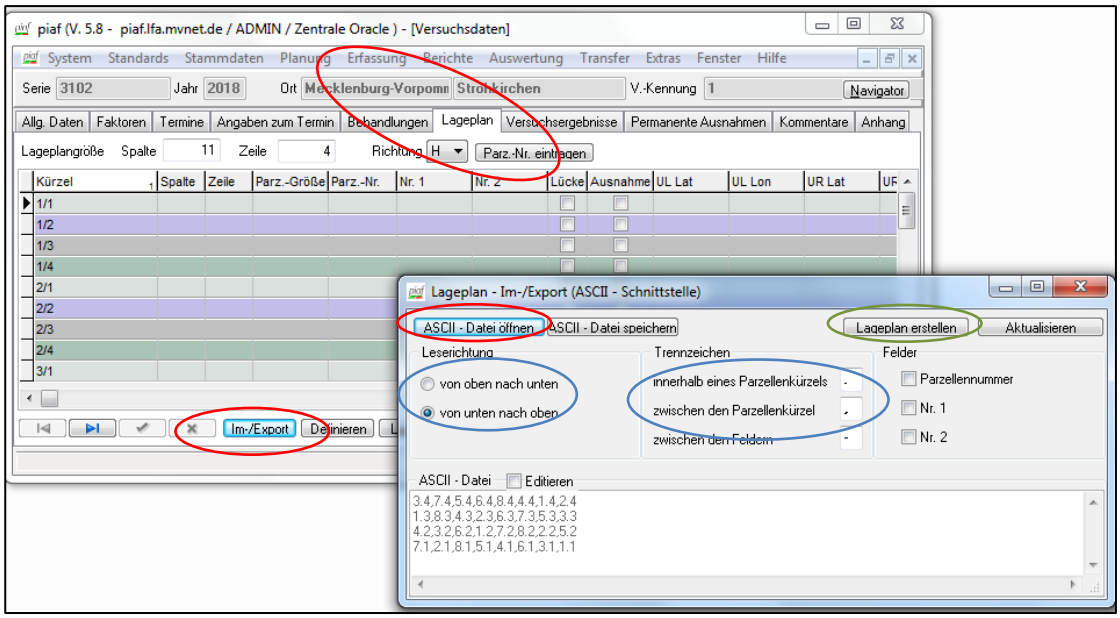


Abbildung 74: Lageplan-Import in PIAF

Wichtig sind die richtige Angabe der Leserichtung (von unten nach oben) sowie der Trenn–zeichen innerhalb eines Parzellenkürzels und zwischen den Parzellenkürzeln. Mit Klick auf den Button ‚Lageplan erstellen' erfolgt dann der Import. In der Ansicht des Lageplanes kann der Import kontrolliert werden. (Abb. 75)

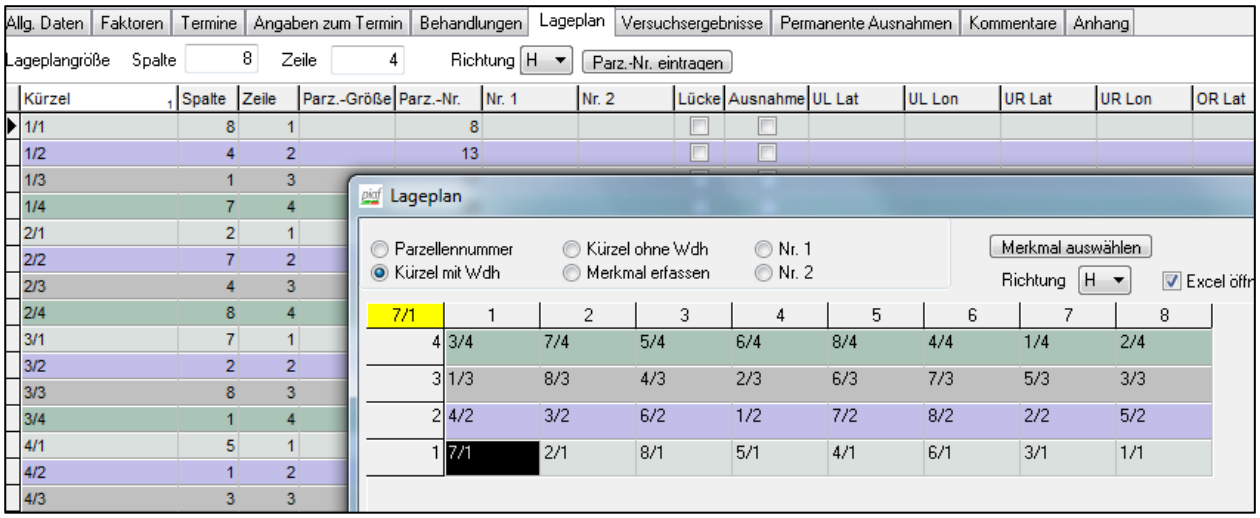


Abbildung 75: Ansicht des Lageplans in PIAF

9.2 Blockungsinformationen - Klass-Merkmale

Im Verfahren Design werden durch die Aktivierung der Option O63 ‚PIAF - Klass-Merkmale' Schnittstellen-Dateien für das Einlesen der Blockungsstrukturen in PIAF abgespeichert (s. 6.9.3). Im Dateinamen dieser Schnittstellen-Dateien wird vorgeschlagen, neben anderen Informationen die Bezeichnung moda und das Dateiattribut txt zu verwenden (optionaler Block B63A, s. Abb.12). Dadurch wird schon am Dateinamen schnell die Schnittstelle (Mobile Daten lesen) erkennbar. Ein weiterer wichtiger Teil des Dateinamens ist die Makrovariable „&PID", die die korrekte Zuordnung der Datei mit den Klass-Merkmalen zum entsprechenden Plan (Plan*0*, Plan*1*, Plan*n*) ermöglicht (s. 6.9.3).

Die moda-Schnittstellen-Dateien haben eine definierte Struktur mit einem Datenkopf, einem oder mehreren Datenblöcken und einer Datei-Ende-Marke (Abb. 76).

Im Datenkopf ist die PIAF-konforme Definition des Versuches festgelegt, in den Datenblöcken werden die KLASS-Merkmale Zeilengruppe, Säule und / oder gebrochener Block des entsprechenden Versuches aufgeführt. Nicht alle notwendigen Angaben für den Datenkopf sind im Verfahren Design bekannt. Sie werden in den Schnittstellen-Dateien mit Fragezeichen gefüllt und sind vor dem Import der Klass-Merkmale über den Editor zu aktualisieren (s. Abb. 75).

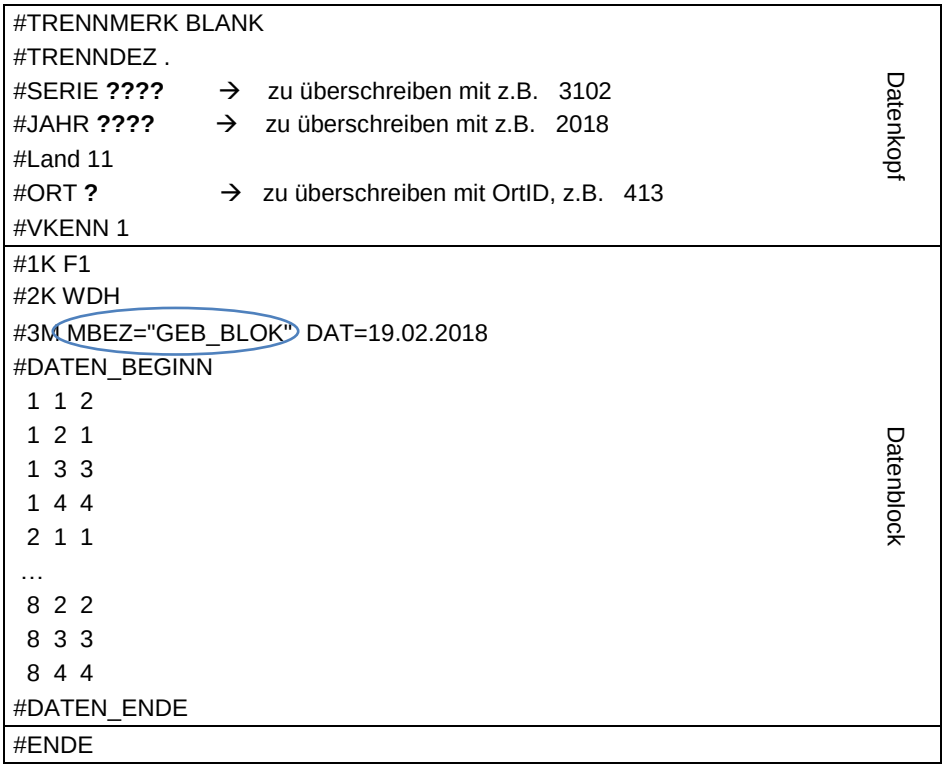


Abbildung 76: Struktur der Schnittstellen-Datei ‚PIAF – Klass-Merkmale', Beispiel: ‚gebrochener Block'

Der Import in PIAF erfolgt über Menüpunkt ‚Transfer/ Mobile Daten lesen'. Über ‚Dateiauswahl' können Pfad und Schnittstellen-Datei ausgewählt werden. Standardmäßig sind die Dateien in dem in der PIAFStat-Grundeinstellung festgelegten RES-Verzeichnis (s. 6.9.3), wenn nicht vom Anwender zur systematischen Ablage der Ergebnisdateien andere Verzeichnisstrukturen geschaffen wurden (Abb. 77).

Wichtig ist, dass bei Zuordnung der Merkmalsbezeichnung (MBEZ) Name1 gewählt wird (Abb. 77). Damit wird festgelegt, dass PIAF die Merkmale in der Schnittstellen-Datei anhand der Label-kurz-Bezeichnungen (= Name1) erkennt.

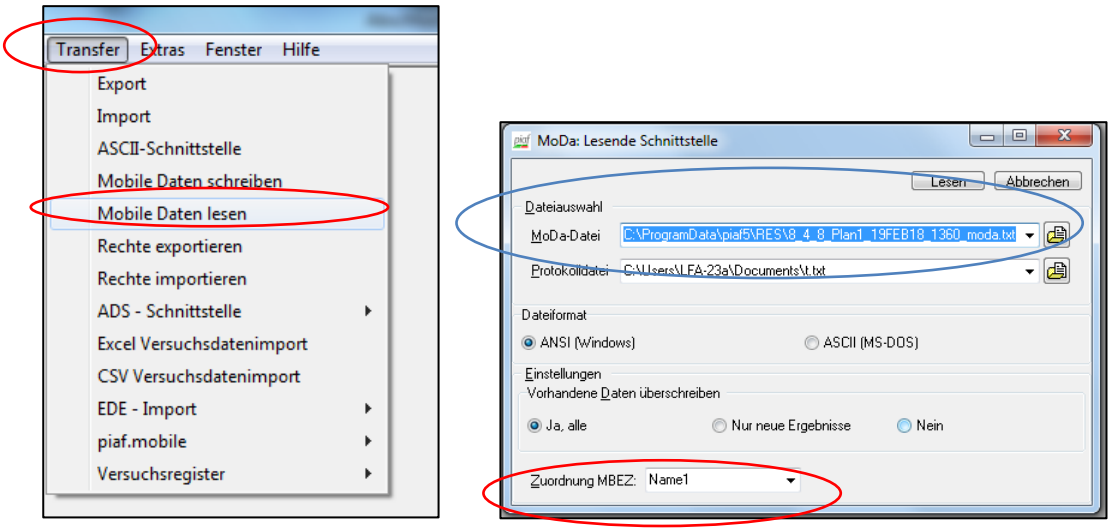


Abbildung 77: Import der Klass-Merkmale in PIAF

Ein kurzes Protokoll (Abb. 78) bestätigt ggf. das korrekte Einlesen der Daten.

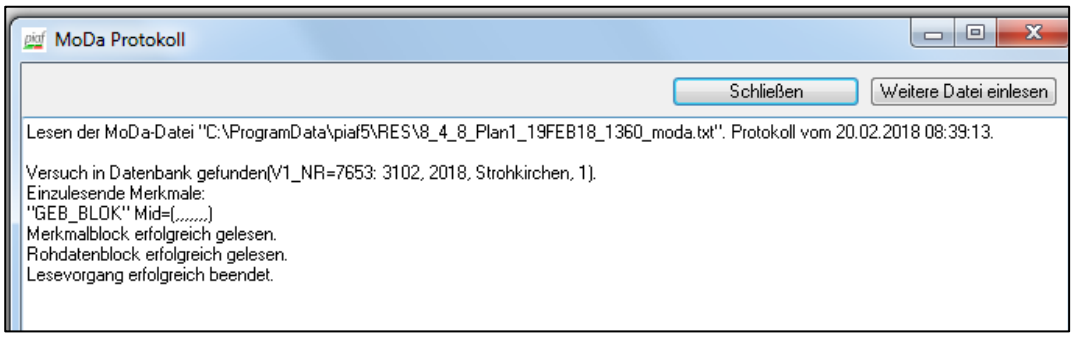


Abbildung 78: Protokoll-Ausgabe nach Import

Zur Überprüfung der korrekten Übernahme in PIAF bietet sich in der Lageplan-Ansicht von PIAF der Aufruf des entsprechenden Klass-Merkmals an, in diesem Fall ‚gebrochener Block' (Abb. 79).

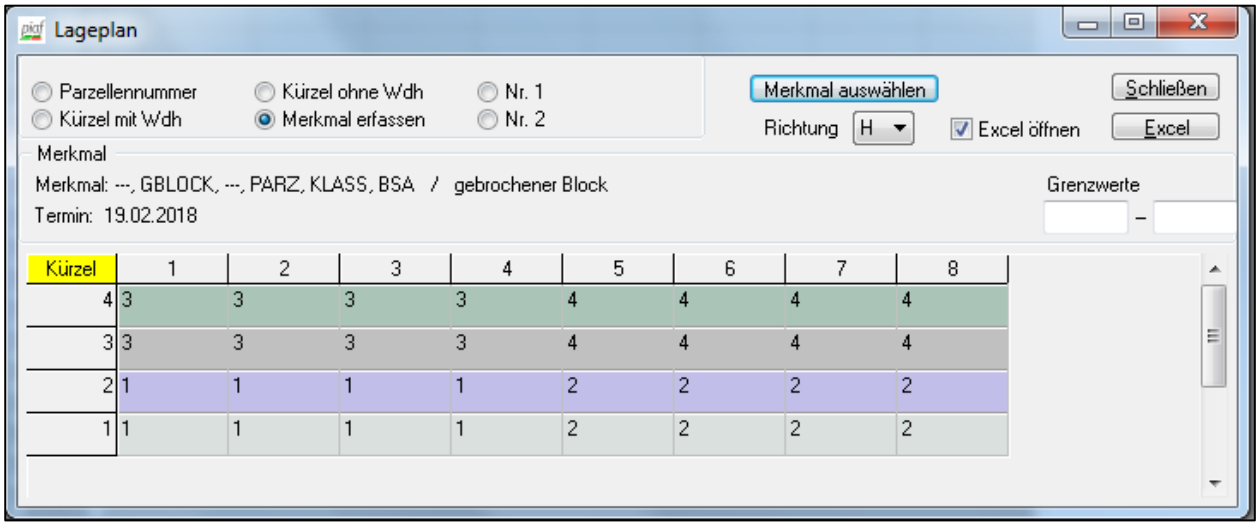


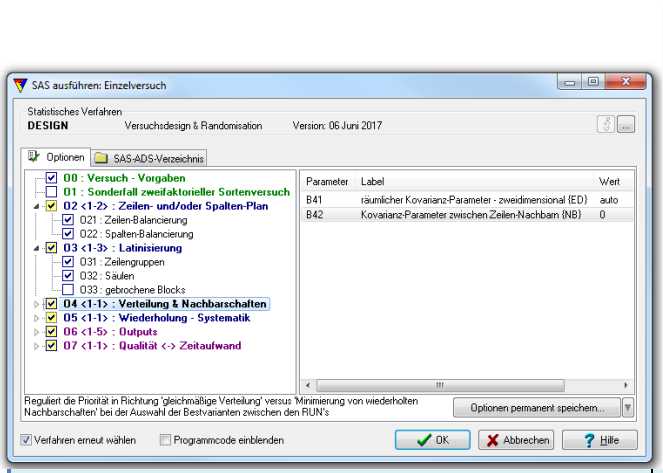
Abbildung 79: Klass-Merkmal ‚gebrochener Block' in der PIAF-Lageplan- Ansicht

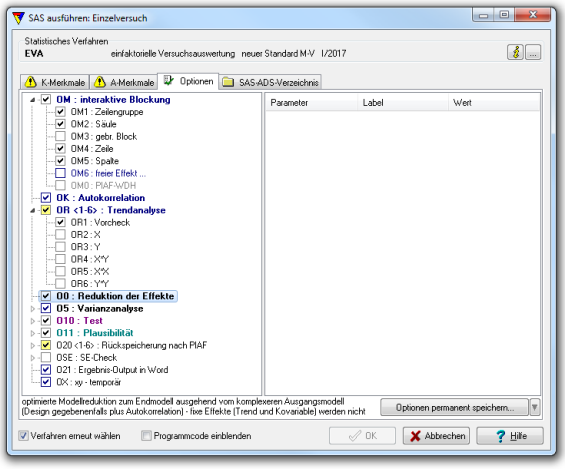
10 Einheit von Design und Auswertung

Ein Grundsatz der Planung uns Auswertung von Feldversuchen ist, dass die Designvorgaben das Auswertungsmodell bestimmen, Design und Auswertungsmodell sollen eine Einheit bilden. Ein Auswertungsansatz, welcher nicht das Design widerspiegelt, stellt kein der Realität angemessenes Modell dar. In diesem Grundsatz liegt keine Kompliziertheit für den Anwender, sondern er erhöht die Chancen, dass Störungen ggf. gut erkannt und adäquat berücksichtigt werden können (Adjustierung, means→ lsmeans, Bodenausgleich ...).

Die Palette der für PIAFStat entwickelten Verfahren greift diesen Ansatz auf. Die Abfolge der Versuchsbearbeitung in dem Verfahrens-Komplex DESIGN → PLAUSI → EVA/ZVA → HOHENHEIM-GÜLZOWER-SERIENAUSWERTUNG baut logisch aufeinander auf und stellt einen in Nutzerführung und Funktionalität abgestimmten Gesamtkomplex dar. Dies gewährleistet, dass von Versuchsplanung über Plausibilitätsprüfung, Einzelversuchsauswertung bis hin zu hochkomplexen Serienauswertungen ein stimmiger Prozess stattfindet, welcher die Chancen auf Fehlerminimierung und Verbesserung von Wiederholbarkeit und Reproduzierbarkeit zusammenfassend verdichteter Ergebnisse für die Praxis erhöht.

Nachfolgend wird beispielhaft die abgestimmte Nutzeroberfläche von einfaktorieller Versuchsplanung (DESIGN) und Einzelversuchsauswertung (EVA) gezeigt (Abb. 80).





Vorgabe im Verfahren DESIGN	adäquate Vorgabe im Verfahren EVA
O21: Zeilen-Balancierung ON	OM4: Zeile ON
O22: Spalten-Balancierung ON	OM5: Spalte ON
O31: Zeilengruppen ON	OM1: Zeilengruppe ON
O32: Säulen ON	OM2: Säule ON
O33: gebrochene Blocks OFF	OM3: gebr. Block OFF
O4 / B41: räumliche ... ON	OK: Autokorrelation ON OR1: Vorcheck ON
→komplexes Design mit vielen Vorgaben	O0: Reduktion der Effekte ON

Abbildung 80: Abgestimmte Nutzeroberflächen für Versuchsplanung (DESIGN) und Versuchsauswertung (EVA)

Für diese Versuchsanlage wurden viele Designvorgaben gestellt - entsprechend komplex ist auch das Ausgangsmodell in der Auswertung. Auch wenn dadurch wahrscheinlich die Zielkriterien simultan sehr gut erfüllt werden und der Versuch somit gegenüber verschiedensten Fehlerarten prophylaktisch gut ‚vorbereitet' ist, wäre zu ‚befürchten', dass das Auswertungsmodell zunächst ‚überfrachtet' ist. Ggf. treten bei der Auswertung mit sehr komplexen Modellen Konvergenz-Probleme auf. Andererseits ist auch kaum zu erwarten, dass alle prophylaktisch

berücksichtigten Fehlerarten tatsächlich zugleich eintreten. Erfahrungsgemäß treten im Einzelfall äußerst selten mehr als ein bis zwei relevante Störgrößen gleichzeitig in relevantem Ausmaß auf.

Insofern bietet es sich an, das Modell im Zuge der Auswertung zu reduzieren. Dies kann in EVA durch die Option O0 automatisiert werden (Tab. 61). Traditionell war eine derartige Modellreduktion im Hinblick auf theoretische Forderungen (α-Einhaltung, Freiheitsgrade-Approximation...) nicht gewollt oder üblich - allerdings waren die Designvorgaben auch relativ überschaubar. Im Zuge der Entwicklung komplexerer Designvorgaben in gemischten Modellen, der Einbeziehung von räumlicher Kovarianz, von Kovariablen oder Post-Blocks u.v.m. wurden zunehmend objektivierte Verfahren der Modellreduktion entwickelt und akzeptiert. Entscheidend ist dabei zum einen, dass die Reduktion nicht subjektiv und nicht z.B. nach Variationskoeffizient oder Grenzdifferenz erfolgt, sondern nach einem objektivierten Minimierungsparameter (AICC). Und zum anderen soll das Ausgangsmodell (vor der Reduktion) das Design völlig widerspiegeln. Erst im Zuge der Reduktion entsteht dann aus dem komplexeren Ausgangsmodell ein simpleres pragmatisches Endmodell, welches nur noch im konkreten Versuch relevante Designvorgaben modelliert.

Die DESIGN-Vorgabe einer räumlichen Varianz in der Fläche (B41) stellt eine zulässige und aussichtsreiche Voraussetzung dar, Heterogenität in der Fläche auch in der Einzelversuchsauswertung zu berücksichtigen. Es ist vorab nicht absehbar, ob diese Berücksichtigung ggf. eher durch ein Autokorrelationsmodell (zufällig, amorph → Schrumpfung) oder durch eine polynome zweidimensionale Regression (fix, polynomisch → ohne Schrumpfung) erfolgen sollte bzw. ob sie ganz verworfen wird. Die in EVA aktivierte Option ‚OK' verbindet Blockung und Autokorrelation und findet automatisch das geeignetste Modell (Zielkriterium: Minimierung AICC (REML); „so viel wie nötig und so wenig wie möglich"). Dieser Automatismus ist in EVA für den Regressionsansatz nicht implementiert. Option ‚OR1' liefert aber einen Vorcheck, ob einzelne Regressionsparameter signifikant ansprechen. Ist dies der Fall, kann man die Auswertung in EVA mit Zuschaltung dieser Regressionsparameter als neues Szenarium ablaufen lassen. Die Auswertungen mit vs. ohne Regression können verglichen werden, wobei i.d.R. das Szenarium, welches den kleineren AICC (ML) liefert, zu bevorzugen ist.

Im Protokoll der Output-Excel-Datei sind in der Rubrik ‚Zusatzinformationen' in der Zeile ‚zufällige Blockungsfaktoren' alle Blockungen aufgeführt, die bei der Einzelversuchsauswertung in PIAFStat (EVA, ZVA) angehakt werden sollten, damit sich Design und Auswertungsmodell decken (Tab. 28).

Tabelle 28: Vorgaben im Verfahren DESIGn entsprechend Abb. 80

Bereich	Bezeichnung	Vorgabe
Parametereingabe	Anzahl Prüfglieder	12
	Anzahl Zeilen	8
	Anzahl Spalten	6
	Zeilenbalancierung	ja
	Spaltenbalancierung	ja
	Anzahl Zeilengruppen	auto / 4
	Anzahl Säulen	auto / 4
	Anzahl Superzeilen	ohne / 8
	Anzahl Superspalten	ohne / 6
	räumlicher Kovarianz-Parameter - zweidimensional	auto / 0.0319
WDH - Systematik	Sortierung nach Variante	Zeile bzw. Zeilengruppe / Zeile bzw. Zeilengruppe
	zufällige Blockungsfaktoren	**Z Sp ZG S**

11 Überleitung

(Stand Mai 2017)

Wissenschaftliche internationale Veröffentlichungen

- Piepho, H.P., Williams, E. R., Michel, V. (2015): Beyond Latin Squares: A Brief Tour of Row-Column-Design. Agronomy Journal 107, 2263-2270.
- Piepho, H.P., Michel, V., Williams, E.R. (2016): Nonresolvable Row-Column Designs with an Even Distribution of Treatment Replications. Journal of agricultural, Biological and Environmental Statistics 21, 227-242.
- Piepho, H.P., Michel, V., Williams, E.R. (2018): Neighbour balance and eveness of distribution of treatment replications in row-column designs. Biometrical Journal. (in preparation)

Vorträge

- Michel, V. (11.10.2016): Design und Randomisation - SAS-QC-Modul im PIAF-Konstrukt. Sitzung der AG PIAF-Allgemein, Würzburg.
- Michel, V. (10.11.2016): Design und Randomisation - Konzepte. Sitzung der PIAF-KG, Berlin.
- Michel, V., Zenk, A. (20.2.2017): Versuchsdesign & Randomisation. LFA-Kolloquium, Gülzow.
- Michel, V., Piepho, H.P. (29.6.2017): Erweiterte Zeilen-Spalten-Pläne - methodische Entwicklungen und praktische Umsetzung in PIAF. Tagung Biometrische Gesellschaft, Neustadt (W).
- Piepho, H.P., Williams, E.R., Michel, V. (1.9.2017): Row-column designs for field trials with an even distribution of treatment replications. CEN-ISBS Conference on Biometrics, Wien
- Michel, V., Zenk, A. (9.4.2018): PIAFStat.Verfahren ‚DESIGN' - optimierte Anlagen bei Prüfglied-Gruppierung, Füllparzellen, Ummantelung einzelner Prüfglieder LFA-Kolloquium, Gülzow. (in Vorbereitung)

Veranstaltungen, von Themenbearbeitern organisiert und geleitet

- Michel, V., Zenk, A. (10.10.2016): Sitzung der AG PIAF-Auswertung - Schwerpunkt Versuchsdesign, Würzburg.
- Michel, V., Zenk, A. (17.05.2017): Workshop 'Versuchsdesign und Randomisation', (Bund, Länder, assoziierte private PIAF-Nutzer), Kassel.
- Michel, V., Zenk, A. (15./16.05.2018): Workshop ‚PIAFStat-Verfahren ‚DESIGN' - optimierte Anlagen bei Prüfglied-Gruppierung, Füllparzellen und bei Spaltanlagen', (Bund, Länder, assoziierte private PIAF-Nutzer), Kassel. (in Vorbereitung)

12 Berücksichtigte Literatur

Azais, J.M., Bailey, R.A., Monod, H. (1993). A catalogue of efficient neighbour-designs with border plots. Biometrics 49, 1252-1261.

Bailey, R.A., Kunert, J., and Martin, R.J. (1990). Some comments on gerechte designs. I. Analysis for uncorrelated errors. *Journal of Agronomy and Crop Science* 165, 121–130.

Bailey, R.A., Kunert, J., and Martin, R.J. (1991). Some comments on gerechte designs. II. Randomization analysis, and other methods that allow for inter-plot dependence. *Journal of Agronomy and Crop Science* 166, 101–111.

Bailey, R.A., Druilhet, P. (2004). Optimality of neighbor-balanced designs for total effects. The Annals of Statistics 32, 1650-1661.

Behrens, W.U. (1956a). Feldversuchsanordnungen mit verbessertem Ausgleich der Bodenunterschiede. Zeitschrift für Landwirtschaftliches Versuchs- und Untersuchungswesen 2, 176-173.

Behrens, W.U. (1956b). Die Eignung verschiedener Feldversuchs-Anordnungen zum Ausgleich der Bodenunterschiede. *Zeitschrift für Acker- und Pflanzenbau* 101, 243–278.

Chan, B.S.P., and Eccleston, J.A. (1998). On the construction of complete and partial nearest neighbour balanced designs. Australasian Journal of Combinatorics 18, 39-50.

Chan, B.S.P., and Eccleston, J.A. (2003). On the construction of nearest neighbour balanced row-column designs. Australian and New Zealand Journal of Statistics 45, 97-106.

Cheng, C.S. (1983). Construction of optimal balanced incomplete block designs for correlated observations. The Annals of Statistics 11, 240-246.

Eccleston, J.A., Chan; B.S.P. (1998). Design algorithms for correlated data. In COMPSTAT 1998: Proceedings in Computational Statistics, eds. R. Payne and P. Green, pp. 41–52. Heidelberg, New York: Physica-Verlag.

Federer, W. T., Nair, R. C. and Raghavarao, D. (1975). Some augmented row-column designs. Biometrics 31,361–375.

Federer, W.T., Basford, K. (1991). Competing effects designs and models for two-dimensional arrangements. Biometrics 47, 1461-1472.

Gilmour, A. R., Cullis, B. R., and Verbyla, A. P. (1997), “Accounting for natural and extraneous variation in the analysis of field experiments”, *Journal of Agricultural Biological and Environmental Statistics*, 2, 269–293.

Jaggi, S., Varghese, C., Varghese, E., Sharma, A. (2015). Web generation of experimental designs balanced for indirect effects of treatments (WEB-DBIE). Computers and Electronics in Agriculture 111, 52-68.

John, J. A., and Williams, E. R. (1995), “*Cyclic and computer generated designs*, *Second edition*”, London: Chapman and Hall.

John, J. A., and Williams, E. R. (2002), “t-latinized designs”, *Australian and New Zealand Journal of Statistics*, 40, 111–118.

Kiefer, J., Wynn, H.P. (1981). Optimum balanced block and Latin square designs for correlated observations. The Annals of Statistics 9, 737-757.

Lawless, J.F. (1971). A note on certain types of BIBD's balanced for residual effects. The Annals of Mathematical Statistics 42, 1439-1441.

Moser, E.B. (1992). Ordination and Display of Multivariate Data. pp1345-1355. In: SAS

Piepho, H.P. (2015): Generating efficient designs for comparative experiments using the SAS procedure OPTEX. *Communications in Biometry and Crop Science* 10, 96-114.

Piepho, H.P., Michel, V., Williams, E.R. (2015): Beyond Latin squares: A brief tour of row-column designs. *Agronomy Journal* 107, 2263-2270.

Piepho, H.P., Michel, V., Williams, E.R. (2016): Nonresolvable row-column designs with an even distribution of treatment replications. *Journal of Agricultural, Environmental and Biological Statistics* 21, 227-242.

Rees, D.H. (1967). Some designs of use in serology. Biometrics 23, 779-791

Schmidtke, J., and Zink, G. (2007). PIAF /PIAFStat (Planning-, information- and evaluation-system for field trials). pp. 203□208. In: Piepho, H.P., Bleiholder, H. (eds.): Agricultural field trials – today and tomorrow. Proceedings of the International Symposium 08–10 October 2007, Stuttgart-Hohenheim, Germany. Beuren: Verlag Grauer.

Smith, A.B., Lim, P. and Cullis, B.R. (2006). The design and analysis of multi-phase plant breeding experiments.Journal of Agricultural Science 144, 393–409.

Schuster, W., von Lochow, J. (1992). Anlage und Auswertung von Feldversuchen. 3. Auflage. Verlag Alfred Strothe, Frankfurt.

Tedin, O. (1931). The influence of systematical plot arrangement upon the estimate of error in field experiments. Journal of Agricultural Science 21, 191-208.

van Es, H.M., Gomes, C.P., Sellmann, M., van Es C.L. (2007). Spatially-balanced complete block designs for field experiments. Geoderma 140: 346-352.

Varghese, E., Jaggi, S., Varghese, C. (2014): Neighbor-balanced row-column designs. Communications in Statistics - Theory and Methods 43, 1261-1276.

Wilkinson, G. N., Eckert, S. R., Hancock, T. W., and Mayo, O. (1983). Nearest neighbour (NN) analysis of field experiments. Journal of the Royal Statistical Society B 45, 151–211.

Williams, E.R., Piepho, H.P., and Whitaker, D. (2011), Augmented p-rep designs. Biometrical Journal 53, 19□27.

Williams, E.R., and Piepho, H.P. (2017). An evaluation of Fisher bias in spatial designs. Journal of Agricultural, Biological and Environmental Statistics (submitted)

Williams, R.M. (1952). Experimental designs for serially correlated errors. Biometrika 39, 151-159.

Wolfinger, R. (1993). Covariance structure selection in general mixed models. Communications in Statistics 22, 1079-1106.

13 Abkürzungsverzeichnis

ADS **A**nforderungs - **D**aten - **S**chnittstelle von PIAF nach PIAFStat

RES **R**ücklieferungs - **E**rgebnis - **S**chnittstelle von PIAFStat nach PIAF

EVA PIAFStat - Verfahren ‚**E**infaktorielle **V**ersuchs-**A**uswertung'

ZVA PIAFStat - Verfahren ‚**Z**weifaktorielle **V**ersuchs-**A**uswertung'

B optionaler **B**lock in der Nutzeroberfläche

O **O**ption (oder Unteroption) in der Nutzeroberfläche

ED **E**ven **D**istribution - hier für gleichmäßige Verteilung der Wiederholungen jeweils von Prüfgliedern in der Versuchsfläche; als allgemeine Eigenschaft, aber auch konkret als Maßzahl für eine gleichmäßige Verteilung: ED_Design (in älteren Versionen ED_weighted bezeichnet) bzw. ED_Pruefglied (in älteren Versionen ED bezeichnet)

NB Nearst **N**eighbor **B**alance - hier für Nachbarschaftsbalancierung nur für direkte Zeilennachbarn; als allgemeine Eigenschaft, aber auch konkret als Maßzahl

PIAF **P**lanungs-, **I**nformations- und **A**uswertungssystem für das **F**eldversuchswesen

PIAFStat eigenständiges Auswertungsmodul im PIAF-Softwaresystem

Wdh. Wiederholung - Versuchselement, wiederholte Parzellen eines Prüfgliedes innerhalb eines Versuches

s siehe ... (Verweis)

s.a. siehe auch ... (Verweis)

<u>primäre Versuchs-Vorgaben im PIAFStat-Verfahren DESIGN</u>

k Anzahl Zeilen - Eingabevariable

r Anzahl Wiederholungen je Prüfglied - ergibt sich aus $\{v, k, s\}$

s Anzahl Spalten - Eingabevariable

v Anzahl Prüfglieder - Eingabevariable