Construction, Application and Extension of Resolvable Balanced Incomplete Block Designs in the Design of Experiments

Bachelor Thesis

by

Steffen Fissler

at the

Universität Konstanz

Department of Computer and Information Science

Algorithmics

1. Supervisor: Prof. Dr. Ulrik Brandes
2. Supervisor: Apl. Prof. Dr. Sven Kosub

Konstanz, 2017
Contents

1 Introduction 2

2 Related Work 4

3 Preliminaries 5
   3.1 Balanced Incomplete Block Designs ............................. 5  
   3.2 Kirkman Packing Designs ........................................... 7

4 Existence of Designs 14
   4.1 Conditions .............................................................. 14  
   4.2 Discovered Families for RBIBD ................................. 16  
      4.2.1 Kirkman Triple Systems ................................. 17  
      4.2.2 (v,4,1) Designs ............................................. 17  
      4.2.3 (2n,2,1) Designs ............................................. 17  
   4.3 Discovered Families for KP ................................. 18

5 Construction Methods 19
   5.1 General Methods - Difference Methods ........................ 19  
      5.1.1 Difference Sets ............................................. 19  
      5.1.2 Difference Systems ........................................ 21  
   5.2 Methods for k=3 .................................................... 22  
      5.2.1 Pure and Mixed Differences ............................... 22  
      5.2.2 PBD Approach ................................................. 24  
      5.2.3 Early Methods ................................................. 25  
      5.2.4 Walecki’s Method ............................................ 26  
      5.2.5 Pure and Mixed Method for Resolvable Designs ........... 26  
      5.2.6 Doubling Method and its Extension ...................... 29  
      5.2.7 Construction of specific KTS and KP .................... 31  
      5.2.8 Overview ...................................................... 33  
   5.3 Method for k=2 ...................................................... 34

6 Discussion 37
Abstract

This overview provides the foundation to explore the practicality of RBIBD and optimal KP in the design of experiments in order to create designs with fixed block size $k$ and $v$ points, $k|v$. We clarify the mathematical restrictions and possibilities of those designs. For this, we collect methods to construct RBIBD and KP. We see that for $k = 2$ all RBIBD can be constructed with ease and for $k = 3$ quite some methods do construct many but not all desired. An overview shows the results of this thesis regarding $k = 3$. We transform some of the methods to algorithms and we clarify the upper bound for arbitrary $k$, giving proof and present the results for various $k$. Further, we transform the theorem of doubling construction and introduce a method showing the difficulties and possibilities for improvement in this method. We introduce an extension to this method to construct a family of non-optimal Kirkman packing designs with $r = M(v) - 1$. 
Chapter 1

Introduction

In experiments you might want to conduct several experiments or treatments with the same group of persons. To do this, you arrange them in subgroups which we call blocks and conduct the experiment with them. To get a better measurement you want to remix the groups and do some more experiments, while any two persons should be together in a block exactly or at most $\lambda$ times. In this report we focus on $\lambda = 1$ and $\lambda \leq 1$, so each participant should meet every other participant once or at most once in a block. To meet other participants, a person needs to be in blocks several times. A collection of blocks with having each participant in exactly one of these blocks is called a round or parallel class. Most interesting is the number of rounds we can construct. The persons and the blocks together form balanced incomplete block designs or BIBD if $\lambda \in \mathbb{N}$. To allow rounds or parallel classes, the fixed size of the subgroups (blocks) $k$ must be a divisor of the amount of persons $v$. If the blocks can be partitioned such that every person is in every round, but meets everybody else $\lambda$ times, we call it resolvable balanced incomplete block design, resolvable BIBD or RBIBD.

These designs are interesting in the design of experiments but still do not have a lot of direct methods of construction. We give an overview of existing designs and an introduction to the most common methods of direct construction as well as look at some particular methods for $k = 3$: Walecki’s method, resolvable pure and mixed method and Doubling method among others. This helps to get a good understanding of the state of the art, the current possibilities and the limitations of existing methods.

Recently, Both et al. [10] wrote about algorithms for the design of experiments. However, there are uncertainties how many rounds can be produced for specific parameters. He uses randomized algorithms to find as many rounds as possible with depth-search within a certain time and does not take into account that there exist many methods to produce direct results.

We want to address the issue of the maximum number of rounds, collect methods to construct those designs and clarify the possibilities and constraints of the designs. We will define and explain relevant basics in chapter 3. Then we look for parameters
and constraints for which our desired designs exist in chapter 4 with a closer view at designs with block size \( k = 3 \) (Kirkman Triple Systems and Kirkman packing designs). Chapter 5 gives information about some of the most used general construction methods to obtain the blocks of BIBD. Then we get to specific construction methods to create the Kirkman Triple Systems, Kirkman packing designs and \((2n,2,1)\)-BIBD). Chapter 6 discusses the results and chapter 7 holds the conclusion of this thesis.
In this part we give a summary of related topics which are interesting for further research and comparison.

There are practical approaches like the software z-Tree which is used for economic experiments. It uses backtracking and exhaustive search instead of direct construction methods. Both et al. [10] improve this approach. We will discuss improvements later. Software approaches like z-Tree [13] use backtracking and exhaustive search instead of using direct construction methods. Another software tool, Sage [36], uses a PBD approach to obtain Kirkman Triple Systems directly. Approaches different from direct methods of construction are linear integer programming [25, 26, 27, 35], or heuristic models [30]. A German introduction for BIBD is given by Beutelspacher [8]. The Oberwolfach-Problem [1, 20] is similar to Kirkman packing designs, but not covered in this thesis. The social golfer problem [37] is about blocks of size 4, yet there are few solutions in this source. Other related fields are resolvable group-divisible designs (resolvable GDD or RGDD) and Kirkman Frames [31, 32], which could help finding good solutions. Wallis [32] also describes Kirkman covering designs. Whereas our described Kirkman packing designs have every point meeting every other at most once, Kirkman covering designs have any two points meeting at least once.
Chapter 3

Preliminaries

This chapter introduces relevant definitions and notations which are used throughout this work.

3.1 Balanced Incomplete Block Designs

3.1 Definition Let \( v \) and \( \lambda \) be positive integers and \( K \) a set (finite or infinite) of positive integers. A \((v, K, \lambda)\)-pairwise balanced design (PBD) is a pair \((X, B)\) where \( X \) is a set of points (or treatments), \( B = (B_i : i \in I) \) is a family of subsets of \( X \), called blocks, such that

(i) \( |X| = v \),
(ii) \( 2 \leq |B_i| \in K, \forall i \in I \),
(iii) the number of indices \( i \in I \) for which \( \{x, x'\} \subset B_i \) is exactly \( \lambda \), for each pair \( x, x' \in X, x \neq x' \).

3.2 Definition If the blocks of a pairwise balanced design have the same size (i.e. \( K = \{k\} \)), we call it a \((v, k, \lambda)\)-balanced incomplete block design (BIBD).

Note: The design is balanced because every pair of points occurs exactly \( \lambda \) times and it is incomplete because no block contains all points.

3.3 Definition If the blocks can be arranged in groups so that the \( \frac{v}{k} \) groups are disjoint and contain in their union each point exactly once, we call it a resolvable BIBD or RBIBD. The groups are called parallel classes or rounds.
3.4 Definition The number of blocks is \( b = |\mathcal{B}| \) and \( r \) is the number of repetitions for each element over all blocks (so each element occurs \( r \) times).

In a resolvable BIBD, \( r \) is also the amount of classes.

3.5 Proposition Given a \((v,k,\lambda)\)-BIBD. It holds that

\[
\begin{align*}
r &= \frac{\lambda(v-1)}{k-1} \\
 b &= \frac{\lambda v(v-1)}{k(k-1)}
\end{align*}
\]

We will calculate \( r \) and \( b \) and examine how often an element occurs in all blocks \((r)\). If we take an element, it needs to be \( \lambda \) times with every other element \((v-1\) other elements) and it can be with \((k-1)\) elements in a block. This leads to \( r = \frac{\lambda (v-1)}{k-1} \).

The number of blocks, \( b \), can be expressed in a resolvable BIBD as groups per class \((\frac{v}{k})\) multiplied by the amount of classes \((r)\): \( b = \frac{v}{k} \cdot r \), or \( b = \frac{\lambda v(v-1)}{k(k-1)} \). In a non-resolvable design we take the repetitions \( r \), which we need for every point of \( v \) and divide this by \( k \), because in every block we cover \( k \) repetitions of points.

![Figure 3.1: Simplest, non-trivial BIBD: (7,3,1)-BIBD](image)

\[
\begin{align*}
B_1 &= \{1, 2, 4\} & B_2 &= \{2, 3, 5\} & B_3 &= \{3, 4, 6\} & B_4 &= \{4, 5, 7\} \\
B_5 &= \{5, 6, 1\} & B_6 &= \{6, 7, 2\} & B_7 &= \{7, 1, 3\}
\end{align*}
\]

3.6 Examples

Let us exemplify the definitions of the previous section by looking at figure 3.1: The \((7,3,1)\)-BIBD.

\( v = 7 \) is the number of points,
First round:
\{1, 2, 0\} \{1, 2, 0\} \{1, 2, 0\}

\(k = 3\) the size of a block,
\(b = 7\) the number of blocks and
\(r = 3\) the number of repetitions for any element. If we have a look at the resolvable
\((9,3,1)\)-BIBD (figure 3.2), we get \(v = 9\) and \(k = 3 \Rightarrow r = \frac{1}{2} = 4\), so \(b = 12\).

**Definition/Notation in other literature BIBD**
In other papers and books, a \((v,k,\lambda)\)-BIBD is also described as a \((v,b,r,k,\lambda)\)-BIBD,
\(2 - (v,k,\lambda)\) Steiner design or Steiner System \(S_\lambda(2,k;v)\).

### 3.2 Kirkman Packing Designs

The now defined resolvable BIBD cover the \(v\) with \((k|v)\) and \(v \equiv k \mod (k-1)\),
as we will see in 4.1. For \(k = 3, \lambda = 1\), those are the \(v\) with \(v \equiv 3 \mod 6\) and
we call the designs Kirkman Triple Systems. However, the \(v\) with \(v \equiv 0 \mod 6\)
are missing to cover all \(v\) with \((k|v)\). To cover them, we will introduce optimal
Kirkman packing designs which take all \(v\) with \(v \equiv 0 \mod 6\), but has \(\lambda \leq 1\). Their
generalization, Kirkman packing designs cover all \(v \in \mathbb{N}\) for \(k = 3\). Note that for
optimal Kirkman packing designs the block size is always \(k = 3\).

The optimal Kirkman packing designs have the maximum number of possible
rounds and are thus complementing with the Kirkman Triple Systems the \(v\) with
3|\(v\). Informally, in terms of resolvable BIBD, we look for designs with \(k = 3\) and
\(\lambda \leq 1\).

The remaining topics in this section are the maximum number of possible rounds,
the existence of Kirkman packing designs and the construction of them. To approach
these topics we will define the number of rounds $r$ again in this section, giving a precise upper bound of rounds for designs which are no BIBD. Then we introduce Nearly Kirkman Triple Systems and how they can be changed to be an optimal Kirkman packing design. Beforehand, the differences to KTS are that (i) $v \equiv 0 \mod 6$ and (ii) there is one "class" with pairs instead of triplets. In the chapter on construction methods we will discuss methods to create Nearly Kirkman Triple Systems.

At the end of 3.2 we will introduce and simplify a general upper bound $M_k(v)$.

### 3.7 Definition

A **Kirkman packing design** of order $v$ and length $r$, or $KP(v,r)$, is a set of blocks (being subsets of $v$ objects) and their partition into $r$ subsets called rounds, with the properties:

(i) each object occurs exactly once per round,

(ii) all blocks in each round are triplets except for at most one, and that one can contain 2 or 4 objects and

(iii) each object pair occurs in at most one block in the design.

### Definition/Notation in other literature

Perfect Stranger Matching [10]

In economics the concept of **Perfect Stranger Matching (PSM)** is identical with all the resolvable designs: RBIBD joined with packing designs of different block sizes. Two participants meet at most once. They call a parallel class a *group allocation* and a collection of parallel classes is similar to a *sequence*. A PSM configuration $(p,g)$ is a design with $p = v$ and $g = k$.

### 3.8 Example

One $KP(8,4)$ (which does not yield $v \equiv 0 \mod 6$) has the four rounds

\[
\begin{align*}
\{1,2,3\} & \quad \{1,4,7\} & \quad \{1,5\} & \quad \{1,6,8\} \\
\{4,5,6\} & \quad \{2,5,8\} & \quad \{2,6,7\} & \quad \{2,4\} \\
\{7,8\} & \quad \{3,6\} & \quad \{3,4,8\} & \quad \{3,5,7\}
\end{align*}
\]

A $KP(8,3)$ would be possible, as

\[
\begin{align*}
\{1,2,3\} & \quad \{1,4,7\} & \quad \{1,5\} \\
\{4,5,6\} & \quad \{2,5,8\} & \quad \{2,6,7\} \\
\{7,8\} & \quad \{3,6\} & \quad \{3,4,8\}
\end{align*}
\]

but this is obviously not the maximum amount of rounds.
One KP(18,8) has eight rounds [37]:

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>{A, B, C}</td>
<td>{A, b, c}</td>
<td>{a, b, C}</td>
<td>{a, B, c}</td>
<td></td>
</tr>
<tr>
<td>{D, E, F}</td>
<td>{D, e, F}</td>
<td>{d, e, F}</td>
<td>{d, E, f}</td>
<td></td>
</tr>
<tr>
<td>{G, H, I}</td>
<td>{G, h, I}</td>
<td>{g, h, I}</td>
<td>{g, H, i}</td>
<td></td>
</tr>
<tr>
<td>{a, h, f}</td>
<td>{a, H, F}</td>
<td>{A, H, f}</td>
<td>{A, h, F}</td>
<td></td>
</tr>
<tr>
<td>{d, b, i}</td>
<td>{d, B, i}</td>
<td>{D, B, i}</td>
<td>{D, b, I}</td>
<td></td>
</tr>
<tr>
<td>{g, e, c}</td>
<td>{g, E, c}</td>
<td>{G, E, c}</td>
<td>{G, e, C}</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>{A, D, G}</td>
<td>{A, d, g}</td>
<td>{a, d, G}</td>
<td>{a, D, g}</td>
<td></td>
</tr>
<tr>
<td>{B, E, H}</td>
<td>{B, e, h}</td>
<td>{b, e, H}</td>
<td>{b, E, h}</td>
<td></td>
</tr>
<tr>
<td>{C, F, I}</td>
<td>{C, f, i}</td>
<td>{c, f, I}</td>
<td>{c, F, i}</td>
<td></td>
</tr>
<tr>
<td>{a, e, i}</td>
<td>{a, E, I}</td>
<td>{A, E, i}</td>
<td>{A, e, I}</td>
<td></td>
</tr>
<tr>
<td>{b, f, g}</td>
<td>{b, F, G}</td>
<td>{B, F, g}</td>
<td>{B, f, G}</td>
<td></td>
</tr>
<tr>
<td>{c, d, h}</td>
<td>{c, D, h}</td>
<td>{C, D, h}</td>
<td>{C, d, H}</td>
<td></td>
</tr>
</tbody>
</table>

We would like to know how many rounds would be possible for a given \( v \) in such a KP\((v, r)\).

3.9 Definition Given a KP\((v, r)\) with the maximal possible amount of rounds, then \( r(v) \) is this largest value of \( r \), while \( b(v) \) is the amount or value of unordered pairs of objects covered by a round; and \( rb(v) := r(v) \cdot b(v) \) is the number of pairs covered in the whole KP\((v, r)\).

3.10 Example

The KP\((8,4)\) (see example 3.8) has with \( r(8) = 4 \) the maximum of possible rounds. \( b(8) = 7 \): In the first round we have the unordered pairs \((1,2)\), \((1,3)\), \((2,3)\), \((4,5)\), \((4,6)\), \((5,6)\), \((7,8)\). \( rb(8) = r(8) \cdot b(8) = 28 \).

The KP\((18,8)\) yields \( r(18) = 8, b(18) = 18 \).

The following lemma provides the upper bound \( M(v) \) for a design.
3.11 Lemma Given KP\((v, r)\) or \((v, k, 1)\)-BIBD. Then \(M(v)\) is the upper bound for rounds, then

\[
r(v) \leq M(v) := \left\lfloor \frac{v(v - 1)}{2 \cdot b(v)} \right\rfloor
\]

Proof: The number of all possible ordered pairs would be \(v \cdot (v - 1)\); unordered we have \(\frac{v(v - 1)}{2} \geq b(v)\) unordered pairs. Thus, \(M(v)\) is the upper bound for rounds.

3.12 Definition If a Kirkman packing design reaches the upper bound \(M(v)\), i.e. \(r(v) = M(v)\), it is an optimal Kirkman packing design.

The question is, for which \(v\) do optimal Kirkman packing designs exist, i.e. \(r(v) = M(v)\).

Take \(v, v \equiv 0 \mod 6, v \geq 18\). The best design we can achieve is one where every point \(a\) meets every other point in a block except one other point \(a'\), and this pair \(aa'\) cannot occur (Nearly Kirkman Triple Systems, see 3.18). Why is this the best design? For the current \(v\) we have \(3|v\). Hence, in one round all points can fit into \(\frac{v}{3}\) blocks of size 3 and in every block there are three points and three unordered pairs, so \(b(v) = v\). This reduces the upper bound to \(M(v) = \left\lfloor \frac{v-1}{2} \right\rfloor \) \(v \equiv \frac{n}{2} - 1\). Now we need to show that \(M(v) = r(v)\). If every point would meet every other point, we would have \(\frac{v(v-1)}{2} \) unordered pairs in our design, but every point \(a\) is in a pair \(aa'\) which cannot occur:

\[
\frac{v(v - 1)}{2} - \frac{v}{2} = \frac{v(v - 2)}{2} \leq b(v)
\]

\[
\frac{v - 2}{2} \leq r(v)
\]

\[
r(v) \geq \frac{v}{2} - 1
\]

This yields the lemma:

3.13 Lemma If \(v\) is an even multiple of 3, i.e. \(v \equiv 0 \mod 6\), then

\[
r(v) = M(v) = \frac{v}{2} - 1
\]
In this subsection we explored Kirkman packing designs KP(v,r) and their maximum amount of rounds, \( r(v) \), and concluded \( r(v) \) reaches the upper bound \( M(v) = \frac{v}{2} - 1 \) for \( v, v \equiv 0 \mod 6 \). Now we need to prove that these designs exist.

**Nearly Kirkman Triple Systems**

**3.14 Definition** Given a graph G. A *matching* is a set of pairwise independent edges of G. If the set covers all vertices, we call it a *perfect matching* or *1-factor*. A spanning subgraph in which every connected component is a triangle is called a *triangle-factor*.

Informally, in a design a 1-factor is a parallel class with blocks of size 2 and a triangle-factor a parallel class with blocks of size 3.

**3.15 Definition** A *Nearly Kirkman Triple System* of order \( v \), or NKTS(v), is a set of blocks (being subsets of \( v \) objects) with \( v \equiv 0 \mod 6 \) and a partition of the blocks into one 1-factor and \( \frac{v}{2} - 1 \) triangle-factors with the properties:

(i) each object occurs exactly once per triangle-factor or 1-factor and

(ii) each object pair occurs in exactly one block in the design

This leads immediately to the following corollary.

**3.16 Corollary** Given a NKTS(v). Delete the 1-factor to obtain an optimal Kirkman packing design.

**3.17 Example**

Take the KP(18,8) from example 3.8 and add the 1-factor

\[
\{A, a\} \quad \{B, b\} \\
\{C, c\} \quad \{D, d\} \\
\{E, e\} \quad \{F, f\} \\
\{G, g\} \quad \{H, h\} \\
\{I, i\}
\]

obtain a NKTS(v).

Rees and Wallis [32] and Rees and Stinson [29] proved the theorem:
3.18 Theorem There exists a Nearly Kirkman Triple System NKTS(v) if and only if $v \equiv 0 \mod 6$ and $v \geq 18$

The upper bound $M(v)$ for arbitrary $k$

The upper bound $M(v)$ is already defined for $k=3$. For $v \equiv 0 \mod 6$ it is: $v/2 - 1$. For $v \equiv 3 \mod 6$ it is: $(v-1)/2$. Now we will generalize those results.

We recall Lemma 3.11: If $M(v)$ is the upper bound for rounds, then

$$r(v) \leq M(v) = \left\lfloor \frac{v(v-1)}{2 \cdot b(v)} \right\rfloor$$

In terms of graph theory, each block of a design can be shown as a complete graph $K_k(V,E)$ with $|E| = \frac{k(k-1)}{2}$ and $|V| = k$, where any point represents a point of the design and an edge represents that its points are connected in the block.

Generalizing definition 3.9, we define that $b_k(v)$ is the amount of unordered pairs of objects covered by a round. In other words, $b_k(v)$ is the amount of pairs covered in a block times the blocks in a round:

$$b_k(v) = \frac{k(k-1)}{2} \cdot \frac{v}{k} = \frac{v(k-1)}{2}$$

This gives us the upper bounds

$$M_k(v) = \left\lfloor \frac{v(v-1)}{2 \cdot b_k(v)} \right\rfloor = \left\lfloor \frac{v-1}{k-1} \right\rfloor$$

Both et al. describe $M_k(v)$ to be a trivial upper bound [10], not specifying it further. We have shown and proven, that this is indeed the maximum number of rounds for such a design and at this moment we are unaware of any proof in the literature yet.

We will give some specific examples for some $k$ which are sufficient for practical purposes.

(i) $M_2(v) = v - 1$

(ii) $M_3(v) = \left\lfloor \frac{v-1}{2} \right\rfloor = \begin{cases} v/2 - 1 & \text{if } v \text{ even} \\ (v-1)/2 & \text{if } v \text{ odd} \end{cases}$
(iii) 
\[ M_4(v) = \left\lfloor \frac{v - 1}{3} \right\rfloor = \begin{cases} 
\frac{(v - 1)}{3} & \text{if } v \equiv 1 \mod 3 \\
\frac{(v - 2)}{3} & \text{if } v \equiv 2 \mod 3 \\
\frac{(v - 3)}{3} & \text{if } v \equiv 0 \mod 3
\end{cases} \]

(iv) 
\[ M_5(v) = \left\lfloor \frac{v - 1}{4} \right\rfloor \]

(v) 
\[ M_6(v) = \left\lfloor \frac{v - 1}{5} \right\rfloor \]

Both et al. state in their paper [10] that the 'longest known sequences are shorter than this trivial upper limit' and for a design with \( v = 24 \) and \( k = 4 \) they illustrate that a maximum length of \( r = 6 \) has been found. In 2007 Pegg [37] already gave such a design with \( k = 4 \) and \( r(24) = 7 \).
Chapter 4

Existence of Designs

The question, which designs exist, has fascinated mathematicians since the 1850’s with Kirkman’s schoolgirl problem [21]:

Fifteen young ladies in a school walk out three abreast for seven days in succession: it is required to arrange them daily so that no two shall walk twice abreast.

We give an overview of the conditions a BIBD necessarily has and show whether those conditions are sufficient. We will do the same for resolvable BIBD. After that we will have a look at Kirkman Packing Designs. The end of this section will be about families, whose existence has been proven, showing why this is quite a difficult field to explore: The variety of proven BIBD and Kirkman Packing Designs is quite sparse.

4.1 Conditions

The following results, which hold for general BIBD, have been summarized by Ray-Chaudhuri [28].

4.1 Proposition Given a \((v, k, \lambda)\)-BIBD, the following conditions are necessary

\[
\lambda(v - 1) \equiv 0 \mod (k - 1)
\]

\[
\lambda v(v - 1) \equiv 0 \mod k(k - 1)
\]

These conditions are just another way to express \(r\) and \(b\) as in 3.5. If \(k = 3\) and \(\lambda = 1\), we call the BIBD Steiner Triple System.
4.2 Corollary (Steiner Triple Systems) For the special case of a $(v,3,1)$-BIBD the necessary conditions of 4.1 reduce to:

$$v \equiv 1 \text{ or } 3 \mod 6$$

In 1975 Wilson [34] proved the following theorem.

4.3 Theorem (Wilson, 1975 [34]) Given $\lambda$ and $k$, there exists $v_0(k, \lambda)$ such that a $(v,k,\lambda)$ design exists for all $v > v_0(k, \lambda)$ satisfying 4.1 (in other literature the necessary condition 4.1 is said to be *asymptotically sufficient*).

If $\lambda = 1$, $k$ is a prime power, 4.1 reduces to

$$v \equiv 1 \text{ or } k \mod k(k - 1)$$

Because $v$ must be a multiple of $k$ in resolvable BIBD (every parallel class has block size $k$ and contains all points), resolvable BIBD have an *additional* necessary condition.

4.4 Proposition Necessary condition for resolvable BIBD

$$v \equiv 0 \mod k$$

For $\lambda = 1$, this reduces with 4.1 to the following.

4.5 Proposition Combination of all necessary conditions for resolvable BIBD for $\lambda = 1$

$$v \equiv k \mod k(k - 1)$$

So, Wilson proved for $v$ larger than a specific threshold and satisfying 4.1 that a $(v,k,\lambda)$-BIBD exists, but this does not yet say anything about resolvable BIBD. However, Lu proved in 1984 [24] similar to Wilson the following theorem.
4.6 Theorem (Lu, 1984 [24]) Given $\lambda$ and $k$, there exists $v_0(k, \lambda)$ such that a resolvable $(v, k, \lambda)$ design exists for all $v > v_0(k, \lambda)$ satisfying 4.1 and 4.3.

Fisher first proved another necessary condition for BIBD in 1940 [14], given in the following theorem.

4.7 Theorem (Fisher, 1940 [14]) Necessary condition for a BIBD

$$b \geq v$$

If $b = v$ we call the BIBD to be *symmetric*\(^1\). The important points of this section are the propositions and the proves of Wilson and Lu [24, 34] showing that BIBD - resolvable BIBD, respectively - do exist, if $v$ is big enough and satisfies the necessary condition(s).

### 4.2 Discovered Families for RBIBD

The existence of particular kinds of BIBD turned out to be quite a mystery. Some families have been discovered, but this is still a sparse field. 4.1 is *sufficient* for $k = 3, 4$ (Hanani, 1961 [15]) and $k = 5$ except $(15,5,2)$ [17] and for $k = 6$ if $\lambda > 1$ (Hanani 1975 [19]). This allows the construction of a variety of BIBD. However, this does *not* mean that their subsets of *resolvable BIBD* also exist. 4.4 ($v \equiv 0 \mod (k - 1)$) is still just a necessary condition and need not be sufficient for RBIBD.

A much shorter overview of proven resolvable designs with 4.1 ($\lambda(v - 1) \equiv 0 \mod (k - 1)$ and $\lambda v (v - 1) \equiv 0 \mod k(k - 1)$) and 4.4 as sufficient condition is [2, ch. 9.4]:

<table>
<thead>
<tr>
<th>$(k, \lambda)$</th>
<th>Condition on $v$</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(2, 1)$</td>
<td>$v = 2n, n \in \mathbb{N}$</td>
<td>Even res. BIBD</td>
</tr>
<tr>
<td>$(3, 1)$</td>
<td>$v \equiv 3 \mod 6$</td>
<td>Kirkman Triple Systems</td>
</tr>
<tr>
<td>$(4, 1)$</td>
<td>$v \equiv 4 \mod 12$</td>
<td>Hanani, 1972[16]</td>
</tr>
<tr>
<td>$(3, 2)$</td>
<td>$v \equiv 0 \mod 3$</td>
<td>Hanani, 1974[18]</td>
</tr>
<tr>
<td>$(4, 3)$</td>
<td>$v \equiv 0 \mod 4$</td>
<td>[2, ch. 13]</td>
</tr>
<tr>
<td>$(6, 10)$</td>
<td>$v \equiv 0 \mod 6$</td>
<td>Baker, 1983[6]</td>
</tr>
</tbody>
</table>

\(^1\)A design is symmetric, if and only if $b = v$ and thus $r = k$; see figure 31 as an example
In fact, the Bruck-Ryser Theorem showed in 1949 [11] that 4.4 cannot be always sufficient, as \( v = k^2 \) satisfies the condition but for \( k = 6 \) there is no resolvable BIBD. Let’s have a closer look at a particular design.

### 4.2.1 Kirkman Triple Systems

#### 4.8 Definition
If we set \( k = 3 \) and \( \lambda = 1 \), we get the family of \((v, 3, 1)\)-BIBD, also called Steiner Triple Systems. The resolvable BIBD we want to focus on are resolvable \((v, 3, 1)\)-BIBD, also called Kirkman Triple Systems (KTS\((v)\)).

#### 4.9 Corollary
A KTS\((v)\) can only exist if \( v \equiv 3 \mod 6 \).

The corollary follows immediately from 4.5. The arising question is: Is this necessary condition also sufficient? Indeed, Ray-Chaudhuri and Wilson proved in 1971 [2, ch. 9.2] that \( v \equiv 3 \mod 6 \) is also a sufficient condition.

#### 4.10 Theorem
(Ray-Chaudhuri & Wilson, 1971 [2]) There exists a Kirkman Triple System of order \( v \) if and only if \( v \equiv 3 \mod 6 \).

### 4.2.2 \((v, 4, 1)\) Designs

#### 4.11 Theorem
A resolvable \((v, 4, 1)\)-BIBD exists if and only if \( v \equiv 4 \mod 12 \).

The proof is similar to the one for the Kirkman Triple Systems (see [16]). Unfortunately, there are very few construction methods known for this family (see [3]).

### 4.2.3 \((2n, 2, 1)\) Designs

The German men’s football league Bundesliga is the perfect example for a \((2n, 2, 1)\) resolvable BIBD. There are 18 teams, so we have a \((18, 2, 1)\) BIBD. Each matchday corresponds to a parallel class of the BIBD and each team plays against each other team once. This whole procedure happens twice, also known as Vor- und Rückrunde or first and second half of the season. The same method we present in 5.3 is indeed used to get the match schedule of the Bundesliga [38].
4.12 Proposition (2n, 2, 1) resolvable BIBD exist for all \( n \in \mathbb{N} \).

The corresponding proof can be found in section 5.3.

4.3 Discovered Families for KP

In [32] and [29] the following theorem is proven in different ways.

4.13 Theorem There exists a Nearly Kirkman Triple System of order \( v \) if and only if \( v \equiv 0 \mod 6, \ v \geq 18 \).

There is no NKTS(6) and no NKTS(12) [23]. The best KP we can get for those \( v \) are KP(6,1) and a KP(12,4).

4.14 Example

A KP(12,4):

- \{1, 5, 9\} \quad \{1, 7, 12\}
- \{2, 6, 10\} \quad \{2, 8, 11\}
- \{3, 7, 11\} \quad \{3, 5, 10\}
- \{4, 8, 12\} \quad \{4, 6, 9\}
- \{1, 6, 11\} \quad \{1, 8, 10\}
- \{2, 5, 12\} \quad \{2, 7, 9\}
- \{3, 8, 9\} \quad \{3, 6, 12\}
- \{4, 7, 10\} \quad \{4, 5, 11\}
Chapter 5

Construction Methods

In this section, we present some construction methods for the introduced designs. We will explore difference methods in two basic approaches and analyze whether they can help us in getting our resolvable \((v,k,1)\)-BIBD.

5.1 General Methods - Difference Methods

Difference methods are divided in the construction of difference sets and difference systems. The basic idea of those concepts is to take one or more initial blocks and increment their points \(\mod v\) to get the other blocks of a BIBD. In practical ways, this naturally leads to the question of how to obtain the initial blocks. One of the most important techniques, which makes use of the Galois Fields, will be presented below. At the end of the subsections we will verify whether the concepts can be used to create resolvable BIBD.

5.1.1 Difference Sets

The following definitions are shown by Anderson [2].

**5.1 Definition:** A \((v,k,\lambda)\) difference set (mod \(v\)) or a cyclic \((v,k,\lambda)\) difference set is a set \(D = \{d_1, ..., d_k\}\) of distinct elements of \(Z_v\) such that each non-zero \(d \in Z_v\) can be expressed in the form \(d = d_i - d_j\) in precisely \(\lambda\) ways.

**5.2 Example:** \(\{1, 2, 4\}\) is a cyclic \((7,3,1)\) difference set.

**5.3 Definition:** We can obtain translates \(D+a\) of a difference set by adding an integer \(a\) to all of \(D\)'s elements: \(\{d_1 + a, ..., d_k + a\}\).
Of course every translate is also a difference set. Taking all the translates with \( a \in \{0, ..., v-1\} \) gives us all the blocks of a symmetric \((v,k,\lambda)\)-BIBD.

Now, what if we change our set of elements from \( \mathbb{Z}_v \) to an additive abelian group? Example: Let’s say \( \mathbb{Z}_4 \oplus \mathbb{Z}_4 \). Here we have ordered pairs \((i,j)\), where \( i, j \in \mathbb{Z}_4 \). Addition is defined mod 4 by \((i, j) + (k, l) = (i+k, j+l)\). The set \( D = \{(0,0), (0,2), (1,0), (3,0), (1,1), (3,3)\} \) is a (16,6,2) difference set.

5.4 Definition: We define a \((v,k,\lambda)\) difference set in an additive abelian group \( G \) of order \( v \) as a set \( D = \{d_1, ..., d_k\} \) of distinct elements of \( G \) such that each non-zero element \( g \) of \( G \) has exactly \( \lambda \) representations as \( g = d_i - d_j \).

We know, if we have a difference set, we win: We can build the blocks of a symmetric design. However, this is not necessarily a resolvable block design! But, how do we get such difference sets?

**Quadratic residue difference sets**

According to Ian Anderson [2], "One of the most important constructions of difference sets makes use of the quadratic residues or squares in \( \text{GF}(q)^1 \)."

We will have a look at the example of a \((7,3,1)\) BIBD. The \( \text{GF}(7) \) or \( \mathbb{F}_7 \) has the following multiplication table:

<table>
<thead>
<tr>
<th>x</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>1</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>3</td>
<td>6</td>
<td>2</td>
<td>5</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>4</td>
<td>1</td>
<td>5</td>
<td>2</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>5</td>
<td>3</td>
<td>1</td>
<td>6</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

The non-zero squares are bold. The primitive element \( \theta \) is 3 (\( \theta^1 \equiv 3, \theta^2 \equiv 2, \theta^3 \equiv 6, \theta^4 \equiv 4, \theta^5 \equiv 5, \theta^6 \equiv 1 \)). We can see that the non-zero squares are exactly the \( \theta^x \) with \( x \) even (also bold). The non-zero non-squares are the other elements 3,6,5.

---

\(^1\)\( \text{GF}(q) \) is a finite or Galois field with \( q = p^a, p \) prime
Anderson [2, ch. 2.2, 2.1.6] states that if \( p^\alpha \equiv 3( \text{ mod } 4) \), then the non-zero squares in GF(q) form a \((q, \frac{1}{2}(q-1), \frac{1}{4}(q-3))\) difference set.\(^2\)

A constraint - by design - of the difference sets so far is the fact that \( b=v \) (\#blocks == \#elements), because the translates (\( \equiv \) blocks) of a difference set are built by developing them \( \mod v \). Assuming \( \lambda = 1 \), this leads to

\[
v = b = \frac{v(v-1)}{k(k-1)} \iff k(k-1) = v-1 \iff v = k(k-1) + 1
\]

However, for resolvable BIBD also \( v \equiv k \mod k(k-1) \) needs to hold. We see, \( 1 \neq k \), thus the constructed symmetric designs will never be resolvable. The question is, how to construct such non-symmetric designs with more blocks than points.

### 5.1.2 Difference Systems

![Figure 5.1: (13,3,1)-BIBD](image)

The sets \{1,2,5\} and \{1,3,9\} give all the differences

The example in figure 5.1 has two sets instead of one. Those two sets together combine all differences \( \lambda \) times.

**5.5 Definition:** Let \( D_1, \ldots, D_t \) be sets of size \( k \) in an additive abelian group \( G \) of order \( v \), with differences arising from \( D_i \) give each non-zero element exactly \( \lambda \) times. Then they form a \((v, k, \lambda)\) difference system.

\(^2\)The designs built by those difference sets are called Hadamard designs
5.6 Corollary: Let $D_1, ... D_t$ form a $(v, k, \lambda)$ difference system in the additive abelian group $G = \{g_0, ..., g_{v-1}\}$. Then the sets $D_i + g_j$, $1 \leq i \leq t$, $0 \leq j \leq v-1$ are the blocks of a $(v, k, \lambda)$ design.

There are further theorems presented by Anderson [2, ch. 2.3], but they are not suitable for the construction of resolvable designs. The question now is, if the difference systems constructed by the previous approaches are even applicable for resolvable BIBD. We know that one difference set generates $v$ blocks, two sets generate $2 \cdot v$ blocks, and so forth. For $b$ blocks we need $t = \frac{v}{k}$ sets (because $b = \frac{v}{k} \cdot r$), which is equivalent to $t = \frac{(v-1)}{k(k-1)}$ sets. Given $\lambda = 1$, this only holds for $v \equiv 1 \mod k(k-1)$, but the necessary condition for resolvable BIBD $v \equiv k \mod k(k-1)$ is never met. Consequently, no resolvable BIBD can be constructed with the difference systems so far.

At least not in their original way. However, we present methods using those basic methods during this chapter.

5.2 Methods for $k=3$

The two first approaches methods we want to present construct only BIBD which are not resolvable. However, these are two powerful methods and sometimes applied in other methods to construct resolvable designs.

A mutation of the PBD approach is used in the Sage algorithms [36] and the method of pure and mixed differences is adapted later for RBIBD. We present these approaches to give a better knowledge of our practically useful methods.

5.2.1 Pure and Mixed Differences

There is also a slightly different approach to construct BIBD, compared to difference systems, which helps covering the differences in a BIBD more easily. If we once again have a look at the $(9, 3, 1)$-RBIBD, we can change the points $\{1, 2, ..., 9\}$ to

$$\{0_1, 1_1, 2_1, 0_2, 1_2, 2_2, 0_3, 1_3, 2_3\}$$

with the suffix $i$ ($0 < i \leq 3 = t$) and the elements $G = \mathbb{Z}_3$. Consider the initial blocks

$$\{1_1, 2_1, 0_2\}, \{1_2, 2_2, 0_3\}, \{1_3, 2_3, 0_1\}, \{0_1, 0_2, 0_3\}.$$
Now we need to cover the differences within the groups between elements with a fixed suffix (e.g. \{0_1, 1_1, 2_1\}); we call them pure differences. The pure (1,1) differences are 1_1 - 2_1 = 2 and 2_1 - 1_1 = 1, i.e. all non-zero differences. Same for the (2,2) and (3,3) differences.

The differences in between the groups need to be covered as well. We call them mixed differences. Those differences between elements of different suffixes additionally need to contain the zeros: The mixed (1,2) differences are 1_1 - 0 = 1 and 2_1 - 0 = 2 in the first block and 0 - 0 = 0 in the last block. This covers everything. We get the differences for (2,1), (1,3), (3,1), (2,3) and (3,2) in the same way and thus have all pure and mixed differences between the elements within the groups and between the groups covered.

If we create the translates of the initial blocks (with fixed suffixes), we get the other eight blocks:

\[
\begin{align*}
\{2_1, 0_1, 1_2\}, \{2_2, 0_2, 1_3\}, \{2_3, 0_3, 1_1\}, \{1_1, 1_2, 1_3\}, \\
\{0_1, 1_1, 2_2\}, \{0_2, 1_2, 2_3\}, \{0_3, 1_3, 2_1\}, \{2_1, 2_2, 2_3\}.
\end{align*}
\]

We now give some more explanations about the origins of this idea to understand the theorem at the end of this method better. We recall that \(v\) is the number of points in a block design, \(k\) the size of the subsets, \(b\) the number of blocks and that each point occurs in \(r\) blocks. This results in

\[
\begin{align*}
\lambda(v - 1) &= r(k - 1) \quad (1.1) \\
KTS(v) &= v - 1 = 2r \quad (2.1) \\
v &= 2r + 1 \quad (3.1) \\
bk &= vr \quad (1.2) \quad \Rightarrow \quad 3b = vr \quad (2.2) \quad \Rightarrow \quad b = \frac{r(2r+1)}{3} \quad (3.2)
\end{align*}
\]

Because in (3.2) we have \(b\) as an integer, so \(r\) must be of the form \(3t\) or \(3t + 1\).

Considering (1.1) and (1.2), the following is clear:

\[
\begin{align*}
r &= 3t + 1 \quad \Rightarrow \quad v = 6t + 3, \quad b = (3t + 1)(2t + 1) \quad (4.1) \\
r &= 3t \quad \Rightarrow \quad v = 6t + 1, \quad b = t(6t + 1) \quad (4.2)
\end{align*}
\]

R.C. Bose gives a construction method in [9, ch. 4.2] for \(v = 6t + 3\). In this construction method, we make use of Mixed and Pure Differences.

It is \(v = 6t + 3\) and we take the residue classes of ( mod \(2t + 1\)). To get all the \(6t + 3\) differences, consider the pairs

\([1, 2t], [2, 2t - 1], \ldots, [t, t + 1]\)
The $i$th pair $[i, 2t+1-i]$ gives the differences $2t+1-2i$ and $2i$. Therefore we get all the classes of $(mod 2t+1)$ except 0 as differences.

5.7 Theorem: Given $v \equiv 3 \mod 6$ and thus having $v$ of the form $v = 6t + 3$, $t \in \mathbb{N}$, then $t = \frac{v-3}{6}$. The initial blocks

$$
\{1, 2t_1, 0_2\}, \{2_1, (2t_1-1)_1, 0_2\}, \ldots, \{t_1, (t+1)_1, 0_2\},
$$

$$
\{1_2, 2t_2, 0_3\}, \{2_2, (2t_2-1)_2, 0_3\}, \ldots, \{t_2, (t+1)_2, 0_3\},
$$

$$
\{1_3, 2t_3, 0_1\}, \{2_3, (2t_3-1)_3, 0_1\}, \ldots, \{t_3, (t+1)_3, 0_1\},
$$

$$
\{0_1, 0_2, 0_3\}
$$

and their translates for fixed suffixes form a mixed difference system which constructs STS($v$).

Unfortunately, the blocks constructed cannot be partitioned into parallel classes and the STS($v$) is not resolvable, but we will use a similar method later.

5.2.2 PBD Approach

Anderson showed [2, ch. 8.2] that a pairwised balanced design PBD($u$,\{3,4\},1) exists for all $u \equiv 0$ or $1 \mod 3$, $u \neq 6$ (see definition 3.1).

$$
v \equiv 1 \text{ or } 3 \mod 6 \Leftrightarrow v = 2u + 1, u \equiv 0 \text{ or } 1 \mod 3
$$

Given $v$ and the corresponding PBD($u$,\{3,4\},1). The $2u+1$ elements of our desired STS($v$) are the $\infty$-sign and two copies $x_1, x_2$ of all elements $x \in \{1, \ldots, u\}$. We take the two copies $x_1, x_2$ of each $x \in \{1, \ldots, u\}$ and $\infty$, having now $2u + 1 = v$ elements. With each block $\{a, b, c\}$ with size 3 we construct a STS(7) with $S = \{\infty, a_1, a_2, b_1, b_2, c_1, c_2\}$, giving us 7 blocks. With each block $\{a, b, c, d\}$ with size 4 we construct a STS(9) with $S = \{\infty, a_1, a_2, b_1, b_2, c_1, c_2, d_1, d_2\}$, giving us 12. All the blocks joined together define our STS($v$).

5.8 Example: Steiner Triple System of order 21 (STS(21))

We want the blocks of a STS (21). $21 = 2 \cdot 10 + 1$. So we need a PBD(10,\{3,4\},1): \{1, 2, 3\}, \{4, 8, 9\}, \{5, 6, 7\},
\{1, 5, 8\}, \{3, 4, 6\}, \{2, 7, 9\},
\{1, 4, 7\}, \{2, 6, 8\}, \{3, 5, 9\},
\{1, 6, 9, 10\}, \{2, 4, 5, 10\}, \{3, 7, 8, 10\}.

We choose \(S = \{\infty, 1, \ldots, 20\}\) of size \(v\) and write the copies \(x_1 = x\) and \(x_2 = x + 10\) for better understanding. With every block of size 3 of the PBD, we construct an STS(7), the example \(\{1, 2, 3\}\) leads to the triples:
\{\infty, 11\}, \{\infty, 2, 12\}, \{\infty, 3, 13\}, \{1, 2, 3\}, \{1, 12, 13\}, \{2, 11, 13\}, \{3, 11, 12\}.
With blocks of size 4 applied to \{1, 6, 9, 10\}, we obtain:
\{\infty, 11\}, \{\infty, 2, 16\}, \{\infty, 9, 19\}, \{\infty, 10, 20\},
\{1, 9, 10\}, \{6, 9, 20\}, \{1, 16, 20\}, \{1, 6, 19\},
\{10, 16, 19\}, \{11, 19, 20\}, \{6, 10, 11\}, \{9, 11, 16\}.

If we do this for all blocks and join the results we get our STS(21).

However, this approach has one handicap: Getting such a PBD\((u,\{3,4\},1)\) with \(u = \frac{v-1}{2}\). Interesting is here, whether there is a simple method to get the PBD.

5.2.3 Early Methods

In 1914 Ball [4] summarized the methods so far to construct RBIBD. We will discuss the mbriefly.

**Frost’s Method**
Ball described in [4] *Frost’s method* for KTS\((v)\) with \(v = 2^{2n} - 1, n \in \mathbb{N}\), the first being \(v \in \{3, 15, 63, 255, 1023\}\).
Those are not that many designs for practical purposes and the only interesting KTS(63) will be constructed differently.

**Anstice’s Method Generalized**
Ball also described [4] Anstice’s method for KTS\((v)\) with \(v = 2p + 1, p = 12m + 7, p\) prime or prime power, \(m \in \mathbb{N}\), the first being \(v \in \{15, 39, 63, 87, 135\}\) and a generalized version is introduced in [2] for \(p = 6m + 1\), so the first being \(v \in \{15, 27, 39, 51, 63, 75, 87, 99, 123\}\). This is already better, but still, we will construct those KTS differently with methods which are easier to implement.

**Gill’s Method**
Gill’s method [4] constructs a KTS immediately with some triplets, but there is no
way in constructing the triplets yet. Also the triplets still need to be sorted into their parallel classes.

5.2.4 Walecki’s Method

Walecki [4] takes an already constructed KTS\((v)\) and returns a KTS\((3n)\). We will use this to construct KTS\((v)\) for \(v \in \{27, 45, 63\}\).

(i) We have the blocks of the KTS\((n)\) for \((n-1)/2\) rounds. If we triple the points \(\{1, 2, ..., n\}\) to \(\{a_1, a_2, ..., a_n, b_1, b_2, ..., b_n, c_1, c_2, ..., c_n\}\), we can arrange them the same way to \((n-1)/2\) rounds.

(ii) The other \(n\) rounds will be constructed as follows: on round \(p\) \((p \in \{0, 1, ..., n-1\})\) we take a triplet \(a_q, b_{q+p}, c_{q+2p}\) and obtain the triplets of the round by taking \(q \in \{1, 2, ..., n\}\) and doing the whole suffix mod \(n\).

We introduce the algorithm 1 Walecki which implements the method. Input are the blocks of a KTS\((n)\), \(\text{preblocks[][]}\), sorted after rounds. This algorithm runs in \(\Theta(v^2)\).

5.2.5 Pure and Mixed Method for Resolvable Designs

This method works for KTS\((v)\) with \(v = 3q, q = 6m + 1\) and \(q\) prime or prime power, \(m \in \mathbb{N}\). The first being \(v \in \{21, 39, 57, 93, 111\}\).

The construction method is similar to the one described in 5.2.1, but uses GF. The proof can be found in [2, ch. 2]. Determine the primitive element \(\theta\) and store it with its powers: \(\theta^2, \theta^3, ..., \theta^{6m}\). Then the blocks

\[A = \{0_1, 0_2, 0_3\},\]
\[B_{ij} = \{\theta^i_j, \theta^{i+2m}_j, \theta^{i+4m}_j\}, 1 \leq i \leq m, 1 \leq j \leq 3,\]
\[C_{ij} = \{\theta^i_j, \theta^{i+3m}_j, \theta^{i+5m}_j\}, 1 \leq i \leq m, 1 \leq j \leq 3 \text{ (mod 3)},\]
\[D_{ij} = \{\theta^i_j, \theta^{i+2m}_j, \theta^{i+4m}_j\}, 1 \leq i \leq m, 1 \leq j \leq 3 \text{ (mod 3)}\]

form a mixed difference system.

To obtain the parallel classes, take \(A, B_{ij}, C_{ij}\) to form the first round. Their translates form the next 6\(m\) rounds. In total there are \(r = 9m + 1\) rounds. The 3\(m\) \(D_{ij}\) and their translates form the last parallel classes.

We introduce the algorithm 2 Res. Pure and Mixed implementing the method. This algorithm runs in \(\Theta(v^2)\).
Algorithm 1 Walecki

1: procedure Walecki(preblocks[bn][k])  ▷ The two-dimensional array goes
blocks times blocksize
2: v ← 3 · n
3: blocks[bn][k]  ▷ empty array
4: currentBlock ← 0
5: for i ← 0, i < (n − 1)/2 do  ▷ (i) For each round, triple the blocks
6:     for j ← 0, j < n/k do
7:         currentBlock ← j + i · n/k
8:         blocks[currentBlock][0] ← preblocks[currentBlock][0]
9:         blocks[currentBlock][1] ← preblocks[currentBlock][1]
10:        blocks[currentBlock][2] ← preblocks[currentBlock][2]
11:        blocks[currentBlock + n][0] ← preblocks[currentBlock][0] + n
12:        blocks[currentBlock + n][1] ← preblocks[currentBlock][1] + n
13:        blocks[currentBlock + n][2] ← preblocks[currentBlock][2] + n
14:        blocks[currentBlock + 2n][0] ← preblocks[currentBlock][0] + 2n
15:        blocks[currentBlock + 2n][1] ← preblocks[currentBlock][1] + 2n
16:        blocks[currentBlock + 2n][2] ← preblocks[currentBlock][2] + 2n
17:     end for
18: end for
19: currentBlock = n(n − 1)/6  ▷ (ii) Construct the other n rounds
20: for p ← 0, p < n do
21:     for q ← 1, q ≤ n do  ▷ a_q
22:         blocks[currentBlock][0] ← q%n
23:     end for
24:     blocks[currentBlock][1] ← (q + p)n + n  ▷ b_{q+p}
25:     blocks[currentBlock][2] ← (q + 2p)n + 2n  ▷ c_{q+2p}
26: end for
27: end procedure
Algorithm 2 Res. Pure and Mixed

1: procedure RPnM(v)
2: \[ q \leftarrow v/3 \]
3: \[ m \leftarrow (q - 1)/6 \]
4: \[ \text{blocks}[] \leftarrow \text{new blocks}[b][k] \]
5: \[ cB \leftarrow 0 \quad \triangleright \text{cB = currentBlock} \]
6: \[ pm[] = [\theta^1, \theta^2, \ldots, \theta^{6m}] \quad \triangleright \text{Get the multiples of the primitive element} \]
7: for \[ l \leftarrow 1; l \leq q \] do \[ \triangleright (i) \text{ First } q = 6m + 1 \text{ rounds} \]
8: \[ \text{blocks}[cB][0] \leftarrow l \quad \triangleright 0_1 \text{ and translates for A} \]
9: \[ \text{blocks}[cB][1] \leftarrow l + q \quad \triangleright 0_2 \text{ and translates for A} \]
10: \[ \text{blocks}[cB][2] \leftarrow l + 2q \quad \triangleright 0_3 \text{ and translates for A} \]
11: \[ cB++ \]
12: for \[ i \leftarrow 1; i \leq m \] do \[ \triangleright D_{ij} \text{ and translates} \]
13: \[ \text{for } j \leftarrow 0; j < 3 \] do \[ \triangleright \theta^j_i \text{ and the} \]
14: \[ \text{blocks}[cB][0] \leftarrow pm[i - 1] + j \cdot q + l - 1 \quad \triangleright \theta^i_{j+1} \text{ and the} \]
15: \[ \text{blocks}[cB][1] \leftarrow pm[i - 1 + 2m] + j \cdot q + l - 1 \quad \triangleright \theta^i_{j+2m} \text{ and the} \]
16: \[ \text{blocks}[cB][2] \leftarrow pm[i - 1 + 4m] + j \cdot q + l - 1 \quad \triangleright \theta^i_{j+4m} \text{ and the} \]
17: \[ cB++ \]
18: \[ \text{blocks}[cB][0] \leftarrow pm[i - 1] + (j \cdot q) + l - 1 \quad \triangleright \theta^i_{j+1} \text{ and the} \]
19: \[ \text{blocks}[cB][1] \leftarrow pm[i - 1 + 3m] + ((j + 1) \cdot q) \% 3 + l - 1 \quad \triangleright \theta^i_{j+3m} \text{ and the} \]
20: \[ \text{blocks}[cB][2] \leftarrow pm[i - 1 + 5m] + ((j + 2) \cdot q) \% 3 + l - 1 \quad \triangleright \theta^i_{j+5m} \text{ and the} \]
21: \[ cB++ \]
22: end for
23: end for
24: end for
25: for \[ i \leftarrow 1; i \leq m \] do \[ \triangleright D_{ij} \text{ and translates} \]
26: \[ \text{for } j \leftarrow 0; j < 3 \] do \[ \triangleright \theta^j_i \text{ for } D_{ij} \]
27: \[ \text{blocks}[cB][0] \leftarrow pm[i - 1] + j \cdot q \quad \triangleright \theta^j_i \text{ for } D_{ij} \]
28: \[ \text{blocks}[cB][1] \leftarrow pm[i - 1 + 2m] + (j + 1) \cdot q \quad \triangleright \theta^{j+2m}_{i+1} \text{ for } D_{ij} \]
29: \[ \text{blocks}[cB][2] \leftarrow pm[i - 1 + 4m] + (j + 2) \cdot q \quad \triangleright \theta^{j+4m}_{i+2} \text{ for } D_{ij} \]
30: \[ cB++ \]
31: for \[ l \leftarrow 0; l < v/k - 1 \] do \[ \triangleright \text{Other blocks in a round} \]
32: \[ \text{blocks}[cB][0] = ((\text{blocks}[cB - 1][0] + 1) \mod q) + j \cdot q \]
33: \[ \text{blocks}[cB][1] = ((\text{blocks}[cB - 1][1] + 1) \mod q) + j \cdot q \]
34: \[ \text{blocks}[cB][2] = ((\text{blocks}[cB - 1][2] + 1) \mod q) + j \cdot q \]
35: \[ cB++ \]
36: end for
37: end for
38: end for
39: end procedure
5.2.6 Doubling Method and its Extension

The doubling method and its extension are so far not described in literature. To understand the idea behind it, we use the theorem 5.9, described and proved in detail in [32].

5.9 Theorem: Doubling Construction Given a graph \( G \). If the graph \( G \) admits an edge-decomposition into an even number of triangle-factors, then the graph \( G \otimes I_2 \) admits an edge-decomposition into triangle-factors.

However, we are not aware of any method in the literature which uses this theorem. The basic idea transformed for Kirkman Packing designs is yet easily formulated:

5.10 Corollary: Given a Kirkman Triple System(\( v \)) with an even number \( r \) of rounds, then there exists a KP(\( 2v, 2r \)).

Note that from now on we will use \( v' \) and \( r' \) as the number of points and rounds of the KTS, while \( v \) and \( r \) are the number and rounds of the constructed KP.

We shape this in a new method and introduce:

Method (Doubling Construction) This method takes a KTS(\( v' \)) with \( r \) even (the first being KTS(\( v' \)) with \( v' \in \{9, 21, 33, 45, 57\} \)) and returns a KP(\( v, r \)) (the first being KP(\( 2v', 2v' \)) with \( v \in \{18, 42, 66, 90, 114\} \)).

(i) As long as there are unused rounds, repeat the following steps (ii)-(v)

(ii) Take two unused rounds \( r_1, r_2 \) from the KTS(\( v' \)) and relabel the elements of \( r_2 \) (e.g. \( x \rightarrow x^* \) or \( x \rightarrow x + v \))

(iii) Take each block \((x_1, x_2, x_3)\) and create three additional blocks from it; for a block from \( r_1 \) this is

\[
\begin{align*}
x_1, x_2, x_3 & \rightarrow x_1^*, x_2^*, x_3^* \\
x_1, x_2, x_3 & \rightarrow x_1^*, x_2^*, x_3^* \\
x_1^*, x_2, x_3 & \rightarrow x_1, x_2^*, x_3^* \\
x_1^*, x_2, x_3 & \rightarrow x_1, x_2^*, x_3^*
\end{align*}
\]

and for a block from \( r_2 \) this is the other way round

\[
\begin{align*}
x_1, x_2, x_3 & \rightarrow x_1, x_2^*, x_3^* \\
x_1, x_2, x_3 & \rightarrow x_1, x_2^*, x_3^* \\
x_1^*, x_2, x_3 & \rightarrow x_1^*, x_2, x_3^* \\
x_1^*, x_2, x_3 & \rightarrow x_1^*, x_2, x_3^*
\end{align*}
\]
(iv) Concatenate r1 and r2 to obtain one round of the KP

(v) Use the other $3 \times v/k \times 2$ blocks to obtain another three rounds

Of practical interest is step (v). In example 5.11 we have each line forming a new round of KP(18,8). However, this is not necessarily always the case. There are $3^{2-v/k}$ possibilities to construct rounds and the difficulty is to find the three fitting ones. For a KP(66,32) those are about one sextillion ($10^{21}$) possibilities. One idea we present, is to fix one of the constructed blocks of r1 and go through the columns of r2 and see which are affected and impossible to use. Two of the remaining blocks in these columns are determined fix. With each of these we can find out how they affect the columns in r1 and so on. After some determining we have the first round. If we delete the blocks of the first round from our pool, the same game for the second round of the KP will be much simpler. Delete these blocks as well and we have the third round remaining and are done.

5.11 Example:[32] KTS(9) → KP(18,8)

The rounds of the KTS shall be:

| 123 456 789 |
| 186 429 753 |
| 147 258 369 |
| 159 267 348 |

We always take two rounds and mix them:

| 123456789 | 1*8*6* 4*2*9* 7*5*3* |
| 123456789 | 1*86 4*29 753 |
| 1*2*3* 4*5*6* 7*8*9* | 1 8 6* 4 2 9* 7 5 3* |
| 1*2*3* 4*5*6* 7*8*9* | 1 8*6 4 2*9 7 5*3 |
| 147258369 | 1*5*9* 2*6*7 3*4*8* |
| 147258369 | 1*5 9 2*6 7 3*4 8 |
| 1*4*7* 2*5*8* 3*6*9* | 1 5 9* 2 6*7 3 4 8* |
| 1*4*7* 2*5*8* 3*6*9* | 1 5*9 2 6*7 3 4*8 |

Now each line forms a round of the KP(18,8).

The offered doubling method only worked for KTS with r’ even. We will change it
slightly and introduce a new method which additionally gives us almost perfect results for KTS with odd $r'$:

**Method (Pseudo-Doubling Construction)** takes a KTS with $r'$ even or odd. This method additionally takes KTS($v'$) with $v'$ odd (the first being KTS($v'$) with $v' \in \{15, 27, 39, 51, 63\}$) and returns the additional KP($2v',2r'-1$) (the first being KP($2v',2r'-1$) with $v \in \{30, 54, 78, 102, 126\}$).

(i) While there is *more than one unused round left* repeat the following steps as in the original method

(ii) Take two unused rounds $r_1, r_2$ from the KTS($v'$) and relabel the elements of $r_2$ (e.g. $x \rightarrow x^*$ or $x \rightarrow x + v$)

(iii) Take each block $(x_1, x_2, x_3)$ and create three additional blocks from it

(iv) Concatenate $r_1$ and $r_2$ to obtain one round of the KP

(v) Use the other $3 \times v/k \times 2$ blocks to obtain another three rounds

(vi) If there is one unused round left (i.e. $r'$ of the KTS is odd), then copy this last round, relabel the copied elements and concatenate the round with the copy to obtain one last round of the KP($2v',2r'-1$)

Obviously, this method does not yield the maximum number of rounds for odd $r$, but one less. This gives more rounds than an in-depth search. That the method is faster, is most likely, but depends on the algorithm.

### 5.2.7 Construction of specific KTS and KP

We will construct some specific KTS and KP directly, as the shown and introduced methods cannot create all of the wanted designs.

**KTS(33):** $r = 16, b = 176$, see [2, ch. 9.1].
These are the blocks of the KTS. This solution gets its blocks from the first six initial blocks (bold) and develops them mod 32. The first is used for 15 translates, the other 5 for 31 translates, giving all 176 blocks.
We present a quick overview for all \( v \) with \( 3|v, v \leq 66 \). \( r^* \) is the biggest number of rounds Both et al. [10] could get. With the methods and algorithms we presented we can, in most cases, construct more rounds than Both et al. \( v = 66 \) is far bigger than the \( v \) Both et al. took into account and should suit most practical purposes in the design of experiments. Unfortunately, \( v \in \{ 24, 51, 54, 60 \} \) requires theorems which are not as easily transformed to computable methods than the ones we

\[ \text{KP}(30,14) \]: \( b = 140 \), see [5].
Points are \( (Z_7 \times \{ 1, 2, 3, 4 \}) \cup \{ \infty_1, \infty_2 \} \).
The first initial parallel class:

\[
\begin{array}{cccccccc}
4_1, 5_2, 0_3 & 6_1, 3_2, 4_3 & 1_1, 6_2, 5_4 & 5_1, 1_2, 6_4 & 2_1, 6_3, 0_4 \\
0_1, 1_3, 3_4 & 3_1, 5_3, 2_4 & 0_2, 3_3, 1_4 & 2_2, 2_3, \infty_1 & 4_2, 4_4, \infty_2 \\
\end{array}
\]

The second initial parallel class:

\[
\begin{array}{cccccccc}
0_1, 2_2, 6_3 & 6_1, 5_2, 1_4 & 0_2, 5_3, 4_4 & 1_2, 0_3, 3_4 & 1_1, 2_1, 4_1 \\
3_2, 4_2, 6_2 & 1_3, 2_3, 4_3 & 6_4, 0_4, 2_4 & 5_1, 5_4, \infty_1 & 3_1, 3_3, \infty_2 \\
\end{array}
\]

The other parallel classes are obtained by developing the first two classes \( \mod 7 \) and have the suffixes and \( \infty \) fixed.

\[ \text{KP}(36,17) \]: \( b = 204 \), see [32].
Points are \( (Z_{17} \times \{ 1, 2 \}) \cup \{ \infty_1, \infty_2 \} \). Develop the initial class \( \mod 17 \).

\[
\begin{array}{cccccccc}
2_1, 7_2, 8_2 & 11_1, 14_2, 4_2 & 13_1, 8_1, 3_2 & 16_2, 1_2, 5_2 & 12_1, 10_2, 13_2 & 7_1, 6_2, 15_2 \\
10_1, 14_4, 4_1 & \infty_1, 5_1, 9_2 & 3_1, 12_2, 0_2 & 0_1, 9_1, 11_2 & 15_1, 16_1, 1_1 & 6_1, 2_2, \infty_2 \\
\end{array}
\]

\[ \text{KP}(48,23) \]: \( b = 368 \), see [5].
Points are \( (Z_{23} \cup \infty) \times Z_2 \). Half of the initial parallel class:

\[
\begin{array}{cccccccc}
\infty_0, 14_0, 22_1 & 1_0, 2_0, 4_0 & 5_0, 10_0, 20_0 & 3_0, 7_0, 21_1 \\
6_0, 15_0, 13_1 & 8_0, 19_0, 18_1 & 9_0, 16_0, 12_1 & 11_0, 17_0, 0_1 \\
\end{array}
\]

The other half of the initial class is constructed by developing the suffix \( \mod 2 \):

\[
\begin{array}{cccccccc}
\infty_1, 14_1, 22_0 & 1_1, 2_1, 4_1 & 5_1, 10_1, 20_1 & 3_1, 7_1, 21_0 \\
6_1, 15_1, 13_0 & 8_1, 19_1, 18_0 & 9_1, 16_1, 12_0 & 11_1, 17_1, 0_0 \\
\end{array}
\]

Develop the initial class \( \mod 23 \).

### 5.2.8 Overview

We present a quick overview for all \( v \) with \( 3|v, v \leq 66 \). \( r^* \) is the biggest number of rounds Both et al. [10] could get. With the methods and algorithms we presented we can, in most cases, construct more rounds than Both et al. \( v = 66 \) is far bigger than the \( v \) Both et al. took into account and should suit most practical purposes in the design of experiments. Unfortunately, \( v \in \{ 24, 51, 54, 60 \} \) requires theorems which are not as easily transformed to computable methods than the ones we
presented. For \( v = 54 \) the method of Pseudo-Doubling can give enough rounds for
a normal use in experiments.

<table>
<thead>
<tr>
<th>( v )</th>
<th>( M(v) )</th>
<th>( r^* )</th>
<th>Descr.</th>
<th>Method</th>
<th>Constructed or alg. exists</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>4</td>
<td>4</td>
<td>KTS</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>12</td>
<td>4</td>
<td>4</td>
<td>KP</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>15</td>
<td>7</td>
<td>7</td>
<td>KTS</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>18</td>
<td>8</td>
<td>8</td>
<td>KP</td>
<td>Doubling</td>
<td>✓</td>
</tr>
<tr>
<td>21</td>
<td>10</td>
<td>9</td>
<td>KTS</td>
<td>RPnM</td>
<td>✓</td>
</tr>
<tr>
<td>24</td>
<td>11</td>
<td>10</td>
<td>KP</td>
<td>[32]</td>
<td>✓</td>
</tr>
<tr>
<td>27</td>
<td>13</td>
<td>11</td>
<td>KTS</td>
<td>Walecki</td>
<td>✓</td>
</tr>
<tr>
<td>30</td>
<td>14</td>
<td>13</td>
<td>KP</td>
<td>directly constructed</td>
<td>✓</td>
</tr>
<tr>
<td>33</td>
<td>16</td>
<td>14</td>
<td>KTS</td>
<td>directly constructed</td>
<td>✓</td>
</tr>
<tr>
<td>36</td>
<td>17</td>
<td>15</td>
<td>KP</td>
<td>directly constructed</td>
<td>✓</td>
</tr>
<tr>
<td>39</td>
<td>19</td>
<td>17</td>
<td>KTS</td>
<td>RPnM</td>
<td>✓</td>
</tr>
<tr>
<td>42</td>
<td>20</td>
<td>-</td>
<td>KP</td>
<td>Doubling</td>
<td>✓</td>
</tr>
<tr>
<td>45</td>
<td>22</td>
<td>-</td>
<td>KTS</td>
<td>Walecki</td>
<td>✓</td>
</tr>
<tr>
<td>48</td>
<td>23</td>
<td>-</td>
<td>KP</td>
<td>directly constructed</td>
<td>✓</td>
</tr>
<tr>
<td>51</td>
<td>25</td>
<td>-</td>
<td>KTS</td>
<td>Generalized Anstice</td>
<td>Possible with Anstice</td>
</tr>
<tr>
<td>54</td>
<td>26</td>
<td>-</td>
<td>KP</td>
<td>Pseudo-Doubling for ( r=25 )</td>
<td>[23]</td>
</tr>
<tr>
<td>57</td>
<td>28</td>
<td>-</td>
<td>KTS</td>
<td>RPnM</td>
<td>✓</td>
</tr>
<tr>
<td>60</td>
<td>29</td>
<td>-</td>
<td>KP</td>
<td>[32]</td>
<td>✓</td>
</tr>
<tr>
<td>63</td>
<td>31</td>
<td>-</td>
<td>KTS</td>
<td>Walecki</td>
<td>✓</td>
</tr>
<tr>
<td>66</td>
<td>32</td>
<td>-</td>
<td>KP</td>
<td>Doubling</td>
<td>✓</td>
</tr>
</tbody>
</table>

### 5.3 Method for \( k=2 \)

We will construct \((2n, 2, 1)\) resolvable BIBD with the help of difference systems.
We already proved that difference systems cannot create a resolvable BIBD in their
original way, so we will change the method. We take \( v' = v - 1 = 2n - 1 \) points
and construct a \((2n - 1, 2, 1)\) BIBD with difference sets and later add the blocks
with the \(2n\)th point to complete the BIBD to a \((2n, 2, 1)\) resolvable BIBD.

The \((2n-1, 2, 1)\) BIBD has \( v' = 2n-1, k' = 2, r' = 2n-2 \) and \( b' = (2n-1)\cdot(n-1) = 2n^2 - 3n + 1 \). We need difference sets to cover all the non-zero \( v' - 1 = 2n - 2 \) differences:

\[
\{0, 2\}, \{2n - 2, 3\}, ... \{n + 1, n\}
\]
Those sets give the differences:

\[(2,2n-1-2), (4,2n-1-4), \ldots (2n-2,1)\]

Figure 5.2 shows the difference sets and their translates for the case of the \((7,2,1)\) BIBD. Now we have a BIBD for \(v' = 2n - 1\). We add the \(2n^{th}\) point \(\infty\) to this BIBD. This is simply done by adding the block \(\{\infty, 1\}\) to our difference sets which, together with this block, form our initial blocks for the \((2n,2,1)\) resolvable BIBD. While on the other initial blocks both integers are incremented \(\mod 2n - 1\), we develop in \(\{\infty, 1\}\) only the integer (how to increment \(\infty\) anyway?). Figure 5.3 shows the initial blocks on the left followed by the translates giving all the blocks of the \((8,2,1)\) resolvable BIBD. The remaining question is, if those BIBD are resolvable. Indeed, the initial blocks and their translates each correspond to a parallel class. Also \(v \equiv k \mod k(k-1) \equiv 2 \mod 2\), i.e. \(v\) even, is the case.

Figure 5.2: \((7,2,1)\)-BIBD, on the upper left the difference sets, followed by their translates
Figure 5.3: (8,2,1)-BIBD, on the upper left the initial blocks, followed by their translates
Chapter 6

Discussion

The goal was to find for arbitrary \( k \) and \( v \) with \( k|v \) methods and algorithms to construct designs.

To achieve the goal in finding methods to construct rounds we did an intensive literature research and found concepts which suit our needs: RBIBD and optimal KP. We discovered that they have already been proven (RBIBD: \( k \in \{2, 3, 4\}, \lambda = 1 \) and optimal KP for \( v \geq 18 \)). We discovered that the approaches to get BIBD are not directly reusable for RBIBD. However, ideas of one method are reused in one of the explained methods: Galois fields and the method of pure and mixed differences are adapted for RBIBD. Other methods (Frost, Gill, Anstice) are mentioned, but do not give a lot of designs and are hard to be transformed into algorithms. Walecki’s method and the method of pure and mixed differences for resolvable designs cover more RBIBD and we contributed two algorithms to describe them. For KP, we took the theorem of doubling construction and contributed a method with specific practical ideas and an extension to use it. We then presented some other direct constructions for \( k = 3 \) with an overview, showing that most KTS and KP until \( v = 66 \) can be constructed by the methods described, only four need deeper research as in how to specifically construct them. We described a method to get all RBIBD with \( k = 2 \), only methods for \( k > 3 \) were not found. However, it seems that Sage uses an algorithm to construct a small family of designs with \( k = 4 \).

Not that successful were our attempts in combining designs of different sizes for \( k = 3 \) and in combining two \((v, 2, 1)\) to one \((2v, 4, 1)\). Shuffling two designs with \( k = 3 \) together is difficult. If we try to combine any design with three points and shuffle them after the first round, the three points replace the elements of the other design and we combine them in a block. These three connections cannot be used later, thus making specific blocks in later rounds unusable. This destroys the structure of the design with holey classes. Of course, maybe there is a pattern creatable such that we combine the holey classes together in a different way. Another idea was to merge two \((v, 2, 1)\) to one \((2v, 4, 1)\)-RBIBD. This thought did not bring the desired designs, as there were no patterns in the blocks of the \((v, 2, 1)\) after which we would have been able to group them as new blocks. Our original idea was to merge six blocks of a \((v, 2, 1)\) into one block of a \((2v, 4, 1)\). But without
a transferable pattern, a block of a \((v, 2, 1)\) is simply a necessary connection in a \((2v, 4, 1)\)-RBIBD, thus not helpful. New methods are obviously not that easily invented.

All together, we covered most of the topics related to RBIBD: PBD, BIBD, RBIBD and their characteristics: Blocks, rounds and existence. We collected and described various of their construction methods with focus on \(k = 2\) and \(k = 3\) and gathered methods or direct examples for the \(v\) with \(k = 3\) and \(v \leq 66\), which are far more than Both et al. used in this overview [10]. We did the same steps for KP and collected the material in more extensive amounts and with more examples and explanations than the current literature and provided definitions for KP and NKTS, which are also not clearly defined in the literature so far. Furthermore, the notation for designs is different in every area and the field of economics names concepts completely different than mathematicians do. We contributed unification of these notations. We merged the mathematical expertise with the current economical standards in the design of experiments in a way, such that non-mathematicians can grasp the ideas and concepts of RBIBD more easily. We improved the definition and derivations for the upper bound of rounds \(M(v)\) which will help economics and others to improve their algorithms with a clearly defined and proven maximal number of rounds \(M(v)\). Both et al. [10] wrote that "Even small problem instances of the SGP [Social Golfer Problem] and, hence of PSM, are computationally expensive, due to the inability to determine if the maximum number of matches have been found." As we have proven, this \(M(v)\) is very clear now. Then we took the theorem of doubling construction and created a method. We extended this method to obtain more optimal KP with non-perfect but in practice sufficient results. We even exemplified on two methods (Walecki and RPnM) how algorithms made of these methods look like. So we see, there are many possibilities to construct designs and we showed that most of the smaller ones for blocks of size 3 and all of block size 2 can be constructed easily.

There are still topics and areas related to our designs which need to be taken into account: The Oberwolfach problem as well as Sage with its code and theorems from Design Theory [7]. Furthermore, we have ideas in mind which might give us more families of KP: Merging two KP instead of KTS in the doubling construction or applying Walecki’s method on KP. There might be quite some methods which can be transformed and combined to construct different families of designs. The code from Sage seems to produce all KTS\((v)\), so that is definitely worth another look. Also, there are the methods of Kotzig and Rosa [23], which even the mathematician Pegg [37] could not reproduce anymore. All these methods and concepts are for \(v\) with \(k|v\). However, there are non-optimal KP and also Kirkman covering designs,
which might be interesting for practical purposes as well. Those are out of scope, but might be interesting for the design of other experiments.
Chapter 7

Conclusion

This overview provides the foundation to explore the practicality of RBIBD and optimal KP in the design of experiments in order to create designs with fixed block size $k$ and $v$ points, $k|v$. We collected designs, examined their usefulness, discussed methods to construct them, created a method, designed algorithms to show how to implement them and compared the number of rounds of our results with the rounds created by Both et al. [10]. The found material was then sorted, compared to the latest results in the field of economics and explored for practicality.

In this thesis, the focus was on collecting and organizing the different topics regarding everything which is related to RBIBD and their role and use in the design of experiments. We clarified the mathematical restrictions and possibilities of those designs and the Kirkman packing designs, recognizing that the practical issues are easy to understand, but difficult to solve. For this, we collected methods; some of them being important in general for BIBD, most of them being used to construct RBIBD or KP and sometimes depending on the general methods or using them in some way. We saw that for $k = 2$ all RBIBD can be constructed with ease,. For $k = 3$ quite some methods do construct a lot of them. An overview shows the results of this thesis regarding $k = 3$. We did transform some of the methods into algorithms, also we clarified the upper bound for arbitrary $k$, giving a proof and presenting the results for various $k$. Further, we transformed the theorem of doubling construction, introduced a method and showed the difficulties and possibilities for improvement in this method. We introduced an extension to this method to construct a family of non-optimal Kirkman packing designs with $r = M(v) - 1$. For bigger $k$ this field is mostly unexplored and needs more attention to gain practically useful methods. We see that the presented methods have the potential to construct their designs a lot faster than any randomized or in-depth search.
Bibliography


  http://www.mathpuzzle.com/MAA/54-Golf%20Tournaments/mathgames_08_14_07.html

[38] *Spielplan* (accessed 2017-12-24).
  https://de.wikipedia.org/wiki/Spielplan__(Sport)
Appendix

Further Examples and Concepts

Pure and Mixed Differences: KTS(21)

An example for the method of pure and mixed differences is the KTS(21) (see theorem 5.8): $v = 21$, thus $t = \frac{21 - 3}{6} = 3$

The initial blocks are

$[1_1, 6_1, 0_2], [2_1, 5_1, 0_2], [3_1, 4_1, 0_2], [1_2, 6_2, 0_3], [2_2, 5_2, 0_3], [3_2, 4_2, 0_3], [1_3, 6_3, 0_1], [2_3, 5_3, 0_1], [3_3, 4_3, 0_1], [0_1, 0_2, 0_3]$

Their six translates for the fixed suffixes are:

$[2_1, 0_1, 1_2], [3_1, 6_1, 1_2], [4_1, 5_1, 1_2], [2_2, 0_2, 1_3], [3_2, 6_2, 1_3], [4_2, 5_2, 1_3], [2_3, 0_3, 1_1], [3_3, 6_3, 1_1], [4_3, 5_3, 1_1], [1_1, 1_2, 1_3], [3_1, 1_1, 2_2], [4_1, 0_1, 2_2], [5_1, 6_1, 2_2], [3_2, 1_2, 2_3], [4_2, 0_2, 2_3], [5_2, 6_2, 2_3], [3_3, 1_3, 2_1], [4_3, 0_3, 2_1], [5_3, 6_3, 2_1], [2_1, 2_2, 2_3], [4_1, 2_1, 3_2], [5_1, 1_1, 3_2], [6_1, 0_1, 3_2], [4_2, 2_2, 3_3], [5_2, 1_2, 3_3], [6_2, 0_2, 3_3], [4_3, 2_3, 3_1], [5_3, 1_3, 3_1], [6_3, 0_3, 3_1], [3_1, 3_2, 3_2], [5_1, 3_1, 4_2], [6_1, 2_1, 4_2], [0_1, 1_1, 4_2], [5_2, 3_2, 4_3], [6_2, 2_2, 4_3], [0_2, 1_2, 4_3], [5_3, 3_3, 4_1], [6_3, 2_3, 4_1], [0_3, 1_3, 4_1], [4_1, 4_2, 4_3], [6_1, 4_1, 5_2], [0_1, 3_1, 5_2], [1_1, 2_1, 5_2], [6_2, 4_2, 5_3], [0_2, 3_2, 5_3], [1_2, 2_2, 5_3], [6_3, 4_3, 5_1], [0_3, 3_3, 5_1], [1_3, 2_3, 5_1], [5_1, 5_2, 5_3], [0_4, 5_1, 6_2], [1_1, 4_1, 6_2], [2_1, 3_1, 6_2], [0_2, 5_2, 6_3], [1_2, 4_2, 6_3], [2_2, 3_2, 6_3], [0_3, 5_3, 6_1], [1_3, 4_3, 6_1], [2_3, 3_3, 6_1], [6_1, 6_2, 6_3].
These are 70 blocks. A check \( b = \frac{\lambda(v-1)}{k-1} = \frac{21 \cdot 20}{3 \cdot 2} = 70 \) confirms that this is the amount of blocks needed.

Unfortunately, if we try to partition the blocks into parallel classes, we will fail.

**Latin Squares and MOLS**

The following concepts are necessary for the theoretical background and proofs of BIBD and KP.

**A.1 Definition** A Latin Square on \( n \) symbols (or: of order \( n \)) is an \( n \times n \) array such that each symbol occurs exactly once in each row and column.

**A.2 Definition** Two Latin Squares \( A \) and \( B \) on \( n \) symbols are called orthogonal, if the ordered pairs \((a_{ij}, b_{ij})\) are all different.

**A.3 Example**

Take two Latin Squares of order 3

\[
A = \begin{bmatrix}
a_{11} & a_{12} & a_{13} \\
a_{21} & a_{22} & a_{23} \\
a_{31} & a_{32} & a_{33}
\end{bmatrix} = \begin{bmatrix}
1 & 2 & 3 \\
2 & 3 & 1 \\
3 & 1 & 2
\end{bmatrix}
\]

\[
B = \begin{bmatrix}
b_{11} & b_{12} & b_{13} \\
b_{21} & b_{22} & b_{23} \\
b_{31} & b_{32} & b_{33}
\end{bmatrix} = \begin{bmatrix}
1 & 2 & 3 \\
3 & 1 & 2 \\
2 & 3 & 1
\end{bmatrix}.
\]

Superposing them, we get the ordered pairs \((1,1), (2,2), (3,3)\), \((2,3), (3,1), (1,2)\). We have all possible ordered pairs, so there can be no third Latin Square orthogonal to both \( A \) and \( B \). Hence, \( N(3) = 2 \) (see definition A.4).

**A.4 Definition** If the Latin Squares \( A_1, A_2, \ldots, A_k \) are such that \( A_i \) is orthogonal to \( A_j \) whenever \( i \neq j \), we call it a set of \( k \) mutually orthogonal Latin Squares (MOLS). Special interest is paid in \( N(n) \), which we define as the biggest \( k \) for a set of order \( n \), i.e. the largest possible set of MOLS of order \( n \).

A and B from example A.3 form a set of 2 MOLS.