# MAKING THE MOLS TABLE

**Charles J. Colbourn**
*** Jeffrey H. Dinitz**

*Department of Combinatorics and Optimization*
*University of Waterloo*
*Waterloo, Ontario, CANADA N2L3G1*

** Department of Mathematics and Statistics*
*University of Vermont*
*Burlington, Vermont 05405 U.S.A.*

## ABSTRACT

This paper is not to be read by the faint of heart. No proofs are given, but it contains statements of a truly alarming number of constructions for transversal designs and incomplete transversal designs.

The paper is a record of an attempt to construct tables of the best results implied by known constructions for the existence of certain classes of mutually orthogonal latin squares and incomplete latin squares.

Sections §1–6 establish the mathematical background for the paper. We begin with basic definitions in §1. Then the following five sections state a fairly complete collection of construction techniques. It may well be impossible to write a complete list of variants of known constructions, and it is certainly beyond reason to do so. We content ourselves with a large battery of the constructions that have been exploited in the literature.

In §7–9, we describe a package developed in MAPLE which instantiates most (but not all) of the constructions in code. Issues in the design of this package are addressed, and a discussion of the architecture of the package is given.

## 1 BASIC DEFINITIONS

First, we must establish the mathematical framework in which we work.

A *transversal design* of *order* or *groupsize* $n$, *blocksize* $k$ and *index* $\lambda$, denoted $TD_\lambda(k,n)$, is a triple $(V, \mathcal{G}, \mathcal{B})$, where

1. $V$ is a set of $kn$ elements;
2. $\mathcal{G}$ is a partition of $V$ into $k$ classes (called *groups*), each of size $n$;
3. $\mathcal{B}$ is a collection of $k$–subsets of $V$ (called *blocks*);
4. every unordered pair of elements from $V$ is either contained in exactly one group, or is contained in exactly $\lambda$ blocks, but not both.

When $\lambda = 1$, one writes simply $TD(k,n)$.

A $TD(k,n)$ is equivalent to the existence of $k-2$ mutually orthogonal latin squares of order $n$, and the various generalizations of transversal designs all have reasonably natural interpretations in that formulation (and also in a third disguise, as "orthogonal arrays"). We have chosen to remain with one notation as much as possible, and have chosen here to use the language of transversal designs. (But see [114] and [115] to convert to the other notations, if desired.)

An *incomplete transversal design* of *order* or *groupsize* $n$, *blocksize* $k$, *index* $\lambda$, and *holesizes* $b_1, \ldots, b_s$, denoted $ITD_\lambda(k,n; b_1, \ldots, b_s)$ for short, is a quadruple $(V, \mathcal{G}, \mathcal{H}, \mathcal{B})$, where

1. $V$ is a set of $kn$ elements;
2. $\mathcal{G}$ is a partition of $V$ into $k$ classes (*groups*), each of size $n$;
3. $\mathcal{H}$ is a set of disjoint subsets $H_1, \ldots, H_s$ of $V$, with the property that, for each $1 \le i \le s$ and each $G \in \mathcal{G}$, $|G \cap H_i| = b_i$;
4. $\mathcal{B}$ is a collection of $k$–subsets of $V$ (*blocks*);
5. every unordered pair of elements from $V$ is
   - contained in a hole, and contained in no blocks; or
   - contained in a group, and contained in no blocks; or
   - contained in neither a hole nor a group, and contained in $\lambda$ blocks.

When $\sum_{i=1}^{s} b_i = n$, an $ITD(k,n; b_1, \ldots, b_n)$ is a *partitioned ITD*, here denoted by $PITD(k,n; b_1, \ldots, b_s)$. We often write the list $b_1, \ldots, b_s$ in "exponential notation", so that $x_1^{y_1} \cdots x_s^{y_s}$ signifies that there are $y_i$ holes of order $x_i$, for each $1 \leq i \leq s$.

Again, when $\lambda = 1$, it can be omitted from the notation. Another notation that we employ for an $ITD_\lambda(k, n; b_1, \ldots, b_s)$ is $TD_\lambda(k, n) - \sum_{i=1}^{s} TD_\lambda(k, b_i)$. It is trivial that the hole sizes could in fact be written in any order, and so when one refers to a specific hole size such as $b_1$, one is really speaking of an arbitrary hole size.

## 2 FILLING, TRUNCATION AND INFLATION

First we introduce the easiest of the constructions, treating some basic equivalences.

**Lemma 2.1** *A $TD(k, n)$ is equivalent to an $ITD(k, n; b_1, \ldots, b_s)$ for any nonnegative integers $b_1, \ldots, b_s$ with $\sum_{i=1}^{s} b_i \leq 1$.*

Actually, holes of order 0 can be assumed present or absent to suit our purposes. Holes of size 1, on the other hand, can always be assumed absent if we choose, because the hole can always be replaced by a block; one cannot, however, assume them to be present unless there is a suitable block available for deletion to form the hole. It is convenient to make a simple convention which avoids treating holes of size 0 and 1 as special cases in each result, namely:

**Convention 2.2** *For all nonnegative integers $k$, there exists a $TD(k, 0)$ and a $TD(k, 1)$.*

More than one hole of size one can occasionally be assumed:

**Lemma 2.3** *[25] A $ITD(k, n; h)$ is equivalent to a $ITD(k, n; h, 1, 1)$ when $(k-1)h < n$. In particular, a $TD(k, n)$ is equivalent to a $TD(k, n) - 3TD(k, 1)$ when $k \leq n$.*

At this point, it is sensible to remark on the basic necessary condition:

**Lemma 2.4** *An $ITD(k,n;h)$ exists only if $h = n$ or $(k-1)h \leq n$. When $(k-1)h = n$, all blocks have exactly one point in the hole.*

The case when $h = n$ corresponds to an incomplete transversal design which has no blocks at all, just one big hole. Nevertheless, to be explicit, we state the following:

**Convention 2.5** *For $n$ a positive integer and $k$ a nonnegative integer, there exists a $TD(k,n) - TD(k,n)$.*

When $(k-1)h+1 = n$, simple counting shows that each element not in the hole lies on exactly one block that does not meet the hole. Deleting these blocks, we obtain:

**Lemma 2.6** *An $ITD(k,(k-1)h+1;h)$ is equivalent to a $PITD(k,(k-1)h+1;h^1 1^{(k-2)h+1})$.*

Lemma 2.3 has a number of generalizations. One can remark, for example, that a simple greedy strategy always produces $1 + \lfloor \frac{n-1}{k-1} \rfloor$ disjoint blocks in a $TD(k,n)$, which can improve upon Lemma 2.3 when $k$ is "small" relative to $n$. In addition, we can examine what happens when there are two or more holes assumed:

**Lemma 2.7** *An $ITD(k,n;b_1,\ldots,b_s)$ always satisfies $(k-1)b_1 + b_2 \leq n$ (in particular, this holds when $b_1$ and $b_2$ are orders of the largest and second largest hole, respectively). Moreover, the ITD always has a block missing the first two holes, unless $(k-1)b_1 + b_2 = n$, $b_2 = b_3 = \ldots = b_s$ and $n = \sum_{i=1}^{s} b_i$. Consequently, an $ITD(k,n;b_1,b_2)$ with $b_1 \geq b_2 > 0$ is equivalent to an $ITD(k,n;b_1,b_2,1)$.*

We collect some other easy constructions in three main categories.

## 2.1 Filling

The basic result for filling an incomplete transversal design is:

**Lemma 2.8** *If an* $ITD(k, n; b_1, \ldots, b_s)$ *exists, and an* $ITD(k, b_1; a_1, \ldots, a_r)$ *exists, then an* $ITD(k, n; a_1, \ldots, a_r, b_2, \ldots, b_s)$ *exists.*

For partitioned $ITD$, one can fill in a more general way:

**Lemma 2.9** *Suppose there is a* $PITD(k, n; b_1, \ldots, b_s)$. *Let* $\varepsilon$ *be a nonnegative integer, and suppose that, for each* $2 \leq i \leq s$, *there is a*

$$TD(k, b_i + \varepsilon) - TD(k, \varepsilon).$$

*Then there exists a* $TD(k, n + \varepsilon) - TD(k, b_1 + \varepsilon)$.

## 2.2 Truncation

Truncation is the operation of removing some points from a group. Here we examine the simplest form of truncation, when all points in a single group are deleted.

**Lemma 2.10** *If a* $TD(k + 1, n)$ *exists, then a* $TD(k, n) - nTD(k, 1)$ *exists.*

**Lemma 2.11** *If a* $TD(k + 1, n; h)$ *exists, then a* $TD(k, n) - TD(k, h) - (n - h)TD(k, 1)$ *exists.*

Removing a level of an $ITD$ also has quite a useful consequence, which has been little exploited previously:

**Lemma 2.12** *Suppose that an* $ITD(k+1, n; b_1, \ldots, b_s)$ *exists and* $\sum_{i=1}^{s} b_i < n$. *Suppose further that, for* $1 \leq i \leq s$, *there exists* $TD(k, b_i)$. *Then there exists a* $TD(k, n) - nTD(k, 1)$.

*If instead there exists* $TD(k, b_i)$ *for* $2 \leq i \leq s$, *then there exists a*

$$TD(k, n) - TD(k, b_1) - (n - b_1)TD(k, 1).$$

A somewhat similar operation can be done with partitioned $ITD$:

**Lemma 2.13** *Suppose that a $PITD(k+1, n; b_1, \ldots, b_s)$ exists, with $b_1 > 0$. Suppose further that, for $2 \le i \le s$, there exists $TD(k, b_i)$. Then there exists a $TD(k, n) - TD(k, b_1) - (n - b_1)TD(k, 1)$.*

## 2.3 Inflation

The main form of inflation is a simple direct product:

**Lemma 2.14** *Suppose that an $ITD(k, n; b_1, \ldots, b_s)$ and a $TD(k, w)$ both exist. Then a $ITD(k, wn; wb_1, \ldots, wb_s)$ exists.*

It bears frequent repetition that filling followed by inflation is *weaker* than inflation followed by filling. To see that it is no stronger, it suffices that each ingredient can be inflated by the same factor and the filling operation remains possible. That it is on occasion weaker follows from the fact that the $ITD$ on $wn$ points may exist, even when the $ITD$ on $n$ points does not exist.

# 3 PBD, GDD AND THE BOSE–SHRIKHANDE–PARKER THEOREM

A *pairwise balanced design* of *order* $v$ and *blocksizes* $K$, denoted $(v, K) - PBD$, is a pair $(X, \mathcal{D})$. $X$ is a set of $v$ elements, and $\mathcal{D}$ is a set of subsets (*blocks*) of $X$ for which $|D| \in K$ for each $D \in \mathcal{D}$. For every 2-subset of elements $\{x, y\} \subset X$, there is exactly one block containing $x$ and $y$.

A *clear set* in a $PBD$ is a set of pairwise disjoint blocks (also called a *partial parallel class*). A *near clear set* in a $PBD$ $(X, \mathcal{D})$ is a subset $\mathcal{D}^c \subseteq \mathcal{D}$ defined as follows. For every block $D \in \mathcal{D}^c$, there is a distinguished element $e_D \in D$, the *tip* of $D$. The set $\mathcal{D}^c$ is near clear if, for each $x \in X$ contained in $\iota_x$ blocks of $\mathcal{D}^c$, $x$ is the tip of at least $\iota_x - 1$ blocks in $\mathcal{D}^c$.

This definition seems unnaturally complicated, so perhaps some examples are needed. A set of blocks that all intersect in a single element, and are otherwise pairwise disjoint, is near clear: Simply taking the common element to be the tip of each block. A different example arises from three blocks which pairwise intersect in one point, but the common intersection is empty. Choosing the three intersection points to be the tips of the three blocks shows that this structure is near clear.

Let us denote a $(v, K) - PBD$ $(X, \mathcal{D})$ with a near clear set $\mathcal{D}^c$ as a $(v, K_b, K_c) - PBD$, where $K_c$ is the sizes of the blocks that actually arise in the near clear set, and $K_b$ is the sizes of the blocks that actually arise among the remaining blocks (note that $K_b \cup K_c \subseteq K$, but equality is *not* necessary, as a $(v, K) - PBD$ need not in general realize every block size in $K$).

## 3.1 First Constructions Using PBD and GDD

Now to the basic (Bose–Shrikhande–Parker) construction:

**Theorem 3.1** *Suppose that a* $(v, K_b, K_c) - PBD$ *exists. Suppose that, for every* $m \in K_b$*, there exists a* $PITD(k, m; 1^m)$*. Further suppose that, for every* $m \in K_c$*, there exists a* $TD(k, m)$*. Then there exists a* $TD(k, v)$*.*

Theorem 3.1 is a fairly standard Wilson-type construction using weight $k$ for pairwise balanced designs [97]. The unusual feature is the use of near clear sets rather than clear sets. We content ourselves with remarking that for blocks in the near clear set, the ingredient used is actually a $TD(k, m) - TD(k, 1)$, and the $TD(k, 1)$ is chosen to coincide with the $k$ copies of the tip element. In this vein, when ingredients exist with more than one hole of size 1, one could permit the blocks of the near clear set to have more than one tip; this would extend the definition of near clear set. However, we know of no applications of this generalization, so we omit it.

When we have a certain types of near clear sets (clear sets being one example), we can say something about incomplete $TD$ as well:

**Theorem 3.2** *Let* $(X, \mathcal{D})$ *be a* $PBD$ *of order* $v$*. Suppose that for some subset* $\mathcal{D}^c \subseteq \mathcal{D}$ *of blocks, we have that there is one element* $x \in X$*, so that for* $D, D' \in \mathcal{D}^c$*,* $D \cap D' \subseteq \{x\}$*. Suppose that for every* $D \in \mathcal{D} \setminus \mathcal{D}^c$*, there exists a* $PITD(k, |D|; 1^{|D|})$*. Fix a block* $F \in \mathcal{D}^c$*, and suppose that for every block* $D \in \mathcal{D}^c \setminus \{F\}$*, there exists a* $TD(k, |D|)$*.*

*Then there exists a* $TD(k, v) - TD(k, |F|)$*.*

Next we constrain the near clear set to be a clear set to obtain:

**Theorem 3.3** *Let* $(X, \mathcal{D})$ *be a* $PBD$ *of order* $v$. *Suppose that some subset* $\mathcal{D}^c \subseteq \mathcal{D}$ *of blocks is a clear set. Suppose that for every* $D \in \mathcal{D} \setminus \mathcal{D}^c$, *there exists a* $PITD(k, |D|; 1^{|D|})$. *Then there exists a* $TD(k, v) - \sum_{F \in \mathcal{D}^c} TD(k, |F|)$.

*In fact, letting* $f = \sum_{F \in \mathcal{D}^c} |F|$, *we obtain the stronger conclusion that there exists the partitioned* $ITD$

$$TD(k, v) - (v - f)TD(k, 1) - \sum_{F \in \mathcal{D}^c} TD(k, |F|).$$

**Theorem 3.4** *Let* $(X, \mathcal{D})$ *be a* $PBD$ *of order* $v$. *Suppose that some subset* $\mathcal{D}^c \subseteq \mathcal{D}$ *of blocks is a clear set. Let* $F \in \mathcal{D} \setminus \mathcal{D}^c$. *Suppose that for every* $D \in \mathcal{D} \setminus \mathcal{D}^c$, *there exists a* $PITD(k, |D|; 1^{|D|})$. *Further suppose that for every* $D \in \mathcal{D} \setminus (\mathcal{D}^c \cup \{F\})$, *there exists a* $TD(k, |D|)$. *Then there exists a* $TD(k, v) - TD(k, |F|)$.

When the clear set is spanning (i.e., the union of the blocks is the set $X$ of all elements in the $PBD$, or it is a *parallel class*), more flexibility exists. We introduce the appropriate language. A *group-divisible design* of *order* $v$, *blocksizes* $K$, and *type* $T = (t_1, \ldots, t_g)$, denoted $(v, K) - GDD$ of type $T$, is a triple $(X, \mathcal{C}, \mathcal{D})$, where $X$ is a set of $v$ elements, $\mathcal{C}$ is a partition of $X$ into $g$ classes (*groups*) $G_1, \ldots, G_g$, where $|G_i| = t_i$; and $\mathcal{D}$ is a set of subsets (*blocks*) of $X$, with the property that when $D \in \mathcal{D}$, we find $|D| \in K$. Moreover, every pair of elements appears together exactly once, either in a group or in a block. Often the type is written in exponential notation.

Now a $(v, K) - PBD$ is equivalent to a $(v, K) - GDD$ of type $1^v$. A $TD(k, n)$ is equivalent to a $(kn, \{k\}) - GDD$ of type $n^k$. Also, converting the groups of a $(v, K) - GDD$ of type $T$ into blocks, we obtain a $(v, K \cup T) - PBD$ in which the images of the groups form a parallel class of blocks. Then restating Theorem 3.3 when the clear set is a parallel class is equivalent to:

**Theorem 3.5** *Suppose that there is a* $(v, K) - GDD$ *of type* $T = (t_1, \ldots, t_s)$. *Suppose that, for each* $m \in K$, *there is a* $PITD(k, m; 1^m)$. *Then there exists a* $PITD(k, v; t_1, \ldots, t_s)$.

Theorem 3.5 shows that one can employ the presence of a single parallel class. How can we use the presence of further parallel classes?

A $PBD$ or $GDD$ with element set $X$ and block set $\mathcal{B}$ is *resolvable* if $\mathcal{B}$ can be partitioned into parallel classes. The partitioning into parallel classes is a *resolution*. Intermediate between $GDD$ and resolvable $PBD$, we may have a $PBD$ in which some, but not all, of the blocks are partitioned into parallel classes. Since resolvable $PBD$ form a special case, we treat this more general situation.

**Theorem 3.6** *Let $(X, \mathcal{D})$ be a $PBD$ of order $v$. Suppose that $\mathcal{D}$ is partitioned into $r+1$ classes $\mathcal{D}_1, \ldots, \mathcal{D}_{r+1}$, where $\mathcal{D}_i$ is a parallel class for $1 \leq i \leq r$, and $\mathcal{D}_{r+1}$ is arbitrary (possibly even empty). Suppose that, for each $D \in \mathcal{D}_{r+1}$, there is a $PITD(k, |D|; 1^{|D|})$.*

*Now for $2 \leq i \leq r$, let $\varepsilon_i$ be a nonnegative integer. Suppose that, for each $2 \leq i \leq r$, and each $D \in \mathcal{D}_i$, there exists a*

$$TD(k, |D| + \varepsilon_i) - TD(k, \varepsilon_i) - |D| \; TD(k, 1).$$

*Let $\sigma = \sum_{i=2}^{r} \varepsilon_i$. Then there exists a*

$$TD(k, v + \sigma) - TD(k, \sigma) - \sum_{D \in \mathcal{D}_1} TD(k, |D|),$$

*a partitioned $ITD$.*

Theorem 3.6 applies equally well to resolvable $GDD$, or $GDD$ with parallel classes; simply treat the groups as blocks forming a parallel class of an equivalent $PBD$.

## 3.2 Incomplete PBD

Group-divisible designs are pairwise balanced designs with a spanning set of holes (the groups). Here we treat pairwise balanced designs with one hole. An *incomplete PBD* of *order* $n$, *blocksizes* $K$, and a *hole* of order $h$ ($(v, h, K) - IPBD$) is a triple $(V, H, \mathcal{B})$. $|V| = v$, $|H| = h$, and $H \subset V$. $\mathcal{B}$ is a set of subsets of $V$, for which $B \in \mathcal{B}$ implies $|B| \in K$. Moreover, $(V, \mathcal{B} \cup \{H\}$ is a $(v, K \cup \{h\}) - PBD$. Since any single block can be taken to form a clear set, we obtain from Theorem 3.3:

**Corollary 3.7** *Suppose there exists a $(v, h, K) - IPBD$. Suppose that for each $m \in K$, there exists a $PITD(k, m; 1^m)$. Then there exists an $ITD(k, v; h)$, and in fact there exists a $PITD(k, v; h^1 1^{v-h})$.*

In a $(v, h, K) - IPBD$ $(V, H, \mathcal{B})$, a *holey parallel class* is a set $\mathcal{P}$ of disjoint blocks, none of which meet the hole, and for which $V = H \cup \bigcup_{P \in \mathcal{P}} P$. One simple way to produce $IPBD$ with a holey parallel class is the following:

**Lemma 3.8** *If there exists a* $(v, K) - GDD$ *of type* $T = (t_1, t_2, \ldots, t_s)$*, then there exists a* $(v, t_1, K \cup \{t_2, \ldots, t_s\}) - IPBD$ *with a holey parallel class with block sizes in* $\{t_2, \ldots, t_s\}$*.*

Later we see other ways to produce $IPBD$ that have many holey parallel classes, so here we examine a method to use their presence:

**Theorem 3.9** *Let* $(V, H, \mathcal{B})$ *be an* $(v, h, K) - IPBD$*. Partition the blocks* $\mathcal{B}$ *into classes* $\mathcal{P}_1, \ldots, \mathcal{P}_r, \mathcal{Q}$ *so that, for* $1 \leq i \leq r$, $\mathcal{P}_i$ *is a holey parallel class. For* $2 \leq i \leq r$*, let* $\varepsilon_i$ *be a nonnegative integer. Now suppose that, for each* $B \in \mathcal{Q}$*, there exists a* $PITD(k, |B|; 1^{|B|})$*. Further suppose that, for each* $2 \leq i \leq r$*, and each* $B \in \mathcal{P}_i$*, there exists a* $PITD(k, |B| + \varepsilon_i; \varepsilon_i^1 1^{|B|})$*. Let* $\sigma = \sum_{i=2}^{r} \varepsilon_i$*. Then there exists a partitioned ITD*

$$TD(k, v + \sigma) - TD(k, \sigma + h) - \sum_{B \in \mathcal{P}_1} TD(k, |B|).$$

Actually, we could take $h = 0$; then the $IPBD$ would be a $PBD$ and the holey parallel classes would be parallel classes. Theorem 3.9 would then reduce to Theorem 3.6.

A $(v, h, K) - IPBD$ can have both parallel classes *and* holey parallel classes. If such an event occurs, we proceed as follows:

**Theorem 3.10** *Let* $(V, H, \mathcal{B})$ *be an IPBD with* $|V| = v$ *and* $|H| = h$*. Suppose that* $\mathcal{B}$ *has a partition into classes* $\{\mathcal{P}_1, \ldots, \mathcal{P}_r, \mathcal{Q}_1, \ldots, \mathcal{Q}_s, \mathcal{R}\}$*, where the* $\{\mathcal{P}_i\}$ *are parallel classes, the* $\{\mathcal{Q}_i\}$ *are holey parallel classes, and* $\mathcal{R}$ *is the remaining set of blocks (possibly empty). Suppose that* $s \geq 1$*. Suppose that, for every* $B \in \mathcal{R}$*, there exists a* $PITD(k, |B|; 1^{|B|})$*.*

*Choose nonnegative integers* $\varepsilon_i$ *for* $1 \leq i \leq r$*, and suppose that, for every* $B \in \mathcal{P}_i$*, there exists a* $PITD(k, |B| + \varepsilon_i; \varepsilon_i^1 1^{|B|})$*. Let* $\sigma = \sum_{i=1}^{r} \varepsilon_i$*.*

*Choose nonnegative integers* $\gamma_i$ *for* $2 \leq i \leq s$*, and suppose that, for every* $B \in \mathcal{Q}_i$*, there exists a* $PITD(k, |B| + \gamma_i; \gamma_i^1 1^{|B|})$*. Let* $\sigma' = \sum_{i=2}^{s} \gamma_i$*.*

*Then two outcomes are possible:*

1. *If there exists a* $TD(k, \sigma + \sigma') - TD(k, \sigma')$*, then there exists a*

$$TD(k, v + \sigma + \sigma') - TD(k, h + \sigma') - \sum_{B \in \mathcal{Q}_1} TD(k, |B|).$$

2. *If there exists a* $TD(k, h + \sigma') - TD(k, \sigma')$*, then there exists a*

$$TD(k, v + \sigma + \sigma') - TD(k, \sigma + \sigma') - \sum_{B \in \mathcal{Q}_1} TD(k, |B|).$$

Some variants are possible, as prior to choosing the two outcomes, we find that two holes, one of size $h + \sigma'$ and the other of size $\sigma + \sigma'$, intersect in $\sigma'$ elements. The last ingredients used to "break the tie" could themselves have holes, which would lead to even more holes in the final result. We do not pursue this.

However, it is necessary to explore what happens when we save back a parallel class instead of a holey parallel class. That leads to the next result:

**Theorem 3.11** *Let* $(V, H, \mathcal{B})$ *be an IPBD with* $|V| = v$ *and* $|H| = h$*. Suppose that* $\mathcal{B}$ *has a partition into classes* $\{\mathcal{P}_1, \ldots, \mathcal{P}_r, \mathcal{Q}_1, \ldots, \mathcal{Q}_s, \mathcal{R}\}$*, where the* $\{\mathcal{P}_i\}$ *are parallel classes, the* $\{\mathcal{Q}_i\}$ *are holey parallel classes, and* $\mathcal{R}$ *is the remaining set of blocks (possibly empty). Suppose that* $r \geq 1$*. Suppose that, for every* $B \in \mathcal{R}$*, there exists a* $PITD(k, |B|; 1^{|B|})$*.*

*Choose nonnegative integers* $\varepsilon_i$ *for* $2 \leq i \leq r$*, and suppose that, for every* $B \in \mathcal{P}_i$*, there exists a* $PITD(k, |B| + \varepsilon_i; \varepsilon_i^1 1^{|B|})$*. Let* $\sigma = \sum_{i=2}^{r} \varepsilon_i$*.*

*Choose nonnegative integers* $\gamma_i$ *for* $1 \leq i \leq s$*, and suppose that, for every* $B \in \mathcal{Q}_i$*, there exists a* $PITD(k, |B| + \gamma_i; \gamma_i^1 1^{|B|})$*. Let* $\sigma' = \sum_{i=1}^{s} \gamma_i$*.*

*Suppose that a* $PITD(k, h + \sigma'; (\sigma')^1 1^h)$ *exists.*

*Then there exists a partitioned ITD,*

$$TD(k, v + \sigma + \sigma') - TD(k, \sigma + \sigma') - \sum_{B \in \mathcal{P}_1} TD(k, |B|).$$

Again some variations are possible that we do not discuss.

One way to construct suitable *IPBD* for Theorems 3.10 and 3.11 is to use the following result:

**Lemma 3.12** *Let* $(V, \mathcal{B})$ *be a resolvable PBD with resolution* $\{\mathcal{P}_1, \ldots, \mathcal{P}_r\}$*. Choose* $B \in \mathcal{P}_1$*. Then* $(V, B, \mathcal{B} \setminus \{B\})$ *is an IPBD whose blocks are partitioned into one holey parallel class* $\mathcal{P}_1 \setminus \{B\}$*, and* $r-1$ *parallel classes* $\mathcal{P}_2, \ldots, \mathcal{P}_s$*.*

One can go further, and consider structures in which there are many holes, and holey parallel classes associated with each. In this direction, one might consider "frames", for example. However, we do not explore this extension.

## 3.3 Making PBD and GDD

Making pairwise balanced designs and group-divisible designs is an industry in itself. Since Wilson's pioneering work on the asymptotic existence of designs (see [97]), constructions of *PBD* and *GDD* have flourished. Indeed one of the main reasons to construct incomplete transversal designs is to use them in constructing various other classes of designs.

We make no effort in this paper to describe all of the available constructions for *PBD* and *GDD*. Instead, we describe here some constructions from *ITD*; in §4 we see a number of other constructions from other classes of designs.

Let us start with easy things. A $TD(k, n)$ is itself a $(nk, \{k, n\}) - PBD$; in fact, it is a $(nk, \{k\}) - GDD$ of type $n^k$.

Deleting any set of elements from a *PBD* produces another *PBD*, in which each deleted element is simply omitted from each block in which it occurred (blocks of size 0, 1, or 2 may result; blocks of size 0 or 1 can be omitted if we choose). Thus every *PBD* gives an enormous variety of smaller *PBD* by this *puncturing* process. However, it should be clear that puncturing a *TD* randomly typically leads to a *PBD* with many block sizes. Since we are interested in being able to apply the theorems given earlier, we are concerned primarily with the cases when puncturing leads to relatively few block sizes.

We describe some concrete instances here. The first is obtained by puncturing points from $\ell$ of the groups.

**Lemma 3.13** *Suppose that a $TD(k+\ell, n)$ exists. Choose integers $b_1, \ldots, b_\ell$ so that $0 \le b_i \le n$ for $1 \le i \le \ell$. Then there exists a $(kn + \sum_{i=1}^{\ell} b_i, \{k, k+1, \ldots, k+\ell\}) - GDD$ of type $n^k b_1^1 b_2^1 \cdots b_\ell^1$.*

Of course, in Lemma 3.13, blocks of sizes $\{k, k+1, \ldots, k+\ell\}$ are all *possible.* But whether a block of a particular size arises depends on the structure of the $TD$ and the actual points deleted. Nevertheless, we can apply Theorem 3.5 to this $GDD$. Knowing the actual block sizes could result in a stronger application of that theorem. We return to this point in §5.7.

Truncating a single group can yield useful $IPBD$:

**Lemma 3.14** *Suppose that a $TD(k+1, n)$ exists. Let $0 \le \rho \le n-1$. Then a $(kn+\rho, \rho, \{k, k+1, n\}) - IPBD$ exists having one holey parallel class of type $n^k$, $n-\rho$ parallel classes of type $k^n$, and the remaining $\rho n$ blocks of size $k$.*

An extreme case of Lemma 3.14 is when the whole group is deleted. This is equivalent to the following well–known result:

**Lemma 3.15** *A resolvable $TD(k, n)$ is equivalent to a $TD(k+1, n)$.*

Puncturing partitioned $ITD$ leads to $GDD$ with groups arising from the holes:

**Lemma 3.16** *Suppose that there exists a $PITD(k+1, n; 1^n)$. Let $0 \le \alpha \le n$. Then there exists a $(nk+\alpha, \alpha, \{n, k+1, k\}) - IPBD$ with a holey parallel class of type $n^k$, a parallel class of type $(k+1)^\alpha k^{n-\alpha}$, and all other blocks of sizes $k$ (whenever $\alpha < n$) and $k+1$ (whenever $\alpha > 0$).*

Puncturing one group of an incomplete $TD$ with one hole leads to:

**Lemma 3.17** *Suppose that there exists an $ITD(k+1, n; h)$. Let $0 \le \alpha < h$. Suppose that there exists a $PITD(k, k+1; 1^{k+1})$ and a $PITD(k, k; 1^k)$. Then there exists a $PITD(k, nk+\alpha, (kh+\alpha)^1 k^{n-h})$.*

Actually, more can be said since $h-\alpha$ holey parallel classes of blocks of size $k$ missing the hole of size $kh+\alpha$ are present, and we have used only a single one here.

We see more sophisticated ways to puncture a $TD$ in §5.7; we give one of the simpler cases here:

**Lemma 3.18** *Suppose that a $TD(k+\ell, n)$ exists with $\ell \geq 2$. Let $1 \leq \alpha \leq n$. Then a $(nk+\alpha+\ell-1, \{k, k+1, k+2, k+\ell\}) - GDD$ of type $n^k\alpha^1 1^{\ell-1}$ exists.*

Using resolvable $TD$, we also obtain:

**Lemma 3.19** *Suppose that a $TD(k+\ell+1, n)$. Then there is a $(nk+\ell, k+\ell, \{k, k+1, n\}) - IPBD$ having a holey parallel class of type $k^{n-1}$, a parallel class of type $n^k 1^\ell$, and $n-1$ parallel classes of type $(k+1)^\ell k^{n-\ell}$.*

Another useful puncture is to delete points from a block, rather than from a group:

**Lemma 3.20** *Suppose that a $TD(k, n)$ exists. Let $\rho$ be an integer satisfying $0 \leq \rho \leq k$. Then there exists a $(k(n-1)+\rho, \rho, \{k, k-1, n, n-1\}) - IPBD$ and a $(k(n-1)+\rho, \{k, k-1, \rho\}) - GDD$ of type $n^\rho(n-1)^{k-\rho}$.*

*In the $IPBD$, blocks of size $n$ appear only if $\rho > 0$ and blocks of size $n-1$ appear only if $\rho < k$. In both the $IPBD$ and the $GDD$, blocks of size $k-1$ appear if and only if $\rho < k$; blocks of size $k$ appear if and only if $\rho > 0$* or *$k \leq n$.*

Deleting a whole block gives, on two occasions, $PBD$ that ought to be noted.

**Lemma 3.21** *If a $TD(n+1, n)$ exists, then there is a $(n^2-1, \{n\})$-GDD of type $(n-1)^{n+1}$. (In fact, they are equivalent.) Further deleting all elements in one group, we obtain a resolvable $(n(n-1), \{n-1\}) - GDD$ of type $n^{n-1}$.*

When a block is deleted from a resolvable $TD$, information about parallel classes can be retained:

**Lemma 3.22** *Suppose that a $TD(k+1, n)$ exists. Let $\rho$ be an integer satisfying $0 \leq \rho \leq k$. Then there exists a $(k(n-1)+\rho, \rho, \{k, k-1, n, n-1\}) - IPBD$ having one holey parallel class of type $k^{n-1}$, one parallel class of type $(n-1)^\rho n^{k-\rho}$, and $n-1$ parallel classes of type $k^{n-k+\rho}(k-1)^{k-\rho}$.*

## 3.4 The Bose–Shrikhande–Parker Theorem

The (general form of the) Bose-Shrikhande-Parker theorem [18, 25] exploits additional structure occurring in some $PBD$. We generalize the notion of parallel class. An *$\alpha$-parallel class* in a $PBD$ $(V, \mathcal{B})$ is a set $\mathcal{C} \subseteq \mathcal{B}$ of blocks, with the property that every $x \in V$ appears in *exactly* $\alpha$ blocks of $\mathcal{C}$. Evidently, a 1–parallel class is just a parallel class.

An $\alpha$-parallel class $\mathcal{C}$ is *symmetric* if every block in $\mathcal{C}$ has size $\alpha$. It is easy to verify in this case that the number of blocks in $\mathcal{C}$ coincides with the number of elements in $V$ — hence the term symmetric.

A *separable PBD* is one whose blocks can be partitioned into 1–parallel classes and symmetric parallel classes; within each class, all blocks have the same size.

Now we can state the Bose–Shrikhande–Parker theorem:

**Theorem 3.23** *Let $(V, \mathcal{B})$ be a $(v, K) - PBD$, and suppose that $\mathcal{B}$ can be partitioned into classes $\mathcal{P}_1, \ldots, \mathcal{P}_r, \mathcal{S}_1, \ldots, \mathcal{S}_s$. For $1 \leq i \leq r$, $\mathcal{P}_i$ is a parallel class. For $1 \leq i \leq s$, $\mathcal{S}_i$ is a symmetric $\alpha_i$-parallel class. Now let $\varepsilon_i \in \{0, 1\}$ for $1 \leq i \leq s$, and suppose that a $PITD(k, \alpha_i; 1^{\alpha_i})$ exists if $\varepsilon_i = 0$, and that a $PITD(k, \alpha_i + 1; 1^{\alpha_i+1})$ exists if $\varepsilon_i = 1$. Let $\sigma = \sum_{i=1}^{s} \varepsilon_i \alpha_i$.*

*Let $\gamma_i$ be a positive integer, for $2 \leq i \leq r$. Suppose that, for $2 \leq i \leq r$ and for each $B \in \mathcal{P}_i$, there exists a $PITD(k, |B| + \gamma_i; \gamma_i^1 1^{|B|})$ exists. Let $\sigma' = \sum_{i=2}^{r} \gamma_i$.*

*Then, if $r \geq 1$, there exists the partitioned $ITD$*

$$TD(k, v + \sigma + \sigma') - TD(k, \sigma + \sigma') - \sum_{B \in \mathcal{P}_1} TD(k, |B|).$$

*If $r = 0$, we instead obtain the partitioned $ITD$*

$$TD(k, v + \sigma) - TD(k, \sigma) - vTD(k, 1).$$

## 3.5 Making Separable PBD

The Bose–Shrikhande–Parker theorem is a generalization of Theorem 3.6 to separable $PBD$. However, it is difficult to find examples of separable $PBD$ that are not resolvable. We see in §§4 that examples arise from cyclic block designs and symmetric block designs. The only other general construction for separable $PBD$ is due to Brouwer [22]:

**Theorem 3.24** *Let $q$ be a prime power, and let $0 < t \leq q^2 - q + 1$. Then there exists a separable $(t(q^2+q+1), \{t, q+t\}) - PBD$ which has a partitioning with $q^2 - q + 1 - t$ parallel classes of blocks of size $t$, and one symmetric $(q+t)$-parallel class.*

# 4 STEINER SYSTEMS, SYMMETRIC DESIGNS AND DIFFERENCE SETS

A *Steiner system* of *order* $v$ and *blocksize* $\kappa$, denoted $S(2, \kappa, v)$, is a $(v, \{\kappa\}) - PBD$. (This is actually a *Steiner 2-design*, but we only have occasion to use the case of $t = 2$ here; it is also a (balanced incomplete) block design, but we only treat the case when $\lambda = 1$. For these reasons, we have adopted the Steiner system notation here.)

An $S(2, \kappa, v)$ is *symmetric* when the number of blocks in the design, namely $\frac{v(v-1)}{\kappa(\kappa-1)}$ is equal to $v$ (i.e., $v = \kappa(\kappa-1)+1$). A symmetric $S(2, \kappa, v)$ is equivalent to a *projective plane* of *order* $n = \kappa - 1$, with $v = n^2 + n + 1$ elements (and $n^2 + n + 1$ blocks or lines). In a projective plane, every two distinct blocks intersect in one element.

An $S(2, \kappa, v)$ is *cyclic* when there is an automorphism of the design that is a $v$–cycle.

First, the basics:

**Lemma 4.1** *Removing one element from a projective plane of order $n$, and treating the resulting blocks of size $n$ as groups, a $TD(n+1, n)$ is produced.*

We can also remove a whole block:

**Lemma 4.2** *Removing one block from a projective plane of order $n$ (or one group from a $TD(n+1, n)$, a resolvable $TD(n, n)$ (*affine plane of order *$n$, or $S(2, n, n^2)$) results.*

*Moreover, every $TD(n,n)$ is resolvable and can be extended to a projective plane.*

## 4.1 Arbitrary $S(2,\kappa,v)$

An $S(2,\kappa,v)$ is, of course, itself a $PBD$. However, sometimes truncating this special type of $PBD$ can lead to extra information. We can truncate points from a single block:

**Lemma 4.3** *If an $S(2,\kappa,v)$ exists, then for $0 \le x \le \kappa$, a $(v-x,\kappa-x,\{\kappa,\kappa-1\})-IPBD$ exists having $x$ holey parallel classes of type $(\kappa-1)^{(v-\kappa)/(\kappa-1)}$. It has blocks of size $\kappa$ unless $x=\kappa$ and the $S(2,\kappa,v)$ is symmetric.*

Next we delete a small number of points, not all from the same block.

**Lemma 4.4** *If an $S(2,\kappa,v)$ exists, then there exists a $(v-3,\{\kappa-2,\kappa-1,\kappa\})-PBD$ in which there are exactly three blocks of size $\kappa-2$, and they form a near clear set.*

*If an $S(2,\kappa,v)$ exists, then there exists a $(v-4,\{\kappa-2,\kappa-1,\kappa\})-PBD$ in which there are exactly four blocks of size $\kappa-2$, and they form a near clear set.*

We can naturally delete points all over if we so desire, but to obtain useful results we want to minimize the number of different block sizes. With this in mind, we give another definition. If $A$ is a set of $s$ points, no three of which lie on a block, then $A$ is an *$s$–arc*.

**Lemma 4.5** *If an $S(2,\kappa,v)$ exists having an $s$-arc, then for all $1 \le x \le s$, there exists a $(v-x,\{\kappa-2,\kappa-1,\kappa\})-PBD$. Blocks of size $\kappa-2$ always occur when $x>1$. Blocks of size $\kappa-1$ appear unless $v=1+(x-1)(\kappa-1)$. Blocks of size $\kappa$ always appear.*

Examples of designs with useful arcs appear in §4.3.

Existence of block designs is a central problem in combinatorial design theory, and there is a huge literature. For existence results, see [117]. Much is known about resolvability of block designs, furnishing many examples of resolvable $PBD$.

## 4.2 Cyclic $S(2,\kappa,v)$

Cyclic $S(2,\kappa,v)$ have been studied extensively; see [113]. A cyclic $S(2,\kappa,v)$ can exist only if $v \equiv 1, \kappa \pmod{\kappa(\kappa-1)}$. When $v \equiv 1 \pmod{\kappa(\kappa-1)}$, all block orbits under the cyclic automorphism have length $v$ (they are *full*). When $v \equiv \kappa \pmod{\kappa(\kappa-1)}$, one block orbit has length $\frac{v}{\kappa}$ (it is *short*), and the rest are full.

A full orbit of blocks can be easily checked to be a $\kappa$-parallel class. Hence every cyclic $S(2,\kappa,v)$ is separable, with $\lfloor\frac{v}{\kappa(\kappa-1)}\rfloor$ $\kappa$-parallel classes, and one parallel class if the short orbit is present, none otherwise.

Hence we can apply the Bose–Shrikhande–Parker theorem to cyclic $S(2,\kappa,v)$.

## 4.3 Symmetric Designs

First we remark on a basic filling result that does not follow from filling the corresponding $PBD$ [56].

**Lemma 4.6** *If a symmetric $S(2,\kappa,v)$ exists and a $TD(k,\kappa)$ exists, then a $TD(k,v)$ exists.*

Next observing that a symmetric design $S(2,\kappa,v)$ is itself a single $((v-1)/(\kappa-1))$-parallel class, we can apply the Bose–Shrikhande–Parker theorem to obtain:

**Lemma 4.7** *If a symmetric $S(2,\kappa,v)$ exists and a $PITD(k,\kappa{+}1;1^{\kappa+1})$ exists, then a $TD(k,v+\frac{v-1}{\kappa-1})-TD(k,\frac{v-1}{\kappa-1})$ exists.*

See also [56].

Certain projective planes have large arcs:

**Theorem 4.8** *The desarguesian projective plane of order $q$ (a prime power) contains a $(q+1)$–arc (an* oval*) if $q$ is odd, and contains a $(q+2)$–arc (a* hyperoval*) if $q$ is even.*

## 4.4 Line–flips in Affine Planes

Suppose that a $TD(n, n)$ exists; this is an affine plane of order $n$. Now choose an integer $x$ with $1 \leq x < n$, and choose one block $B$. Delete all points from $x$ of the groups *except* those on block $B$. Next delete all points on block $B$ in the remaining $n - x$ groups. The resulting $PBD$ has blocks of five types:

1. a single block on $x$ points, which is the truncation of $B$;
2. $n - 1$ disjoint blocks each of size $n - x$, which are the truncations of the blocks disjoint from $B$ in the affine plane;
3. $n - x$ disjoint blocks each of size $n - 1$, which are the truncations of the remaining groups;
4. blocks of size $n - x - 1$ that do not intersect the truncation of $B$ (in the affine plane, they did intersect $B$);
5. blocks of size $n - x + 1$ that do intersect the truncation of $B$.

Types (1) and (2) form a parallel class; so also do types (1) and (3). Thus adding a point at infinity to the blocks of type (1) and (2), called a *type A extension*, gives a $\{n - x - 1, n - x + 1\}$-GDD of type $(n - 1)^{n-x}(x + 1)^1$. On the other hand, adding a point at infinity to blocks of types (1) and (3) gives a $\{n - x - 1, n, n - x + 1\}$-GDD of type $(n - x)^{n-1}(x + 1)^1$ (a *type B extension*). Greig [42] observes that either GDD can be extended with a further point at infinity to form a $PBD$ on $(n - x)(n - 1) + x + 2$ with block sizes $\{n, n - x - 1, n - x + 1, x + 2^\star\}$. The superscript $\star$ indicates that a block of size $x+2$ is present, and that all other blocks have sizes from $\{n, n-x-1, n-x+1\}$.

## 4.5 Difference Sets

Singer [79] showed that the desarguesian projective plane of order $q$ (a prime power) has a representation as a cyclic difference set. This provides a mechanism for finding other configurations in desarguesian planes.

Let $\Gamma$ be an additively written group of order $v$. A $\kappa$-subset $D$ of $\Gamma$ is a $(v, \kappa, \lambda)$-*difference set* of *order* $n = \kappa - \lambda$ if every nonzero element of $\Gamma$ has exactly $\lambda$ representations as a difference $d - d'$ of distinct elements from $D$. The difference set is *abelian* or *cyclic* if the group $\Gamma$ has the corresponding property.

The development of a difference set $D$ under the action of the group $\Gamma$ is a symmetric design; when $\lambda = 1$, it is a projective plane of order $n$. Thus our earlier remarks apply to the symmetric design. But here we may obtain more information.

We consider a $(q^2+q+1, q+1, 1)$–difference set $D$ over the cyclic group $Z_{q^2+q+1}$, using the usual representation over the integers modulo $q^2+q+1$. For any divisor $d$ of $q^2+q+1$, denote by $D_{i,d}$ the elements of $D$ that are congruent to $i$ modulo $d$. For an arbitrary subset $\mathcal{I} \subset \{0, 1, \ldots, d-1\}$, let $D_{\mathcal{I},d} = \bigcup_{i \in \mathcal{I}} D_{i,d}$. Then we have the following result, first studied by Brouwer [22] and later extended by Greig [42]:

**Theorem 4.9** *Let $D$ be a $(q^2+q+1, q+1, 1)$–difference set over the integers modulo $q^2+q+1$. Let $d$ be a divisor of $q^2+q+1$, and $\mathcal{I} \subseteq \{0, 1, \ldots, d-1\}$. Then the collection of blocks*

$$\{\{x+i\} : 0 \le i \le q^2+q,\ x+i \bmod d \in \mathcal{I}\}$$

*is a pairwise balanced design on $|\mathcal{I}|\frac{q^2+q+1}{d}$ elements.*

The relevance of Theorem 4.9 is that it produces a $PBD$ having at most $d$ different block sizes.

## 4.6 Configurations in Projective Planes

In §4.5, we saw that projective planes arising from difference sets can embed a pairwise balanced design that often has "few" block sizes. We are interested in this phenomenon for a number of reasons. It provides a way to construct pairwise balanced designs, of course. But what is more critical for us is that it tells us something about the structure of the $TD(n+1, n)$ that arises from the plane — and this information can be helpful in predicting the block sizes that result when we puncture the $TD$. There is a third reason as well, namely that when $PBD$ live in a projective plane, we can use this to produce more $PBD$. We pursue this in §4.7, but for now we explore results on when $PBD$ live in projective planes.

### *Ovals, Hyperovals and Denniston Arcs*

Arcs (ovals and hyperovals) form one important class of pairwise balanced designs inhabiting projective planes (Theorem 4.8), although the $PBD$ themselves are quite trivial. However, associated with the exterior lines of a hyperoval are a number of important $PBD$ contained in the plane [77]:

**Theorem 4.10** *The desarguesian projective plane of order $q = 2^\alpha$ contains*

1. *a resolvable $(\binom{q}{2}, \{\frac{q}{2}\}) - PBD$;*
2. *a $(\binom{q+2}{2}, \{\frac{q}{2}+1, q+1\}) - PBD$; and*
3. *a resolvable $(\binom{q+1}{2}, \{\frac{q}{2}, q\}) - PBD$.*

Denniston arcs [34] provide a generalization of these:

**Theorem 4.11** *The desarguesian projective plane of order $q = 2^\alpha$ contains, for every $1 \leq \beta < \alpha$,*

1. *a resolvable $(2^{\alpha+\beta} - 2^\alpha + 2^\beta, \{2^\beta\}) - PBD$;*
2. *a $(2^{2\alpha} + 2^{\alpha+1} - 2^{\alpha+\beta} - 2^\beta + 1, \{2^\alpha - 2^\beta + 1, 2^\alpha + 1\}) - PBD$; and*
3. *a resolvable $((2^\alpha - 2^\beta)(2^\alpha + 1), \{2^\alpha - 2^\beta, 2^\alpha\}) - PBD$.*

Greig [42] employs ovals in planes of odd order to prove:

**Theorem 4.12** *If $q$ is an odd prime power, the desarguesian plane of order $q$ contains*

1. *a GDD on $\binom{q}{2}$ points with uniform group size $\frac{q-1}{2}$, and block sizes in $\{\frac{q-1}{2}, \frac{q+1}{2}\}$; and*
2. *a GDD on $\binom{q+1}{2}$ points with uniform group size $\frac{q+1}{2}$, and block sizes in $\{\frac{q+1}{2}, \frac{q+3}{2}\}$.*

## *Subplanes and Baer Subplanes*

More complex examples are given by subplanes of a plane. Simple numerical arguments show that a projective plane of order $q$ can have a projective subplane of order $p$ only if $q \geq p^2$. In the positive direction, we have [13]:

**Lemma 4.13** *The desarguesian projective plane of order $p^\alpha$ has a subplane of order $p^\beta$ whenever $\beta \mid \alpha$.*

The extreme case when $\alpha = 2\beta$ is especially important. In this case, the subplane is a *Baer subplane*, and some elementary counting arguments provide us with useful information. Let $q = p^\beta$ and $q^2 = p^{2\beta}$. Let $(V, \mathcal{B})$ be the plane of order $q^2$, and $(X, \mathcal{D})$ be its Baer subplane of order $q$.

**Lemma 4.14** *1. Every point $x \in X$ lies on $q+1$ lines of $\mathcal{B}$ that intersect $X$ in $q+1$ points, and on $q^2-q$ lines of $\mathcal{B}$ that contain only $x$ from $X$.*

*2. Every point of $V \setminus X$ lies on one line of $\mathcal{B}$ that intersects $X$ in $q+1$ points, and lies on $q^2$ lines of $\mathcal{B}$ that intersect $X$ in one point.*

*3. Hence, all lines of $\mathcal{B}$ intersect $X$ in either 1 or $q+1$ points.*

Removing the points in $X$ from the plane yields a $(q^4-q, \{q^2-q, q^2\})-PBD$. Considering any point $x \in X$, we find that the blocks containing $X$ form a parallel class of this $PBD$, and hence we in fact obtain a $(q^4-q, \{q^2-q, q^2\})-GDD$ of type $(q^2-q)^{q+1}(q^2)^{q^2-q}$, $(Y, \mathcal{C})$.

Now consider a block of size $q^2-q$ in $\mathcal{C}$. It cannot intersect any group of size $q^2-q$ (lines meet at a single point in the projective plane of order $q^2$), so it must intersect all groups of size $q^2$. In fact, all blocks and groups of size $q^2-q$ are disjoint, so we have [80]:

**Lemma 4.15** *If a projective plane of order $q^2$ has a Baer subplane of order $q$, there exists a $(q^4-q, \{q^2\})-GDD$ of type $(q^2-q)^{q^2+q+1}$.*

Now a block of size $q^2$ from $\mathcal{C}$ must intersect all groups of size $q^2$, and precisely $q$ of the groups of size $q^2-q$. Thus we can delete all but $x$ of the groups of size $q^2$ to obtain:

**Lemma 4.16** *If a projective plane of order $q^2$ has a Baer subplane of order $q$, then for all $0 \leq x \leq q^2 - q$, there exist*

1. *a $((q^2-q)(q+1)+xq^2, \{q+x, x\}) - GDD$ of type $(q^2-q)^{q+1}(q^2)^x$; and*
2. *a $((q^2-q)(q+1)+xq^2, \{q+x, q^2\}) - GDD$ of type $(q^2-q)^{q+1}x^{(q^2)}$.*

Baer subplanes can be exploited further yet; see, for example, [51] for the following:

**Lemma 4.17** *The desarguesian projective plane of order $q^2$ can be partitioned into $q^2-q+1$ element-disjoint Baer subplanes (each on $q^2+q+1$ points).*

Considering any line of the plane, simple counting shows that it intersects one Baer subplane of this partition in $q+1$ points, and the remaining $q^2-q$ subplanes in one point each. So retaining points of $t$ of the subplanes in the partition, we obtain:

**Lemma 4.18** *¿From the desarguesian projective plane of order $q^2$ ($q$ a prime power), for each $1 \leq t \leq q^2-q$, we obtain a $(t(q^q+q+1), \{q+t, t\}) - PBD$ in which the blocks of size $t$ are partitioned into $q^2-q+1-t$ parallel classes.*

Other specific planes have subplanes of interest: the Hughes plane of order 9 has a subplane of order 2 [33], and the Hughes plane of order 25 contains subplanes of orders 2 and 3 [70]. A complete survey of subplanes is *not* attempted here.

## *Affine Subplanes*

Now we examine other structures in planes. In the direction of affine planes residing in projective planes, Ostrom and Sherk [66] and Rigby [71] proved:

**Theorem 4.19** *The desarguesian projective plane of order $q$ (a prime power) contains an affine plane of order 3 (an $S(2,3,9)$) if and only if $q \equiv 0, 1 \pmod 3$.*

The notion of "containment" in Theorem 4.19 is that a subset of the points is selected, and the intersections of all lines with these points induce shorter lines; then keeping all such truncated lines on two or more points gives the affine plane.

### *Subsquares*

Often a subplane (projective or affine) is not present, but useful portions are. For example, considering the standard construction of the desarguesian plane, we find [30]:

**Lemma 4.20** *In the desarguesian plane of order $p^\alpha$, for each $0 \le \beta \le \alpha$, there is embedded a $(3p^\beta + 1, \{3, p^\beta + 1\}) - PBD$ having three blocks of size $p^\beta + 1$ meeting in a single point, and all other blocks of size three.*

Actually, a more convenient way to express this is to observe that, when we remove the point common to the three "long" blocks, we form a $TD(p^\alpha + 1, p^\alpha)$. Truncating to the $TD(3, p^\alpha)$ on the three special groups, and interpreting this $TD$ as a latin square, we are essentially noting in Lemma 4.20 that this latin square of size $p^\alpha$ has a subsquare of size $p^\beta$.

Using the structure of the finite field, one can extend this to obtain:

**Lemma 4.21** *In the desarguesian plane of order $p^\alpha$, for each $0 \le \beta \le \alpha$, there is embedded a $((p+1)p^\beta + 1, \{p+1, p^\beta + 1\}) - PBD$ having $p+1$ blocks of size $p^\beta + 1$ meeting in a single point, and all other blocks of size $p+1$.*

When $\beta = \alpha - 1$ in Lemma 4.21, all lines of the plane meet the sub-$TD$ in $p^\beta + 1$, $p+1$, or 1 points. The latter tangent lines induce a design in the dual plane; this tangent design is a $\{p^\alpha, p^\alpha - p^{2\beta-\alpha} + 1\}$-GDD of type $(p^\alpha - p^\beta)^{p^\alpha + p^\beta}(p^\alpha - p^{\alpha-\beta})^1$. Further extensions are studied in [29].

## 4.7 Line-flips in Projective Planes

When a $PBD$ is embedded in a projective plane, we can exploit the structure of the enclosing plane to form other $PBD$. Simply taking all points of the plane *not* in the $PBD$, for example, gives:

**Lemma 4.22** *If a $(v, K) - PBD$ is embedded in a projective plane of order $n$, and there is a one–to–one correspondence between blocks of the $PBD$ and lines of the plane so that each block is extended to the corresponding line (which* *may require adding blocks of sizes 0 and 1 to the PBD), then there exists a* $(n^2+n+1-v, \overline{K}) - PBD$*, where* $\overline{K} = \{n+1-s : s \in K\}$*.*

*In fact, the number of blocks of size* $n+1-s$ *in the resulting PBD is the same as the number of blocks of size s in the original PBD.*

One can also do a "line–flip", by choosing some block of the $PBD$, deleting the points on this block and instead adding the points on the line of the plane which extends this block, but not on the block itself [42]. One obtains the following:

**Lemma 4.23** *Suppose that a* $(v, K) - PBD$ *is embedded in a projective plane of order n, and there is a one–to–one correspondence between blocks of the PBD and lines of the plane so that each block is extended to the corresponding line (which may require adding blocks of sizes 0 and 1 to the PBD). Suppose further that the embedded PBD has a block of size s. Let* $\widehat{K} = \{s-1, s+1 : s \in K\}$*. Then there exists a* $(v+n+1-2s, n+1-s, \widehat{K}) - IPBD$*.*

Examples of $PBD$ that inhabit projective planes are given in §4.6, and also arise from Theorem 4.9.

# 5 WILSON'S THEOREM

Wilson's theorem, and all of its variants discussed here, start with a transversal design (or incomplete transversal design) of order $t$ of blocksize $k+\ell$. We refer to this design as the *master design.*

The master design is always taken to be $(V, \mathcal{G}, \mathcal{B})$, although it may have additional structure, or holes. Let $\mathcal{G} = \{G_1, \ldots, G_k, E_1, \ldots, E_\ell\}$. Let $E_i = \{x_{i1}, \ldots, x_{it}\}$. For each $x_{ij} \in \bigcup_{i=1}^{\ell} E_i$, let $w_{ij}$ be a nonnegative integer, the *weight* of $x_{ij}$. For each block $B \in \mathcal{B}$, let $w_i^B = w_{ij}$ when $B \cap E_i = \{x_{ij}\}$.

## 5.1 Transversal Designs as Master Designs

First we give what has come to be accepted as the basic form of Wilson's theorem, although Wilson [96] gave it in the case that $w_{ij} \in \{0, 1\}$ for all $1 \le i \le \ell$, $1 \le j \le t$, and $\lambda = \mu = 1$.

**Theorem 5.1** *Suppose that a $TD_\mu(k+\ell,t)$ exists. Suppose that for each $B \in \mathcal{B}$, there exists*

$$TD_\lambda(k, m + \sum_{i=1}^{\ell} w_i^B) - \sum_{i=1}^{\ell} TD_\lambda(k, w_i^B).$$

*Then there exists a*

$$TD_{\lambda\mu}(k, mt + \sum_{i=1}^{\ell}\sum_{j=1}^{t} w_{ij}) - \sum_{i=1}^{\ell} TD_{\lambda\mu}(k, \sum_{j=1}^{t} w_{ij})$$

Of course, if we can fill some or all of the holes, further incomplete transversal designs will result.

When $\mu = 1$, we can obtain different holes as well:

**Theorem 5.2** *Suppose that a $TD_1(k+\ell,t)$ exists. Let $F \in \mathcal{B}$. Suppose that for each $B \in \mathcal{B} \setminus \{F\}$, there exists*

$$TD_\lambda(k, m + \sum_{i=1}^{\ell} w_i^B) - \sum_{i=1}^{\ell} TD_\lambda(k, w_i^B).$$

*Suppose that, for $1 \le i \le \ell$, there exists*

$$TD_\lambda(k, \sum_{j=1}^{t} w_{ij}) - TD_\lambda(k, w_i^F).$$

*Then there exists*

$$TD_\lambda(k, mt + \sum_{i=1}^{\ell}\sum_{j=1}^{t} w_{ij}) - TD_\lambda(k, m + \sum_{i=1}^{\ell} w_i^F).$$

We now change the structure of some of the ingredients. Let $\mathcal{B}_1$ be the blocks $B$ for which $x_{11} \in B \in \mathcal{B}$, and let $\mathcal{B}_2 = \mathcal{B} \setminus \mathcal{B}_1$.

**Theorem 5.3** *Suppose that a $TD_1(k+\ell,t)$ exists. Suppose that for each $B \in \mathcal{B}_2$, there exists*

$$TD_\lambda(k, m + \sum_{i=1}^{\ell} w_i^B) - \sum_{i=1}^{\ell} TD_\lambda(k, w_i^B).$$

*Let $\alpha_1, \alpha_2, \ldots, \alpha_r$ be positive integers with $\sum_{i=1}^{r} \alpha_i \leq m$. Suppose that for each $B \in \mathcal{B}_1$, there exists*

$$TD_\lambda(k, m + \sum_{i=1}^{\ell} w_i^B) - \sum_{i=1}^{r} TD_\lambda(k, \alpha_i) - \sum_{i=1}^{\ell} TD_\lambda(k, w_i^B).$$

*Then there exists*

$$TD_\lambda(k, mt + \sum_{i=1}^{\ell}\sum_{j=1}^{t} w_{ij}) - \sum_{i=1}^{r} tTD_\lambda(m, \alpha_i) - \sum_{i=1}^{\ell} TD_\lambda(k, \sum_{j=1}^{t} w_{ij}).$$

Finally we change the structure of all of the ingredients:

**Theorem 5.4** *Suppose that a $TD_\mu(k+\ell, t)$ exists. Let $\alpha_1, \alpha_2, \ldots, \alpha_r$ be positive integers with $\sum_{i=1}^{r} \alpha_i \leq m$. Suppose that for each $B \in \mathcal{B}$, there exists*

$$TD_\lambda(k, m + \sum_{i=1}^{\ell} w_i^B) - \sum_{i=1}^{r} TD_\lambda(k, \alpha_i) - \sum_{i=1}^{\ell} TD_\lambda(k, w_i^B).$$

*Then there exists*

$$TD_{\lambda\mu}(k, mt + \sum_{i=1}^{\ell}\sum_{j=1}^{t} w_{ij}) - \sum_{i=1}^{r} TD_{\lambda\mu}(m, \alpha_i t) - \sum_{i=1}^{\ell} TD_{\lambda\mu}(k, \sum_{j=1}^{t} w_{ij}).$$

## 5.2 Incomplete Transversal Designs as Master Designs

In the preceding constructions, we saw how incomplete transversal designs can be used in conjunction with a master design that is a transversal design. Here we examine variants where the master design itself is incomplete.

**Theorem 5.5** *Let $\beta_1, \ldots, \beta_u$ be positive integers with $\sum_{a=1}^{u} \beta_a \leq t$. Suppose that there exists a master design, a*

$$TD_\mu(k+\ell, t) - \sum_{a=1}^{u} TD_\mu(k+\ell, \beta_a).$$

*For* $1 \leq a \leq u$, *let* $O_a$ *be the points in the ITD that lie in the* $a^{th}$ *hole of size* $\beta_a$ *(so that* $O_a$ *contains* $\beta_a(k+\ell)$ *elements in total). Let* $z_{ia} = \sum_{x_{ij} \in (E_i \cap O_a)} w_{ij}$. *Suppose that for each* $B \in \mathcal{B}$, *there exists*

$$TD_\lambda(k, m + \sum_{i=1}^{\ell} w_i^B) - \sum_{i=1}^{\ell} TD_\lambda(k, w_i^B).$$

*Suppose further that, for* $1 \leq a \leq u$, *there exists*

$$TD_{\lambda\mu}(k, m\beta_a + \sum_{i=1}^{\ell} z_{ia}) - \sum_{i=1}^{\ell} TD_{\lambda\mu}(k, z_{ia}).$$

*Then there exists a*

$$TD_{\lambda\mu}(k, mt + \sum_{i=1}^{\ell}\sum_{j=1}^{t} w_{ij}) - \sum_{i=1}^{\ell} TD_{\lambda\mu}(k, \sum_{j=1}^{t} w_{ij})$$

Actually, the holes that arise on the extra $\ell$ levels, and the holes that arise from the holes in the master design in the construction are not disjoint. In Theorem 5.5, we have elected to fill the latter and leave the former. In Theorem 5.6, we do the opposite. Since the theorems differ only in the last set of ingredients, they look cosmetically similar. Nevertheless, we state the conditions of the theorem in their entirety.

**Theorem 5.6** *Let* $\beta_1, \ldots, \beta_u$ *be positive integers with* $\sum_{a=1}^{u} \beta_a \leq t$. *Suppose that there exists a master design, a*

$$TD_\mu(k+\ell, t) - \sum_{a=1}^{u} TD_\mu(k+\ell, \beta_a).$$

*For* $1 \leq a \leq u$, *let* $O_a$ *be the points in the ITD that lie in the* $a^{th}$ *hole of size* $\beta_a$ *(so that* $O_a$ *contains* $\beta_a(k+\ell)$ *elements in total). Let* $z_{ia} = \sum_{x_{ij} \in (E_i \cap O_a)} w_{ij}$. *Suppose that for each* $B \in \mathcal{B}$, *there exists*

$$TD_\lambda(k, m + \sum_{i=1}^{\ell} w_i^B) - \sum_{i=1}^{\ell} TD_\lambda(k, w_i^B).$$

*Suppose further that, for* $1 \leq i \leq \ell$, *there exists*

$$TD_{\lambda\mu}(k, \sum_{j=1}^{t} w_{ij}) - \sum_{a=1}^{u} TD_{\lambda\mu}(k, z_{ia}).$$

*Then there exists a*

$$TD_{\lambda\mu}(k, mt + \sum_{i=1}^{\ell}\sum_{j=1}^{t} w_{ij}) - \sum_{a=1}^{u} TD_{\lambda\mu}(k, m\beta_a + \sum_{i=1}^{\ell} z_{ia}).$$

## 5.3 Du Variations

Du [38] considers another use of incomplete transversal designs in Wilson's theorem. Denote by $ITD^s(k,t;h)$ an $ITD(k,t;h)$ that has $s$ disjoint holey parallel classes of blocks. An $ITD^1(k,t;h)$ is equivalent to a $PITD(k,t;h^1 1^{t-h})$. Then we have:

**Theorem 5.7** *Suppose that a $ITD^s(k,t;h)$ exists. Suppose that a $TD(k,m)$ exists. Let $w_1,\ldots,w_s$ be nonnegative integers, and suppose that an $ITD(k,m+w_i;w_i)$ exists for each $i=1,\ldots,s$. Then a $TD(k,mt+\sum_{i=1}^{s} w_i)-TD(k,mh+\sum_{i=1}^{s} w_i)$ exists.*

*If $w_i = 0$ for some $i$, $1 \le i \le s$, then the stronger result is obtained that a*

$$TD(k, mt + \sum_{i=1}^{s} w_i) - TD(k, mh + \sum_{i=1}^{s} w_i) - (t-h)TD(k,m)$$

*exists.*

## 5.4 Another Variant

Colbourn [28] establishes the following:

**Theorem 5.8** *If there exists a $ITD(k, n+h; h)$ for which $(k-2)h = n$, and there exists a $TD(k,m)$, then there exists an $ITD(k, mn+(m-1)h; n^m (h(m-1))^1)$.*

## 5.5 Wojtas Structures

For ease of exposition, we assume henceforth that $\lambda = \mu = 1$; the extensions to higher index are, for the most part, routine.

An examination of the propositions used in Brouwer [20] reveals that many, due to Wojtas, arise by inflating objects before filling them. We develop a framework here for presenting such constructions generally.

A *partial transversal design* of *order* or *groupsize* $n$, *blocksize* $k$, denoted here by $PTD(k,n)$, is a triple $(V,\mathcal{G},\mathcal{B})$, where

1. $V$ is a set of $kn$ elements;
2. $\mathcal{G}$ is a partition of $V$ into $k$ classes (called *groups*), each of size $n$;
3. $\mathcal{B}$ is a collection of $k$–subsets of $V$ (called *blocks*);
4. every unordered pair of elements from $V$ is either contained in exactly one group, or is contained in *at most one* block. Elements appearing together in a group do not appear together in a block.

A *hole* $H$ of order $h$ in a $PTD(k,n)$ $(V,\mathcal{G},\mathcal{B})$ is a set $H \subseteq V$ with $|H \cap G| = h$ for each $G \in \mathcal{G}$, and $H \cap B = \emptyset$ for each $B \in \mathcal{B}$.

Two holes $H_1$ and $H_2$ are *compatible* if $H_1 \cap H_2 \subset G$ for some $G \in \mathcal{G}$. Compatibility is a *weaker* condition than disjointness. A *Wojtas structure* of *order* $n$, *blocksize* $k$, and *holesizes* $\mathcal{M}$, denoted $WS(k,n,\mathcal{M})$, is a $PTD(k,n)$ $(V,\mathcal{G},\mathcal{B})$ together with a set $\mathcal{H}$ of holes, so that

$$\mathcal{M} = \{\frac{|H|}{k} : H \in \mathcal{H}\}.$$

In addition, every pair of distinct holes from $\mathcal{H}$ are compatible. Moreover, every pair $\{x,y\} \subset V$ that does not appear in a group, *either* appears in exactly one block of $\mathcal{B}$, *or* appears in exactly one hole of $\mathcal{H}$, *but not both.*

If, in the definition, we replaced the single word "compatible" by the stronger word "disjoint", we would repeat the definition of incomplete transversal design. We make some simple (but important) observations.

**Lemma 5.9** *A* $WS(k,n,\{1\})$ *is equivalent to a* $TD(k,n)$.

**Lemma 5.10** *If a* $WS(k,n,\mathcal{M})$ *exists, and a* $TD(k,m)$ *exists for each* $m \in \mathcal{M}$*, then a* $TD(k,n)$ *exists. Indeed, a* $TD(k,n) - TD(k,m)$ *exists for every* $m \in \mathcal{M}$.

**Lemma 5.11** *If* $WS(k,n,\mathcal{M})$ *and* $TD(k,w)$ *both exist, a* $WS(k,wn,\{wm : m \in \mathcal{M}\})$ *exists.*

Of course, when the holes are all disjoint, we have incomplete transversal designs. Sometimes we can employ other useful patterns of holes as well. Many variations are possible, but we just develop one generalization here.

A Wojtas structure $WS(n,k,\mathcal{M})$ $(V,\mathcal{G},\mathcal{H},\mathcal{B})$ is *partitionable of type*

$$(e_1,\ldots,e_\ell;\mathcal{N},c),$$

denoted $PWS(k,n;e_1,\ldots,e_\ell;\mathcal{N},c)$, if $\mathcal{M}=\mathcal{N}\cup\{e_1,\ldots,e_\ell\}$, and the holes are

$$\mathcal{H}=\{H_{\infty_1},H_{\infty_2},\ldots,H_{\infty_\ell}\}\cup\{H_{ij}:1\le i\le c, 1\le j\le d\},$$

so that, for each $1\le i\le c$, $H_{\infty_1},H_{\infty_2},\ldots,H_{\infty_\ell},H_{i1},H_{i2},\ldots,H_{id}$ are all *disjoint*, $\mathcal{N}$ contains all hole sizes among the $\{H_{ij}\}$, and, for all $1\le i\le c$, $\bigcup_{j=1}^{\ell} H_{\infty_j}\cup\bigcup_{j=1}^{d} H_{ij}=V$. Moreover, $H_{\infty i}$ is a hole of size $e_i$.

One might think about the $\{H_{ij}:1\le j\le d\}$, $1\le i\le c$ as being "partial parallel classes of holes"; each, together with the special holes $\{H_{\infty_j}\}$, forms a "parallel class of holes".

In fact, we have the following equivalence:

**Lemma 5.12** *A* $PITD(k,n;b_1,\ldots,b_s)$ *is equivalent to a*

$$PWS(k,n;b_1,\ldots,b_s;\emptyset,c)$$

*for all* $c$. *For every* $0\le\ell\le s$, *it implies the existence of a*

$$PWS(k,n;b_1,\ldots,b_\ell;\{b_{\ell+1},\ldots,b_s\},1).$$

**Lemma 5.13** *A resolvable* $TD(k,n)$ *is equivalent to a* $PWS(k,n;\emptyset;\{1\},n)$.

Inflation, as in Lemma 5.11 works again, but we obtain a stronger result:

**Lemma 5.14** *If a* $PWS(k,n;e_1,\ldots,e_\ell;\mathcal{M}^\star,c)$ *exists and a* $TD(k,w)$ *exists, so also does a* $PWS(k,wn;we_1,\ldots,we_\ell;\{wm:m\in\mathcal{M}^\star\},c)$.

Naturally, since a $PWS$ is a $WS$, one can apply Lemma 5.10 to fill the holes. However, the structure of the partitioning can be exploited to obtain a more sophisticated result:

**Lemma 5.15** *Suppose that there exists a* $PWS(k,n;e_1,\ldots,e_\ell;\mathcal{M}^\star,c)$. *Let* $\gamma_1,\ldots,\gamma_c$ *be nonnegative integers, and write* $\sigma=\sum_{i=1}^{c}\gamma_i$. *Suppose that, for every* $m\in\mathcal{M}^\star$, *and every* $1\le i\le\ell$, *there exists*

$$TD(k,m+\gamma_i)-TD(k,\gamma_i).$$

*Suppose further that, for* $2\le i\le\ell$, *there exists*

$$TD(k,e_i+\sigma)-TD(k,\sigma).$$

*Then there exists*

$$TD(k,n+\sigma)-TD(k,e_1+\sigma).$$

Again, we have the phenomenon that in the middle of the construction, we have $\ell$ holes of sizes $\sigma+e_i$ for $1\le i\le\ell$, but they all intersect in a hole of size $\sigma$. Lemma 5.15 gives one way to fill all but one of the holes. Here is another:

**Lemma 5.16** *Suppose that there exists a* $PWS(k,n;e_1,\ldots,e_\ell;\mathcal{M}^\star,c)$. *Let* $\gamma_1,\ldots,\gamma_c$ *be nonnegative integers, and write* $\sigma=\sum_{i=1}^{c}\gamma_i$. *Suppose that, for every* $m\in\mathcal{M}^\star$, *and every* $1\le i\le\ell$, *there exists*

$$TD(k,m+\gamma_i)-TD(k,\gamma_i).$$

*Suppose further that, for* $1\le i\le\ell$, *there exists*

$$TD(k,e_i+\sigma)-TD(k,e_i).$$

*Then there exists*

$$TD(k,n+\sigma)-\sum_{i=1}^{\ell}TD(k,e_i).$$

When $\ell=1$ and $\mathcal{M}=\{m\}$, we can also avoid filling one of the parallel classes of holes, to obtain:

**Lemma 5.17** *Suppose that a $PWS(k, n; e; \{m\}, c)$ exists. Let $\gamma_1, \ldots, \gamma_c$ be nonnegative integers, and write $\sigma = \sum_{i=1}^{c} \gamma_i$. Suppose that there exists, for every $1 \leq i \leq c$, a*

$$TD(k, m + \gamma_i) - TD(k, \gamma_i).$$

*Then there exists*

$$TD(k, n + \sigma) - TD(k, e + \sigma) - \sum_{i=1}^{(n-e)/m} TD(k, m).$$

The power of Wojtas structures in general is that, rather than filling them immediately, one can inflate them and then fill them. This can often yield better results than are obtained by filling them and then inflating.

The additional power of partitioned Wojtas structures is the more sophisticated manner in which they can be filled.

It appears that Wojtas structures and partitioned Wojtas structures can lead to new incomplete transversal designs, but of course we have seen no ways to construct them except via equivalences to transversal designs and incomplete transversal designs.

## 5.6 Making Wojtas Structures

By now, it should come as no surprise that one way to construct Wojtas structures is to use Wilson's theorem in its many disguises. But an easier way to get some Wojtas structures is by removing one group from incomplete $TD$:

**Lemma 5.18** 1. *If a $TD(k+1, n)$ exists, then a $PWS(k, n; 0; \{1\}, n)$ exists.*

2. *If an $ITD(k+1, n; h)$ exists, then a $PWS(k, n; h; \{1\}, h)$ exists. If in addition a $TD(k, h)$ exists, then a $PWS(k, n; 0; \{1\}, n-h)$ exists.*

3. *If an $ITD(k+1, n; b_1, \ldots, b_s)$ exists, and $TD(k, b_i)$ exist for $2 \leq i \leq s$, then a $PWS(k, n; b_1; \{1\}, b_1)$ exists.*

Using Wilson–type constructions, more general Wojtas structures can be made. Here is a variant of Theorem 5.1, using the same notation:

**Theorem 5.19** *Suppose that a $TD(k+\ell,t)$ exists. Let $\mathcal{D} \subset \mathcal{B}$. so that if $B \cap D \subset E_i$ for distinct $B, D \in \mathcal{D}$ and $1 \leq i \leq \ell$, then $w_i^B = 0$. Suppose that for each $B \in \mathcal{B} \setminus \mathcal{D}$, there exists*

$$TD(k, m + \sum_{i=1}^{\ell} w_i^B) - \sum_{i=1}^{\ell} TD(k, w_i^B).$$

*Suppose that for each $1 \leq i \leq \ell$, there exists a*

$$TD(k, \sum_{j=1}^{t} w_{ij}) - \sum_{j=1}^{t} TD(k, w_{ij}).$$

*Then there exists a*

$$WS(k, mt + \sum_{i=1}^{\ell} \sum_{j=1}^{t} w_{ij}, \mathcal{M})$$

*where $\mathcal{M} = \{m + \sum_{i=1}^{\ell} w_i^B : B \in \mathcal{D}\}$.*

When a single level is used, one can in fact make partitioned $WS$:

**Theorem 5.20** *Suppose that a $TD_\mu(k+1,t)$ exists. Suppose that for each $B \in \mathcal{B}$, there exists*

$$TD_\lambda(k, m + w_1^B) - TD_\lambda(k, w_1^B),$$

whenever $w_1^B$ *is nonzero. Let $\zeta$ be the number of $\{w_{1j} : 1 \leq j \leq t\}$ which are zero. Then we obtain a Wojtas structure $WS(mt + \sum_{j=1}^{t} w_{ij}, \{1, m, \sum_{j=1}^{t} w_{ij})\}$ in which all blocks of size $m$ form $\zeta$ holey parallel classes for the hole of size $\sum_{j=1}^{t} w_{ij}$.*

Numerous variants are possible as well, but we do not consider them here.

## 5.7 Thwarts

Naturally, applications of Wilson's theorem depend on the presence of appropriate ingredients, and a natural question is to determine the ways in which the blocks of a $TD(k+\ell,t)$ intersect the points of nonzero weight in the $\ell$

"extra" groups. With this in mind, we give a definition. Let $\ell$ be a non-negative integer, and let $\mathcal{I} = \{i_1, \ldots, i_s\}$ with $0 \leq i_1, i_2, \cdots, i_s \leq \ell$. Further suppose that $0, s_1, s_2, \cdots, s_\ell \leq t$. Let $(X, \mathcal{G}, \mathcal{B})$ be a $TD(k+\ell, t)$ with $\mathcal{G} = \{G_1, \ldots, G_k, E_1, \ldots, E_\ell\}$. Then an $(\ell, \mathcal{I}, s_1, s_2, \ldots, s_\ell)$ *-thwart* is a set $S = \bigcup_{j=1}^{\ell} S_j$, where $S_j \subseteq E_j$ with $|S_j| = s_j$ for each $1 \leq j \leq \ell$, such that for every $B \in \mathcal{B}$, $|B \cap S| \in \mathcal{I}$.

Thwarts provide a convenient notation for simpler applications of Wilson's theorem, in which it is sufficient to know the number of points of intersection of each block with the points of nonzero weight in the extra groups. When different weights are chosen, however, more detailed structural information is required. Here we consider the structure of various thwarts, and we defer discussion of weightings until §9.3.

Given a set $\mathcal{I}$, let $\overline{\mathcal{I}}_\ell = \{\ell - i : i \in \mathcal{I}\}$.

**Lemma 5.21** *If a $TD(k+\ell, t)$ contains an $(\ell, \mathcal{I}, s_1, \ldots, s_\ell)$-thwart, it also contains an $(\ell, \overline{\mathcal{I}}_\ell, t-s_1, \ldots, t-s_\ell)$-thwart.*

## *Levels*

The simplest thwarts are found by simply truncating $\ell$ groups in each possible way to obtain:

**Lemma 5.22** *Let $\ell$ be a positive integer, and let a $TD(k+\ell, t)$ exist. Then for all choices of integers $s_1, \ldots, s_\ell$ satisfying $0 \leq s_i \leq t$ for $1 \leq i \leq \ell$, the TD contains a $(\ell, \{0, 1, 2, \ldots, \ell-1, \ell\}, s_1, \ldots, s_\ell)$–thwart.*

Often we refer to such a thwart as $\ell$ *levels* in the $TD$.

## *Spikes and Stairs*

If all of the points of nonzero weight are on a single block, we obtain an $(\ell, \{0, 1\}, 1, 1, \ldots, 1)$–thwart, which we call a *spike*. Every $TD(k+\ell, t)$ contains such a spike.

We can choose one point to be of nonzero weight on each level, so that no block intersects the points of nonzero weight in more than two points. We call the resulting $(\ell, \{0,1,2\}, 1, 1, \ldots, 1)$–thwart a *stair*. Stairs are essentially the analogue of arcs in projective planes. In fact, if the $TD(k+\ell, t)$ is the truncation of the $TD(t+1, t)$ arising from the desarguesian plane, the existence of an oval in the plane ensures that the stair is present for all choices of $\ell$. If the $TD(k+\ell, t)$ arises in another way, we cannot assume to inherit the structure of a plane. Nevertheless, if $\binom{\ell-1}{2} < t$, simple counting ensures that the stair is present.

One can find an intermediate structure between spikes and stairs. Suppose that one point of nonzero weight is chosen on each extra group so that there are $s$ blocks that are disjoint on the extra levels and intersect the extra levels in $x_1, \ldots, x_s$ positions, respectively. This is an $(\ell, \{0,1,2,x_1,\ldots,x_s\}, 1, 1, \ldots, 1)$–thwart, provided that no block intersects one of the $s$ chosen blocks in more than two points of nonzero weight in the extra levels. Call this a *generalized stair*, and use the notation $(\ell; x_1, \ldots, x_s)$–*stair*. The presence of such thwarts is not as easily checked, but when $s = 1$ it suffices to ensure that $\binom{\ell-1}{2} - \binom{x_1}{2} + 1 < t$. Another case is when $s = 2$ and $x_1 + x_2 = \ell$; then the thwart is always present.

## *Stairs, Spikes and Levels*

One can take a stair, spike or generalized stair on some levels, and truncate the remaining levels. For example, a spike involving $u$ levels and a $v$ disjoint levels truncated to $s_1, \ldots, s_v$ points leads to a $(u+v, \{0,1,\ldots,v+1, v+u\}, 1^u s_1 s_2 \cdots s_v)$–thwart, in which only one block of size $v+u$ is present if $u > 1$.

Similarly, one can take a stair or generalized stair together with some truncated levels. For example, to an $(\ell; x)$–stair, we can append a truncated level on $s \le t - \binom{\ell}{2} + \binom{x}{2} + 1$ points in a $TD(k, t)$ to obtain a $(\ell+1, \{0,1,2\}, 1^\ell s^1)$–thwart (in other words, adding a truncated level does not introduce a new intersection, provided that the truncated level is short enough).

Numerous variations are possible; the stair or spike can meet the added level(s) in one (or more) point(s). An $(\ell; x_1, x_2)$–stair could also have a further truncated level, and the blocks of size $x_1$ and $x_2$ could each meet or miss the truncated level.

### *Subplanes*

When a projective plane of order $t$ contains a subplane of order $s$, the corresponding $TD(t+1,t)$ contains an $(s+1,\{0,1,s+1\},s,s,\cdots,s)$–thwart. If the subplane is a Baer subplane, then the "0" can be omitted.

Instead deleting a point outside the subplane from the projective plane, we obtain an $(s^2+s+1,\{0,1,s+1\};1^{s^2+s+1})$–thwart.

### *Subsquares*

When a $TD(k,t)$ contains a $TD(3,t)$ having a "subsquare", i.e. a sub-$TD(3,s)$, then the $TD(k,t)$ contains a $(3,\{0,1,3\},s,s,s)$–thwart. When $t=2s$, the "0" can be omitted.

### *Affine Subplane*

When a projective plane of order $t$ contains an affine subplane of order $s$, the $TD(t+1,t)$ contains a $(s+1,\{0,1,s\},s-1,s-1,\cdots,s-1)$–thwart.

### *Trinity*

Wojtas [100] observed that one can truncate three levels but obtain a thwart with blocks interecting in 1, 2 and 3 points only — none in 0 points. The precise condition under which one can obtain such a $(3,\{1,2,3\},s_1,s_2,s_3)$–thwart is *open* for $TD(k,t)$ in general, although some bounds on $s_1,s_2,s_3$ are given in [30]. When the $TD(3,t)$ involved arises from a cyclic latin square (which can be assumed if we are free to choose three groups of the $TD(t+1,t)$ from the desarguesian plane of prime order $t$), a sufficient condition is that $s_1+s_2+s_3 \geq 2t-1$. The conditions for arbitrary $TD(k,n)$ seem very difficult; see [30] for some other observations in this regard.

## 6 DIRECT CONSTRUCTIONS

Until this point, we have concentrated on recursive methods, and despite a large collection of constructions being introduced, we have failed to construct any examples. Let us remedy that situation. Lemma 4.1, together with the well–

known fact that projective planes exist (at least) for all prime power orders, gives the main set of basic ingredients:

**Theorem 6.1** *If $t$ is a prime power, a $TD(t+1,t)$ exists.*

Surprisingly little else in the way of *general* direct constructions is known, although much is known from clever hand and machine computations in specific cases. The main device used is to assume that the $TD$ has a "reasonably large" automorphism group acting on it, and to use the structure of the automorphisms to reduce the computational search.

We require some basic definitions. Let $(\Gamma, \odot)$ be a group of order $g$. A $(g,k;\lambda)$-*difference matrix* is a $k \times g\lambda$ matrix $D = (d_{ij})$ with entries from $\Gamma$, so that for each $1 \leq i < j \leq k$, the multiset

$$\{d_{i\ell} \odot d_{j\ell}^{-1} : 1 \leq \ell \leq g\lambda\}$$

contains every element of $\Gamma$ $\lambda$ times. When $\Gamma$ is abelian, typically additive notation is used, so that differences $d_{i\ell} - d_{j\ell}$ are employed.

A $(g,k;\lambda,\mu;u)$-*quasi-difference matrix* (QDM) is a matrix $Q = (q_{ij})$ with $k$ rows, with each entry either empty (usually denoted by $-$) or containing a single element of $\Gamma$. Each row contains exactly $\lambda u$ empty entries, and each column contains at most one empty entry. Furthermore, for each $1 \leq i < j \leq k$, the multiset

$$\{q_{i\ell} - q_{j\ell} : 1 \leq \ell \leq \lambda(g-1+2u)+\mu, \text{ with } q_{i\ell} \text{ and } q_{j\ell} \text{ not empty}\}$$

contains every nonzero element of $\Gamma$ $\lambda$ times, and contains 0 $\mu$ times.

The essential connections with transversal designs follow:

**Lemma 6.2** *1. A $(g,k;\lambda)$-difference matrix gives a $TD_\lambda(k+1,g)$.*

*2. A $(g,k;\lambda,\mu;u)$–quasi–difference matrix with $\mu \leq \lambda$ gives a $TD_\lambda(k,g+u) - TD_\lambda(k,u)$.*

*3. A $(g,k;1,0;u)$–quasi–difference matrix gives an $ITD^{g-u(k-2)}(k,g+u;u)$.*

The latter statement gives one means to construct master designs for Du's variation, Theorem 5.7.

One main device for constructing quasi–difference matrices is to use $V(m,t)$ vectors. See [27] for a definition, and for the following:

**Theorem 6.3** *A $V(m,t)$ vector gives a $(mt+1, m+2; 1, 0; t) - QDM$. A $V(m,t)$ vector exists if $m$ and $t$ are not both even,*

1. *whenever $mt+1 \leq 5000$, $m-1 \leq t$, $m \leq 10$ and $mt+1$ is prime, except when $m=9$ and $t=8$, as no $V(9,8)$ exists.*
2. *whenever $mt+1 \leq 5000$, $m-1 \leq t$, $m \leq 6$ and $mt+1$ is a prime power, except when $m=3$ and $t=5$, as no $V(3,5)$ exists.*

Related computational constructions for $(mt+1, m+2; 1, 0; t) - QDM$ are reported in [28]:

**Theorem 6.4** *A $(mt+1, m+2; 1, 0; t) - QDM$ exists when $mt+1$ is a prime, $mt+1 < 5000$, $m-1 \leq t$, and*

1. *$t \equiv 2 \pmod 4$ and $m \in \{2, 4, 6\}$; or*
2. *$t \equiv 4 \pmod 8$ and $m \in \{2, 4\}$.*

Abel [2] found another useful family:

**Theorem 6.5** *For $11t+1$ a prime, there exists a $(11t+1, k; 1, 0; t) - QDM$ exists for $k = 11$ if $198 < 11t+1 < 600$ and for $k = 12$ if $600 < 11t+1 < 992$.*

He also found $(9 \cdot 4 + 1, 9; 1, 0; 4)-$ and $(9 \cdot 8 + 1, 9; 1, 0; 8) - QDM$.

Despite these few more general computational results, most direct constructions are one–of–a–kind. For $TD$, we summarize in Table 1 known direct constructions, not obtained by one of the three previous constructions of $QDM$.

In some cases, a direct construction yields an idempotent $TD(k,t)$ , which is a $PITD(k,t;1^t)$. In Table 2, direct constructions of idempotent $TD$ having the same blocksize as the largest known $TD$ on the same parameters are reported. Ganter, Mathon and Rosa [41] actually construct a $TD(4,10)$ having four disjoint parallel classes, the maximum known to date.

| Order | Blocksize | Reference(s) |
|---|---|---|
| 6 | 3 | [89] |
| 10 | 4 | [18] |
| 12 | 7 | [53] |
| 14 | 5 | [90] |
| 15 | 6 | [76] |
| 18 | 5 | [94] |
| 20 | 6 | [91] |
| 21 | 7 | [65] |
| 22 | 5 | [94] |
| 24 | 7 | [3, 72, 104] |
| 26 | 6 | [26] |
| 28 | 6 | [1, 73] |
| 30 | 6 | [5] |
| 33 | 7 | [2] |
| 34 | 6 | [2] |
| 36 | 7 | [73] |
| 38 | 6 | [5] |
| 39 | 6 | [60] |
| 40 | 9 | [3] |
| 42 | 5 | [94] |
| 44 | 6 | [5] |
| 48 | 7 | [3, 73] |
| 51 | 7 | [2] |
| 52 | 6 | [1] |
| 55 | 7 | [60] |
| 56 | 9 | [60] |
| 80 | 11 | [3] |
| 112 | 15 | [3] |
| 160 | 11 | [3] |
| 176 | 16 | [3] |
| 208 | 16 | [3] |
| 224 | 15 | [3] |
| 352 | 20 | [3] |
| 416 | 20 | [3] |
| 544 | 20 | [3] |
| 640 | 11 | [3] |
| 896 | 15 | [3] |

**Table 1** Direct Constructions for $TD$

| Order | Blocksize | Reference(s) |
|---|---|---|
| 6 | 3 | – |
| 10 | 4 | [41] |
| 14 | 5 | [11] |
| 18 | 5 | [106] |
| 20 | 6 | [2] |
| 22 | 5 | [6] |
| 34 | 6 | [2] |
| 38 | 6 | [2] |
| 42 | 5 | [11] |
| 44 | 6 | [2] |
| 52 | 6 | [2] |

**Table 2** Direct Constructions for Idempotent $TD$

A large number of *ad hoc* constructions for incomplete $TD$ appear in the literature; we do not attempt to catalogue them all here. In Table 3, we report some of the small incomplete $TD$ that have been constructed directly.

| | | | |
|---|---|---|---|
| $ITD(4,6;2)$ | [40] | $ITD(5,8;2)$ | [84] |
| $ITD(6,10;2)$ | [23] | $PITD(5,9;2^1 1^7)$ | [108] |
| $PITD(5,11;2^1 1^9)$ | [85] | $ITD(5,12;3)$ | [84] |
| $PITD(5,12;2^6)$ | [10] | $ITD(6,15;3)$ | [28] |
| $PITD(5,14;3^1 1^{11})$ | [4] | | |

**Table 3** Some Incomplete $TD$

Of course, many more incomplete $TD$ have been constructed directly. Some constructions of $TD$ have proceeded by making an $ITD$ with a hole and filling the hole; see [1, 5, 26, 94]. For $ITD(4,t;h)$, see [49] and references therein. For $ITD(5,t;h)$, see [4, 38, 39] and references therein for a number of direct constructions. For $ITD(4,hn;h^n)$, see [36, 86]. For $ITD(5,hn;h^n)$, see [10, 12, 55, 64, 85]; Dinitz and Stinson [36] also give some $ITD(6,2n;2^n)$ and $ITD(8,2n;2^n)$.

Colbourn [27] gives a number of constructions for $QDM$ leading to $ITD$, and some sporadic examples appear in [61].

In terms of sporadic designs that find uses, we mention also the elliptic semiplane of Baker [8], which is a $\{7\}-GDD$ of type $3^{15}$; and the $\{9\}-GDD$ of type $3^{33}$ by Mathon [59].

## 7 A KNOWLEDGE BASE FOR *ITD*

In planning the *CRC Handbook of Combinatorial Designs*, it became amply clear that one of the fundamental tables used by combinatorial design theorists, and by many researchers in applications areas, is the table of the best current lower bounds on the number of mutually orthogonal latin squares. Brouwer's table [20] has had an almost unbelievable impact on our ability to establish existence theorems for designs of all types.

However, as the years have passed since the last circulation of [20], constructions have required progressively more general types of incomplete MOLS, and tables of these have not been compiled. Our first task was to decide what objects to tabulate for the *Handbook*, and our second was to decide what objects to tabulate for uses in constructions to make the first set of tables.

Our next task was to determine how to incorporate knowledge of the many constructions in §2–6 into an "inference engine" to apply the constructions. Finally, the main task was to apply this battery of constructions to produce the output tables desired.

§7–9 gives a summary of our efforts to build a computational environment. We first report on the fundamental objectives. Then we develop an outline of the architecture of the system, and discuss basic implementation decisions. We then describe some of the routines in more detail, in particular explaining the variants of Wilson's theorem actually coded. Finally, we examine what we believe to be the failures and successes of our approach.

## 7.1 Fundamental Objectives

We quickly realized that we could not report all of the consequences of applying every variant of every construction to all known ingredients. Thus we made some decisions initially about what information to report in the *Handbook*. Brouwer's table [20] set a standard in reporting the existence of MOLS to order 10000, and we felt that any shorter table would negatively impact the usefulness of the results. In view of the importance of *idempotent* MOLS ($PITD(k,t;1^t)$), we set as a goal the inclusion in this main table of an indication of when an idempotent solution can be found, again to order 10000.

Turning to incomplete MOLS, the candidates for what to tabulate are numerous. However, even a cursory examination of the literature shows that incomplete MOLS with one hole ($ITD(k,t;h)$) and uniform MOLS with holes ($PITD(k,hn;h^n)$) both play major roles. We determined to tabulate MOLS of side $t$ with one hole of side $h$ for $t \leq 1000$ and $h \leq 50$, and uniform MOLS with holes of type $h^n$ for $h \leq 50$ and $n \leq 20$.

These fundamental objectives place minimum requirements on what must be tabulated and computed, but place no maximum on how much information the system may store and use to produce the basic output tables.

Other essential objectives beyond the output became clear. Justifications for table entries are needed in a machine–generated form that can be read (perhaps with effort) and deciphered. We did not set as a basic requirement that these justifications be presented with the table, as we determined that the approximately 50000 justifications for the table of MOLS with one hole would not be terribly edifying.

A final objective is that the environment must support the introduction of new ingredients, and new constructions, without starting "from scratch". In particular, efficiency of updating the table is a serious concern.

# 8 ARCHITECTURE OF THE SYSTEM

Here we explore the major decisions made in designing the computational environment. The first main decisions concern how to represent the knowledge contained in the results reviewed in §2–6. The required knowledge comes in a number of flavours:

1. parameters for which an $ITD$ is known to exist;
2. parameters for which other combinatorial structures are known to exist;
3. constructions for $ITD$ and other structures.

We first examine how this knowledge is stored, and then examine how consequences of the constructions are determined.

## 8.1 The Data Base

We have already examined the question of what structures are to be recorded for output. Two remaining decisions have a great impact on the behaviour of the system: what structures to *store*, and what structures to *use*. The distinction that we are drawing here is perhaps subtle. Structures to be "stored" are those to be explicitly placed as a record in our database. Structures to be "used" are those stored externally, and those generated by procedures in our environment.

We treat the question of what to store, and how to interact with this data base, here. A careful reading of the constructions in §2–6 suggests that a wide variety of structures are employed in constructions. We decided not to store anything except for $TD$ and $ITD$; knowledge about other structures is provided by procedures that can either (a) store a table internally, or (b) read a different database. Having limited our attention to $ITD$, we needed to make a decision concerning which types of $TD$ and $ITD$ to store. In principle, essentially any $ITD$ might be useful in one or more constructions. However, realistic limits on storage, and the need for constructions to have ingredients that are likely to be useful, dictates that one choose only certain types of $ITD$. After an examination of the literature, and *much* discussion, we decided to store

1. $TD(k,t)$ for $t \leq 10000$;
2. $PITD(k,hn;h^n)$ for $h=1$ and $n \leq 10000$, and for arbitrary $h \leq 50$ and $hn \leq 1000$;
3. $ITD(k,t;h)$ for $t \leq 1000$ and $h \leq 50$; and
4. $PITD(k,a+bn;a^1b^n)$ for $a=b=1$ and $n \leq 10000$, and for $a,b \leq 50$ and $a+bn \leq 1000$.

We also store an *authority* with each entry, and this authority is a plain text string.

To be precise, we do not store any actual $ITD$. What is stored, for parameters $t$ and $h$, for example, is the value $k-2$ where $k$ is the largest integer for which an $ITD(k,t;h)$ exists. Numbers stored are actually numbers of incomplete MOLS rather than the block size of the $ITD$, but of course this is a simple translation. One might argue that certain information is lost by storing only the value $k-2$, and not the whole $ITD$ — some of the constructions given do depend on the structure of the $ITD$. However, the dependence on the structure of the $ITD$ occurs primarily for $TD(t+1,t)$ arising from desarguesian planes, and these can be recognized from the parameters alone. Moreover, since a $TD(k,10000)$ has $10^8$ blocks, one questions the wisdom of storing it.

The wide variety of ingredients up to order 1000 suffices to handle constructions of $TD$ and idempotent $TD$ up to order 10000. But even more ingredients of order less than 1000 might be useful. We elected to "process" such ingredients as they are generated, rather than storing them.

Interaction with the database is carefully controlled. Each routine is permitted to read any database entry. However, updating the database is under the control of a small set of routines. The first of these, `interesting`, determines whether the $ITD$ lies within bounds on blocksize, holesize, and order. The main routine is `record`. It first determines whether the $ITD$ is interesting, and then performs some error–checking on it (more about this later). If it fails basic error tests, the system stops without saving any information from the current run — errors are considered to be fatal, always. If the $ITD$ passes, a determination is made by checking the current database entry to see if it is new. If it is interesting, apparently correct, and new, then the database is updated. Whenever an update occurs, output to the user is generated in the form of a command that indicates precisely which record was updated, and the authority for the update. The `record` routine goes on to explore certain consequences of the update; we return to this in the next section.

In order to permit distributed computation (loosely speaking), there is also a mechanism for updating the database entry without exploring the consequences of the update. The routine `rerecord` carries this out (the "interesting" check and error–checking are done regardless, so that one cannot subvert the error–checking by simply using `rerecord` instead of `record`).

This strategy isolates all of the database updates in one main routine. It prevents an entry from being updated without authority, and ensures that proper comparison against the previously best entry is made prior to update.

## 8.2 Procedural Knowledge

Our database excludes knowledge of other combinatorial structures, and also knowledge about the constructions of §2–6. We treat these in turn.

We planned for the eventuality that extensive databases might exist for block designs, $PBD$, symmetric designs, and so on. Since adequate databases of these do not exist at the present time, we encapsulated knowledge about existence results for each structure in a procedure. Each procedure contains a table as an internal data structure. However, the procedure is designed so that it can be easily replaced by a filter accessing a secondary database.

The representation of the constructions is a complex issue. Arguably, the constructions themselves are a form of data, and ought to be stored as such. One can imagine a large list of constructions, along with a general routine that "knows" how to apply a generic construction. We did not choose this path for two reasons. Firstly, the generality of the constructions mitigates against finding an efficient general mechanism for applying an arbitrary construction; perhaps more honestly, we should say that *we* saw no way to do this with reasonable efficiency. Secondly, it was not clear to us how to gain selective control over application of the constructions. Some are more fruitful than others, and more effort in exploring unusual parameter sets for the ingredients is worthwhile. For these reasons, we decided to represent constructions as procedures.

## 8.3 Marrying the Procedures and the Data

Procedures can be loosely classified into the following categories:

- *database interface*: routines that determine when and how to update the database.
- *generators*: routines that generate parameter sets for existing combinatorial objects. Some generate by simple table look–up in an internal data structure; others access the database and repeatedly apply a construction from §2–6.
- *fillers*: Once an object is generated, a fill routine for that class of object incorporates knowledge about equivalences to other objects, and about ways in which to "fill holes".
- *output and tracing routines*: routines to extract information in tabular form from the database. These do not alter the database.
- *utility routines*: simple procedures for commonly repeated operations.

Drawing a distinction between generators and fillers is key in our approach. Let us take an easy example. Parameters for existing difference sets can be generated; one wants to explore the consequences of the existence of one. It may lead directly to a number of pairwise balanced designs by considering residue classes. It certainly leads to a symmetric design. This is in turn a block design (which is a $PBD$). The block design can be truncated to form $IPBD$ and $GDD$. Each has its own way to be "filled" to form $ITD$ of various types, and these $ITD$ may themselves imply the existence of further $ITD$ by truncation, filling, or inflation. So our initial observation that it is easy to generate parameters for difference sets, while true, tells only a small part of the story. The situation is perhaps even more complicated with Wojtas structures, where the interaction of inflation and filling leads to numerous possibilities.

We examined every known construction with a goal of identifying what object(s) the construction generates, and how this object is converted (perhaps in many stages) to an interesting $ITD$. We then decoupled each construction into a generator routine that makes the first intermediate object, and a sequence of filler routines that convert one object into another, ultimately appealing to the database interface to decide whether our efforts have met with success. The main advantage is that, once a filler for a type of object is included, *all* routines (generators and other fillers) that make this class of object can simply pass it to the fill routine. This philosophy permeates the constructions of §2–5; in Wilson's Theorem 5.1, for example, the $ITD$ produced is in general not interesting by our rigid standard — but a filler for $ITD$ with disjoint holes may succeed in producing a large number of interesting $ITD$ from it!

One benefit that we anticipated (and encountered) is that constructions in the literature rarely exploit systematically the consequences of the intermediate objects constructed – it is just too tedious. Having fillers acting autonomously, however, permits unforeseen consequences to be discovered. The picture is not completely rosy, however. Often constructions that generate and fill in one step capture the interesting ways to fill the intermediate structure; although other ways to handle it may exist, they may lead to $ITD$ that have blocksize too small, or order too large, to be useful.

The normal "discovery" of an $ITD$ leads from a generator, through a sequence of fillers, to the database interface. There is no global plan; each routine knows what objects it can pass on to which other routines. Once at the database interface, we have discussed some of the work that `record` does. But `record` also explores the consequences of what is reported to it, so it in essence acts as a filler itself. Specifically, `record` first determines whether the order of the object is too large for it to lead to anything interesting. Fortunately, all of the constructions make $ITD$ from $ITD$ (and other objects) of smaller or equal order. The constructions also all have the property that they do not increase the blocksize. If the $ITD$ given to `record` is too large or the blocksize is too small, it simply abandons it.

If the $ITD$ given to `record` has small enough order and large enough blocksize, then *if it is not interesting*, or *it improves the blocksize in a database entry*, `record` passes it to `fill`. The `fill` routine carries out the filling and truncation operations in §2, and also makes the Wojtas structures if possible as in Lemma 5.18. It also may generate a sequence of further `record` operations. Once `fill` has attempted to record new entries, it invokes `inflate` to carry out the standard inflation from §2, and invokes `fillwojtas` to handle the Wojtas structures. Some care is required here to ensure that one doesn't simply cycle through these filler routines and `record`. The primary danger is with equivalent objects; these are filled only when they are *new* and *interesting*.

Evidently, a substantial effort is invested in every *ITD* that is reported to `record`; this is consistent with our philosophy that it does not matter where the ingredient came from — one wants to capture its consequences. This is particularly important if the *ITD* itself is not interesting, i.e. it alone will not alter a database entry. Any consequences of it not captured at this point are lost, unless it is generated again (by the same or a different method).

Nevertheless, we do draw the line short of exploring all consequences. We do not explore, for each *ITD* coming to `record`, its uses beyond simple truncation in making *PBD* and *GDD*; nor do we explore its use either as a master design or as an ingredient in Wilson's theorem. Interesting *ITD* may update a database entry, and if they do, their consequences in making *PBD* and *GDD*, and in Wilson's theorem, can be explored by the appropriate generator at a later time.

## 8.4 Operation of the System

The system starts with an empty database. A maximum order, a maximum holesize, and a minimum blocksize, are selected; these serve as parameters for `interesting` to determine what can be placed in the database. One then simply invokes generators in any order desired. Of course, some are likely to be more effective than others. Wilson's theorem with no ingredients has no consequences, so one must first "prime the pump" (or "prime power the pump"). A sensible regimen is to first apply generators arising from direct constructions – prime powers, difference and quasi–difference matrices, and `inputcases`. Then some fodder is present for the generators using symmetric designs, block designs, and *PBD*. Next constructions that make *PBD* and *GDD* from *ITD*, and Wilson's theorem can be applied.

No matter what order one applies the generators in, however, there is a problem. The generators cannot be ordered so that one generator only uses consequences of an earlier generator. Generators that employ ingredients from the database must be repeatedly applied until all consequences relevant to the blocksize, holesize and order specified are uncovered.

Even this hides an important problem. If, when a database entry is updated, `record` passes the *ITD* to `fill` but `fill` does not find appropriate ingredients to fill it, we need a mechanism to discover when these ingredients become available. It won't suffice simply to generate the *ITD* again, as `record` will recognize that the *ITD* does not cause a database update the second time, and hence rejects it without passing to `fill`. The answer may appear terribly easy — just fill it anyways! But it is an enormous waste of time to repeatedly fill $ITD$ whose consequences have been explored already. It seems difficult, moreover, to determine *a priori* whether new ingredients are now present that might lead to new consequences. Hence we added a facility for doing a forced fill; `forcefill` is a simple routine that passes each database entry in turn to `fill`, with the result that the new consequences, if any, are uncovered.

The system can run interactively or in batch mode. In either case, the permanent database is updated only at the end of the run, so that any abnormal or error termination does not corrupt the database.

## 8.5 Error Checking

A major concern in making tables is ensuring that the resulting information is accurate. The main error checking is in `record`, although every generator and filler routine attempts basic error checks as well. Since we do not have the actual $ITD$, there is no way to be sure that the construction actually produces the object. Hence we resorted to three types of error checks. The first is completely automated, and verifies that the parameters of the $ITD$ meet known necessary conditions. Some redundancy is present to facilitate simple error checks; for example, for a $PITD(k, a + bn; a^1b^n)$, the values $a + bn$, $a$, $b$ and $n$ are all separately reported, allowing a basic cosmetic test to be made.

Initial runs of the system were provided with a second database of $TD$, kindly provided by Andries Brouwer and run in 1993 (this database contains information from the constructions in [20, 25], along with some recent direct constructions). Any variations between the database entries for $TD$ and this revised table of Brouwer were reported, and each was examined by hand to ensure the correctness of the construction used. This was, without a doubt, the best debugging tool that we had; hand verification of all of the $TD$ entries would be nightmarish.

Finally, `record` reports explicitly objects for which the blocksize exceeds the square root of the order. These are rare and, although permitted by the necessary conditions, are remarkable enough to warrant some specific hand checks. It would be irresponsible to suggest that no errors could pass these error checks unnoticed. Nevertheless, one's confidence can be increased by substantial hand–checking and tracing, and this we have done.

## 8.6 Authorities

Perhaps the major disadvantage of having all of the routines decoupled is that no one routine or construction is solely responsible for causing a database update to occur. Nevertheless, one must be able to reproduce the "reasoning" applied by a sequence of routines in eventually effecting a database update. Brouwer [20] assigned to each entry a number, which indicated the single proposition responsible for the entry. The reader is left to deduce the parameters employed, which can typically be done with little effort. Our case is somewhat more complex; the set of constructions is richer, the ingredients are more general, and the constructions themselves are multistage. Our authorities are commensurately more complex. The price one pays for this is that the table of authorities is approximately seventy times larger than the table of actual numerical entries; hence reporting even an abbreviated authority becomes problematic. Simply saying "Wilson" is too little information to be useful; one must at least say which variant of the basic Wilson theorem, which thwart, and what weights are used. This in turn often leads to an intermediate structure, and one must say with what authority it was filled. We abandoned the thought of making a small table of authorities, and decided to retain authorities online that suffice for us to reconstruct the mechanism that led from generator to database entry.

Our rule of thumb is that each routine that generates or fills an object says, in broad terms, what it did to the object. A generator for symmetric designs, for example, may use the simple authority `symdes(9)` for the projective plane of order 9. Fillers indicate where they got the main ingredient that they are filling (i.e., they employ the authority provided for it), and in general do *not* indicate what ingredients are being used in the fill. For example, `fillbibd(symdes(9))` is an authority for a basic block design fill of a projective plane of order 9. Now `fillbibd` does not say what it did to the object; each routine is meant to be simple enough that this can be deduced without guesswork, but perhaps with some effort.

The biggest headache is with variants of the Wilson construction. Here we typically give in the authority the variant used, the type of the thwart, the parameters of the master design, and the weight $m$ on the main levels; the luckless reader is left to deduce the weights on the extra levels, and the parameters of the other ingredients. For example, `W1 N(9) wt 7` is an application of Theorem 5.1 using $m = 7$ on a $TD(k, 9)$; one must deduce from the parameters of the design reported what weights have been used, and hence what ingredients are needed. This is not as hard as it sounds, with some practice. One has the advantage of having a table of what ingredients exist, so that most potential choices for weights are eliminated immediately.

One might envisage recording sufficient information that each authority could be systematically translated into an existence proof. We saw no compelling reason to do this; it is true that a library would then exist that one could consult for verification of every entry — but it would be, for the most part, a very boring library! Therefore, we adopted the approach that authorities would remain internal to the environment, for verification purposes. Moreover, authorities are not proofs; rather, they constitute a high level trace which, together with the actual code, can be used with diligence to construct a proof.

# 9 IMPLEMENTING THE SYSTEM

Here we discuss some issues in the implementation of the system, and provide some detail on what precisely is realized from §2–6 in the system as it stands today (June 14, 1996). We do not attempt a complete description of every variant, but content ourselves to present the primary content. If no statement is made to the contrary, one should assume that the variants alluded to are those mentioned in §2–6. Applications of Wilson's theorem form a special case; there we are more explicit about what is included.

## 9.1 The Platform

Choosing an underlying platform on which to build the environment outlined is not an easy task. We have two main data structures with which to deal: tables of integers and tables of strings representing authorities. While the latter is primarily being written to and rarely consulted (by the system), the former is consulted very frequently and updated on occasion. Dynamic hash tables provide the right combination of easy access, sparse storage, and extendability.

This requirement led us to consider MAPLE, which provides dynamic hash tables as a primitive data structure. MAPLE also afforded the simple number–theoretic routines that we needed. Naturally, a major concern with any symbolic algebra environment is efficiency, but we decided that the presence of appropriate data structures, the support for mathematical computations, and the ease with which new routines could be prototyped and developed, outweighed our concerns about efficiency.

Hence the system has been developed as a package in MAPLE, and currently it runs only under Release 2 of MAPLE V.

## 9.2 Generators I: Other Designs

We have already seen the basic mechanism by which an entry is recorded, and its easier consequences explored automatically by `fill`, `inflate`, and `fillwojtas`. This frees the remaining routines from concern with the simpler bookkeeping. Here we consider the classes of generators, other than Wilson–type; the discussion must be read in conjunction with the discussion of fillers to obtain a reasonable understanding.

The simplest generator is `inputcases`. It just invokes `record` to capture sporadic direct constructions that are known; see §6. Next simplest are a number of generators that encapsulate direct constructions of $ITD$: prime powers, $V(m,t)$ vectors and related $QDM$, and the Dinitz–Stinson construction [36].

Turning to more complex objects, `bibd` is a generator that contains information about existence of block designs, cyclic block designs, resolvable block designs, and sporadic $PBD$. This information is taken essentially as is from [113, 116, 117]. Each is passed to an appropriate fill routine. In addition, truncations as in §4.1 are carried out, and the $PBD$ and $GDD$ produced are passed to appropriate fillers. Along similar lines, `symdes` carries information about symmetric designs; it invokes routines to examine the consequences of subplanes, ovals, and line–flips on these. The routine `diffset` concerns residue classes in difference sets. Here the difference sets have to be known explicitly – parameters alone won't do.

The next main class contains routines `tdgdd`, `itdgdd` and `abntdgdd`. Each makes $PBD$ and $GDD$ from $ITD$ as discussed in §3.3. We reiterate that only those $ITD$ that have been stored in the database become candidates for making $PBD$ and $GDD$. Some information is lost thereby, since truncations of objects with three different hole sizes are not treated, for example.

Before proceeding to Wilson's theorem, we want to spend a moment to point out that, while Wilson's theorem is by far the most prolific generator, one ought not to dismiss the more "classical" techniques using designs. Two reasons come to mind. Wilson's theorem permits many of the same truncations as does general truncation of $PBD$; comparing §3.3 and §5.7 reveals this. However, a quick comparison is not sufficient to notice that truncations for $PBD$ constructions are more general — for example, removing a block from a $TD$ gives useful $PBD$ [9], while no analogous thwart is of any use in Wilson's theorem. A second reason is that while Wilson's theorem provides the best bound in most cases, the bound is rarely large. Sporadic bounds that arise from configurations in projective planes, while not frequent, are often substantially better. Thus care must be taken to obtain consequences of other designs; this is especially true since Wilson's theorem can be voracious in its appetite for ingredients.

Adding new generators is a relatively easy task, whether they make block designs with extra properties, $PBD$ with extra properties, $GDD$, symmetric designs, cyclic difference sets, or some flavour of $ITD$. If they make something else that may be useful, corresponding filler routines must be supplied as well.

## 9.3 Generators II: Wilson's Theorem

There can be little doubt that the many variations of Wilson's theorem produce the vast majority of the results. This has caused an epidemic of versions of the theorem; some get so complex that we cannot find any ingredients to actually apply them, and some are just too constrained to take advantage of the ingredients that we know. Striking a balance is important, but not easy. Which variants should be used, with which types of thwarts, and what weightings on the thwarts? We have adopted the strategy here of starting with general statements and evolving them into useful constructions in two stages; first, in §5.7 we described possible structures on the points of nonzero weight in the extra levels, and here we describe weights applied to those points in the thwart.

The amazing flexibility of Wilson's theorem is a blessing and a curse. The blessing is obvious, the curse perhaps less so. It is that there does not appear to be any sane manner in which to explore all of the consequences of the theorem, even with the known ingredients and a reasonable bound on the order. Almost any $ITD$ might be useful as an ingredient, and almost any thwart might happen to have sufficient ingredients present for the right weights.

Our initial goal was to have generators to cover the applications of Wilson's theorem and its variants that actually appear in the literature. We exceeded this goal in many areas, and fell short in others. In retrospect, the reason for falling short is completely obvious: We do not have enough ingredients stored to cover every application discussed in the literature. The two worst omissions are $ITD(k,t;h_1,h_2)$ and $ITD^s(k,t;h)$; in both cases, we were able to generate sufficient information about these "on the fly" to cover the primary applications, but their omission necessarily limits what the environment can discover.

We proceeded by implementing generators for a number of simple thwarts with simple weighting functions, and generalized those that were productive to more complex thwarts and/or more general weights. It cannot be said often enough that the approach of "implement every routine as generally as possible" is doomed to fail unless one is Methuselah with a lot of computers. This means inevitably that we are reporting *some* consequences of Wilson's theorem. Hopefully we cover the most interesting and most productive variants, but we have no assurance even of that. With this proviso in mind, it is perhaps most important that we say what is included. We also report on some utility routines that are used throughout the Wilson–type generators.

We classify the applications by the type of thwart.

## *One Level*

When the thwart is a single level, the most extensive implementation has been done. All of the variants of Wilson's theorem are treated to some extent here. In particular, we try all possible assignments of weights to the points of the thwart. This is accomplished by using a simple routine for discrete convolution, as follows. For each choice of weight $m$ on the main levels, we determine a vector $(x_0, x_1, \ldots, x_c)$ in which $x_i$ is the largest value of $k$ for which an $ITD(k+2, m+i; i)$ is known. The limit $c$ can be easily calculated given a lower bound on the block size of interest, and $m$. The *$s$–fold convolution* of the vector is a vector $(y_0, \ldots, y_{cs})$, where

$$y_j = \max(\min(x_{\gamma_1}, \ldots, x_{\gamma_s}) : 0 \leq \gamma_i \leq c \text{ for } 1 \leq i \leq s, \text{ and } \sum_{i=1}^{s} \gamma_i = j).$$

A utility routine calculates $s$–fold convolutions, permitting us to determine the best way to put a total weight of $j$ on a level with $s$ points of nonzero weight.

Variants using incomplete master designs (with one hole), and incomplete ingredients on the main levels (with spanning sets of holes) are included.

## *Two Levels*

We are unable to implement all weightings on all two level thwarts, because we have stored no information explicitly about $ITD$ with two holes. Nevertheless, a substantial amount can be done. First we choose a weight $m$ on the main levels. In the simplest implementation, we put weights 0 and 1 on the points of one level, and then determine what the maximum block size is should we put a weight of $i$ on a point of the second level. Again we obtain a vector $(x_0, x_1, \ldots, x_c)$ as in the one level case; discrete convolution then permits us to try all possible distributions of weights on the second level.

This is generalized somewhat. Suppose that $d$ is a divisor of $m$ and write $m = de$. For each choice of divisor $d$, we assign weights on one level from $S = \{0, 1, d\}$, $\{0, d\}$, or $\{1, d\}$. Since a weight of $d$ is to be permitted on the first level, we must determine whether an $ITD(k, m+d+i; d, i)$ exists to decide whether a weight of $i$ can be used in the second level. We do not have this information, *but* if a $PITD(k, m+d+i; i^1 d^{e+1})$ exists, and a $TD(k, d)$ exists, we can produce such an ingredient. In this way we again determine the possible weights on the second level and use discrete convolution.

Fewer variants have been explored here, due to the complexity of the weightings encountered. Only weights of 0 and 1 have been considered on the extra levels when the master design is incomplete, for example.

## *One Spike*

In the case of one spike (§5.7), again all possible weight assignments to the spike have been examined, using discrete convolution. The implementation actually replaces the role of the spike by a sub-$TD$ (a spike is just a sub-$TD$ of order 1), and weights on all of the points of the sub-$TD$ among the extra levels are chosen independently.

In [25], the Remark following Corollary 1.4 indicates that one ingredient is not needed if the spike has maximum size; however, this appears not to be correct, and our spike implementation does not use that remark.

## *Spike and a Level*

One spike and one level are treated as follows. In the first place, weights 0 and 1 are placed on points of the level, and arbitrary weights on points of the spike. Then secondly, weights 0 and 1 are placed on points of the spike, and arbitrary weights on points of the level. If the spike meets the level, the "corner" point plays a special role as any block meeting the corner which is not the spike block meets the thwart in *only* the corner point. This enables us to weight it with fewer constraints.

## *Spike and Two Levels*

A very basic implementation of a spike and two levels, using weights 0 and 1, was coded but led to few results, but much computation time!

## *Generalized Stair and a Level*

A number of different thwarts were tried here. First, a thwart is tried which is an $(\ell; x_1)$–stair and a single level, where the points of the level do not lie on a block that meets the stair in two or more points. The level may, however, share a "corner" with the generalized stair, which we assume (if present) to be on the block of size $x_1$. The generalized stair is given weight 1, and all possible weight distributions on the points of the level are considered.

Next we try a "goalpost" thwart. This is an $(a+b; a, b)$–stair with the $a$–block and $b$–block on different levels; then a level is added in which all points do not occur on blocks that meet the stair in two or more points. There is the possibility to add either or both of the corner points to the level, where the $a$– and $b$–blocks meet it. We may also consider the case where the $a$– and $b$–blocks meet the level in the same point. All of these variants are tried, but just for weight 1.

When no extra level is present, a goalpost is essentially two spikes, which may or may not meet at a corner, but are otherwise on disjoint levels. Here we tried weight 1 on one spike, arbitrary weights on the other spike, and an arbitrary weight on the corner (which is less constrained).

Next we try a "trident" thwart, which is a $(a+b+c; a, b, c)$–stair. Only weight 1 is attempted.

## *Trinity*

The three level thwarts that have all blocks meeting them in 1, 2 or 3 points (and the complementary thwarts which blocks meet in 0, 1 or 2 points) were vexating. Computing the bounds from [30] for such thwarts is not computationally easy. We implemented a routine which calculates a table that says, given the sizes of two levels, what the bounds are for the size of the third level so that the thwart can be assumed present. However, the only computation run to completion was for the case that the master $TD$ arises from a plane of prime order, making the bounds easy to compute. All weights used on the trinity three level thwart and its complementary thwart were taken to be 1. A variant was explored that extends the three level thwart to four levels, permitting intersections of size 1, 2, 3 and 4; however, it was not terribly productive.

### *Subsquares*

Here we tried weight 1 on subsquare thwarts and their complementary thwarts. We also tried the $(k, \{0, 1, k\}; t^k)$–thwarts arising from sub-$TD$.

### *Projective Subplanes*

We employed thwarts arising from Baer subplanes and the partitions into Baer subplanes, with all weights being 1. We employed projective subplanes of order 2 to make $(7, \{0, 1, 3\}; 1^7)$–thwarts, and assigned weight 1.

### *Affine Subplanes*

We employed the $(4, \{1, 3, 4\}; (t-2)^4)$ complementary thwart arising from puncturing a point from an $AG(2, 3)$, using weight 1.

This description is meant to convey the flavour of what is present for Wilson–type generators in the system. Each also generates Wojtas structures for inflation prior to filling, and each employs at least a rudimentary application of Wilson's theorem using ingredients that are idempotent on the main levels.

The Du variations (§5.3) are only implemented for master designs that arise from direct constructions of quasi–difference matrices, and no effort has thus far been made to extend their use in the system (lack of ingredients being a big hurdle to surmount).

## 9.4 Fillers

Fillers are provided for numerous intermediate structures encountered. We have examined `fill`, a basic filler for $ITD$, already; we have also encountered `fillwojtas`. It implements the filling and inflation of Wojtas structures discussed in §5.5.

Fillers exist in addition for block designs (using arcs when known), $GDD$, $PBD$, resolvable block designs, resolvable $PBD$, and symmetric designs. For separable $PBD$, the Bose–Shrikhande–Parker theorem is implemented as a filler.

One can see that fillers may invoke a number of other fillers in addition to `record`.

# 10 EXPERIENCE WITH THE SYSTEM

In at least one regard, our venture was a success. We succeeded in producing tables of $TD$ and $ITD$ for the *CRC Handbook of Combinatorial Designs*; see [114] and [115]. In some ways, our expectations were exceeded. For example, the surprisingly good results for idempotent $TD$ were not anticipated. Nor did we expect to make substantial improvements in the existence of $ITD(5, t; h)$ where substantial effort has been invested [38, 39]; but the computations provided a valuable set of updates, reported in [4] (along with new direct constructions).

## 10.1 A *Post Mortem*

The successes were matched by some disappointments. We review them here, so that when the inevitable time comes to make the next generation of MOLS table, some of our experience can be profitably used. The real disappointment is efficiency.

One would like to just tell the system that a new ingredient has been found, and have it be "clever" enough to explore what impact that might have; but this process is agonizingly slow. Depending on the ingredient, any time from seconds to days can be expended exploring its consequences. Generality and efficiency are fighting each other here, and our desire is to make the generators and fillers *even more general.* Clearly, some significant effort must be made to store more and recompute less; the balances here in generality/efficiency, and storage/time, are subtle. We cannot but feel that useful auxiliary information could be stored, either to reduce recomputation, or to equip the generators with priorities that permit them to explore rich veins first. Having said this, however, we see no easy mechanism to accomplish it.

A valuable type of information that we have not been able to collect is when a construction "nearly" succeeds. Constructions that lack one key ingredient suggest important directions for research. Our approach, however, makes this determination well nigh impossible. As researchers, we tend to ask three types of questions:

- Given certain parameters, what is the largest blocksize that can be made with current constructions?
- Given a new $ITD$ (or other object), what consequences does it have taken with known constructions and known ingredients?
- Given a construction, what does it make with known ingredients?

Our approach emphasizes the last question. The first can be answered in our system only by table look-up. The second question poses the worst problems, as we have seen. Emphasizing construction–based approaches appears to be sensible for making tables, but some real thought should be given to how to insert new ingredients efficiently.

Having read the first sections, the reader would be forgiven for having a feeling of *deja vu* in the latter sections. Indeed many of the constructions are quite similar, and this poses a problem with redundancy. Two generators need not make disjoint sets of objects, but of course it is more efficient if they do. Some of the inefficiencies that we have encountered relate to the difficulty of having each object generated by a single generator, insofar as that is possible.

Another type of redundancy occurs in the code. This happens especially where an object is created with a set of incompatible holes; a filler which "knew" how to fill a partial $TD$ with prescribed holes intersecting in prescribed ways would simplify a number of the Wilson–type generators. We tried adding such a filler routine, but efficiency considerations forced us to retreat.

The authorities ought to be easier to read by an expert in the field, and not require that one pore over code to see what happened. While that occurrence is fortunately rare, one feels that a more systematic mechanism for assigning authorities would eliminate the need to refer to a specific routine in the package.

Excepting efficiency, these disappointments are somewhat minor, given our objectives.

## 10.2 Some Concluding Remarks

Even such a lengthy discussion fails to convey all of the details, and that is as it must be. Rather than recapitulating earlier remarks, we close with some speculations.

Perhaps the most interesting speculation is what the true values of the number of MOLS are. We haven't succeeded in finding 15 MOLS of order 8360. Can it really be possible that they fail to exist? Frankly, we doubt it. (In fact, we would be surprised if that result weren't improved upon within a year.) But what is the truth here? 20? 200? 2000? 8000? We have no idea. It is diverting to imagine what would happen to combinatorial design theory if we suddenly constructed $\frac{t}{2}$ MOLS for every order $t > 6$. Making tables as we have done shouldn't be taken as evidence that the lower bounds obtained are in any way close to the truth; or that the known constructions are related even remotely to the ones that reveal something close to the truth. We hope that another breakthrough of the magnitude of Wilson's theorem does occur, but as to the form it might take, we make no guesses, educated or otherwise.

Another interesting speculation is whether it is feasible with current technology to build a "Design Theory Expert". More precisely, could we build one that manages within a finite amount of time to say something interesting? We would have to equip it with an enormous variety of constructions, and a huge library of data. That seems possible, although the effort involved would be daunting. Its mandate could be to find new designs with existing constructions; or it could be to find new constructions. People succeed at this (sometimes), partly because they acquire a sense of what of the current knowledge is relevant to the task at hand. Their knowledge may be incomplete, but nevertheless the right ingredients all hove in view at the same time, and presto! Managing to code what we mean by "interesting" and "relevant" is the real problem. We expect that systems with more limited expertise are more likely to be feasible. Our system falls short of being such an expert, primarily because it has no sense of when a construction is likely to succeed or likely to fail; and it doesn't learn about situations where it has wasted its time before, nor why. It does, however, have a substantial library of knowledge in the database and procedures upon which to build expertise and learning.

## Acknowledgments

Our primary indebtedness is to Andries Brouwer, for blazing the trail many years ago in [20], and for his participation in [114]. Too many people to mention here have provided useful pointers to the literature and comments; thanks to Julian Abel, Frank Bennett, Malcolm Greig, Esther Lamken, Ron Mullin, Paul Schellenberg, Doug Stinson, Yin Jian Xing, Hantao Zhang, Xiao Jun Zhu, and Zhu Lie. Julian Abel deserves special mention for his great contributions in checking the tables and improving the results.

## REFERENCES

[1] R.J.R. Abel, "Four mutually orthogonal latin squares of orders 28 and 52", *J. Combinat. Theory* A58 (1991), 306–309.

[2] R.J.R. Abel, private communications, 1993-94.

[3] R.J.R. Abel and Y.W. Cheng, "Some new MOLS of order $2^n p$ for $p$ a prime power", *Austral. J. Combin.* 10 (1994), 175–186.

[4] R.J.R. Abel, C.J. Colbourn, J.X. Yin and H. Zhang, "Existence of transversal designs with block size five and any index $\lambda$", preprint.

[5] R.J.R. Abel and D.T. Todorov, "Four MOLS of order 20, 30, 38 and 44", *J. Combinat. Theory* A64 (1993), 144–148.

[6] R.J.R. Abel, X. Zhang and H. Zhang, "Three mutually orthogonal idempotent latin squares of orders 22 and 26", *J. Stat. Plan. Infer.*, to appear.

[7] R.D. Baker, "Whist tournaments", *Proc. Sixth Southeastern Conf. Combin., Graph Theory, Computing* (1975) 89–100.

[8] R.D. Baker, "An elliptic semiplane", *J. Combin. Theory* A25 (1978), 193–195.

[9] F.E. Bennett, "Pairwise balanced designs with prime power block sizes exceeding 7", *Ann. Disc. Math.* 34 (1987), 43–64.

[10] F.E. Bennett, C.J. Colbourn and L. Zhu, "Existence of certain types of three HMOLS", *Discrete Math.*, to appear.

[11] F.E. Bennett, K.T. Phelps, C.A. Rodger and L. Zhu, "Constructions of perfect Mendelsohn designs", *Discrete Math.* 103 (1992), 139–151.

[12] F.E. Bennett and L. Zhu, "Existence of HSOLSSOM($h^n$) where $h$ is even", preprint.

[13] T. Beth, D. Jungnickel and H. Lenz, *Design Theory*, Cambridge University Press, Cambridge, 1986.

[14] R.C. Bose, "On the applications of properties of Galois fields to the problem of construction of hypergraecolatin squares", *Sankhya* 3 (1938), 328–338.

[15] R.C. Bose, I.M. Chakravarti and D.E. Knuth, "On methods of constructing sets of mutually orthogonal latin squares using a computer", *Technometrics* 2 (1960), 507–516.

[16] R.C. Bose, I.M. Chakravarti and D.E. Knuth, "On methods of constructing sets of mutually orthogonal latin squares using a computer II", *Technometrics* 3 (1961), 111–118.

[17] R.C. Bose and S.S. Shrikhande, "On the construction of sets of mutually orthogonal latin squares and the falsity of a conjecture of Euler", *Trans. Amer. Math. Soc.* 95 (1960), 191–209.

[18] R.C. Bose, S.S. Shrikhande and E.T. Parker, "Further results on the construction of mutually orthogonal latin squares and the falsity of Euler's conjecture", *Canad. J. Math.* 12 (1960), 189–203.

[19] R.K. Brayton, D. Coppersmith and A.J. Hoffman, "Self–orthogonal latin squares of all orders $n \neq 2, 3, 6$", *Bull. Amer. Math. Soc.* 80 (1974), 116–118.

[20] A.E. Brouwer, "The number of mutually orthogonal latin squares — a table to order 10000", Research Report ZW 123/79, Math. Centrum, Amsterdam, 1979.

[21] A.E. Brouwer, "On the existence of 30 mutually orthogonal latin squares", Math. Centrum Report ZW77, Amsterdam, 1980.

[22] A.E. Brouwer, "A series of separable designs with application to pairwise orthogonal latin squares", *Europ. J. Combin.* 1 (1980), 39–41.

[23] A.E. Brouwer, "Four MOLS of order 10 with a hole of order 2", *J. Stat. Plan. Infer.* 10 (1984), 203–205.

[24] A.E. Brouwer, "Recursive constructions of mutually orthogonal latin squares", *Ann. Discrete Math.* 46 (1991), 149–168.

[25] A.E. Brouwer and G.H.J. van Rees, "More mutually orthogonal latin squares", *Discrete Math.* 39 (1982), 263–281.

[26] C.J. Colbourn, "Four MOLS of order 26", *J. Comb. Math. Comb. Comput.* 17 (1995), 147–148.

[27] C.J. Colbourn, "Some direct constructions for incomplete transversal designs", *J. Stat. Plan. Infer.*, to appear.

[28] C.J. Colbourn, "Construction techniques for mutually orthogonal latin squares", in: *Combinatorics Advances*, Kluwer, to appear.

[29] C.J. Colbourn, J.H. Dinitz and D.R. Stinson, "More thwarts in transversal designs", *Finite Fields Applic.*, to appear.

[30] C.J. Colbourn, J.H. Dinitz and M. Wojtas, "Thwarts in transversal designs", *Des. Codes Crypt.* 5 (1995), 189–197.

[31] C.J. Colbourn, J. Yin and L. Zhu, "Six MOLS of order 76", *J. Comb. Math. Comb. Comput.*, to appear.

[32] D.J. Crampin and A.J.W. Hilton, "On the spectra of certain types of latin squares", *J. Combin. Theory* A19 (1975), 84–94.

[33] R.H.F. Denniston, "Subplanes of the Hughes plane of order 9", *Proc. Cambridge Phil. Soc.* 64 (1968), 589–598.

[34] R.H.F. Denniston, "Some maximal arcs in finite projective planes", *J. Comb. Theory* A6 (1969), 317–319.

[35] M.J. de Resmini, "On the Dempwolff plane", in: *Finite Geometries and Combinatorial Designs*, Amer. Math. Soc., 1987, pp. 47–64.

[36] J.H. Dinitz and D.R. Stinson, "MOLS with holes", *Discrete Math.* 44 (1983), 145–154.

[37] D.A. Drake and H. Lenz, "Orthogonal latin squares with orthogonal subsquares", *Archiv der Math.* 34 (1980), 565–576.

[38] B. Du, "On incomplete transversal designs with block size five", *Util. Math.* 40 (1991), 272–282.

[39] B. Du, "On the existence of incomplete transversal designs with block size 5", *Discrete Math.* 135 (1994), 81–92.

[40] L. Euler, "Recherches sur une nouvelle espèce de quarrés magiques", *Verh. Zeeuw. Gen. Weten. Vlissengen* 9 (1782), 85–239.

[41] B. Ganter, R. Mathon and A. Rosa, "A complete census of (10,3,2) designs and of Mendelsohn triple systems of order ten. II. Mendelsohn triple systems with repeated blocks", *Congr. Numer.* 22 (1978), 181–204.

[42] M. Greig, "Designs from configurations in projective planes", unpublished, 1992.

[43] R. Guérin, "Aspects algebraiques de problème de Yamamoto", *C.R. Acad. Sci.* 256 (1963), 583–586.

[44] R. Guérin, "Sur une généralisation de la méthode de Yamamoto pour la construction de carrés latins orthogonaux", *C.R. Acad. Sci.* 256 (1963), 2097–2100.

[45] R. Guérin, "Existence et propriétés des carrés latin orthogonaux II", *Publ. Inst. Stat. Univ. Paris* 15 (1966), 215–293.

[46] H. Hanani, "On the number of orthogonal latin squares", *J. Combin. Theory* 8 (1970), 247–271.

[47] A.S. Hedayat and E. Seiden, "On the theory and application of sum composition of latin squares and orthogonal latin squares", *Pacific J. Math.* 54 (1974), 85–113.

[48] K.E. Heinrich, "Near–orthogonal latin squares", *Util. Math.* 12 (1977), 145–155.

[49] K.E. Heinrich and L. Zhu, "Existence of orthogonal latin squares with aligned subsquares", *Discrete Math.* 59 (1984), 241–248.

[50] K.E. Heinrich and L. Zhu, "Incomplete self–orthogonal latin squares", *J. Austral. Math. Soc.* A42 (1987), 365–384.

[51] J.W.P. Hirschfeld, *Projective Geometries over Finite Fields*, Oxford University Press, 1979.

[52] J.D. Horton, "Sub–latin squares and incomplete orthogonal arrays", *J. Combin. Theory* A16 (1974), 23–33.

[53] D.M. Johnson, A.L. Dulmage and N.S. Mendelsohn, "Orthomorphisms of groups of orthogonal latin squares", *Canad. J. Math.* 13 (1961), 356–372.

[54] T.P. Kirkman, "On the perfect $r$–partitions of $r^2 - r + 1$", *Transactions of the Historic Society of Lancashire and Cheshire* (1850), 127–142.

[55] E.R. Lamken, "The existence of orthogonal partitioned incomplete latin squares of type $t^n$", *Discrete Math.* 89 (1991), 231–251.

[56] J.H. van Lint, *Combinatorial Theory Seminar*, *Lect. Notes Math.* 832, Springer, 1974.

[57] H.F. MacNeish, "Euler squares", *Ann. Math. (NY)* 23 (1922), 221–227.

[58] H.B. Mann, "The construction of orthogonal latin squares", *Ann. Math. Statist.* 13 (1942), 418–423.

[59] R.A. Mathon, unpublished.

[60] W.H. Mills, "Some mutually orthogonal latin squares", *Proc. Eighth S.E. Conf. Combin. Graph Theory Computing*, 1977, pp. 473–487.

[61] R.C. Mullin, "Finite bases for some PBD-closed sets", *Discrete Math.* 77 (1989), 217–236.

[62] R.C. Mullin, P.J. Schellenberg, D.R. Stinson and S.A. Vanstone, "Some results on the existence of squares", *Ann. Discrete Math.* 6 (1980), 257–274.

[63] R.C. Mullin and D.R. Stinson, "Holey SOLSSOMs", *Util. Math.* 25 (1984), 159–169.

[64] R.C. Mullin and L. Zhu, "The spectrum of HSOLSSOM($h^n$) where $h$ is odd", *Util. Math.* 27 (1985), 157–168.

[65] A.V. Nazarok, "Five pairwise orthogonal latin squares of order 21", *Issled. oper. i ASU*, 1991, pp. 54–56.

[66] T.G. Ostrom and F.A. Sherk, "Finite projective planes with affine subplanes", *Can. Math. Bull.* 7 (1964), 549-560.

[67] E.T. Parker, "Construction of some sets of mutually orthogonal latin squares", *Proc. Amer. Math. Soc.* 10 (1959), 946–949.

[68] E.T. Parker, "Nonextendability conditions on mutually orthogonal latin squares", *Proc. Amer. Math. Soc.* 13 (1962), 219–221.

[69] C. Pellegrino and P. Lancelotti, "A construction of pairs and triples of $k$–incomplete orthogonal arrays", *Ann. Disc. Math.* 37 (1988), 251–256.

[70] L. Puccio and M.J. de Resmini, "Subplanes of the Hughes plane of order 25", *Arch. Math.* 49 (1987), 151–165.

[71] J.F. Rigby, "Affine subplanes of finite projective planes", *Can. J. Math.* 17 (1965), 977-1009.

[72] C.E. Roberts Jr., "Sets of mutually orthogonal latin squares with 'like subsquares' ", *J. Combinat. Theory* A61 (1992) 50–63.

[73] C.E. Roberts Jr., "Sets of mutually orthogonal latin squares with 'like subsquares' II", preprint.

[74] R. Roth and M. Peters, "Four pairwise orthogonal latin squares of order 24", *J. Combin. Theory* A44 (1987), 152–155.

[75] F. Ruiz and E. Seiden, "Some results on the construction of orthogonal latin squares by the method of sum composition", *J. Combin. Theory* A16 (1974), 230–240.

[76] P.J. Schellenberg, G.H.J. van Rees and S.A. Vanstone, "Four pairwise orthogonal latin squares of order 15", *Ars Combinat.* 6 (1978), 141–150.

[77] E. Seiden, "A method of construction of resolvable BIBD", *Sankhya* A25 (1963), 393–394.

[78] E. Seiden and C.J. Wu, "Construction of three mutually orthogonal latin squares by the method of sum composition", in: *Essays in Probability and Statistics*, Shinko Publishing, Tokyo, 1976.

[79] J. Singer, "A theorem in finite projective geometry and some applications to number theory", *Trans. Amer. Math. Soc.* 43 (1938), 377–385.

[80] D.A. Sprott, "A series of symmetrical group divisible incomplete block designs", *Ann. Math. Statist.* 30 (1959), 249–251.

[81] D.R. Stinson, "On the existence of 7 and 8 mutually orthogonal latin squares", *Ars Combin.* 6 (1978), 113–115.

[82] D.R. Stinson, "On the existence of 30 mutually orthogonal latin squares", *Ars Combin.* 7 (1979), 153–170.

[83] D.R. Stinson, "A generalization of Wilson's construction for mutually orthogonal latin squares", *Ars Combin.* 8 (1979), 95–105.

[84] D.R. Stinson, "The equivalence of certain incomplete transversal designs and frames", *Ars Combin.* 22 (1986), 81–87.

[85] D.R. Stinson and L. Zhu, "On sets of three MOLS with holes", *Discrete Math.* 54 (1985), 321–328.

[86] D.R. Stinson and L. Zhu, "On the existence of MOLS with equal–sized holes", *Aequat. Math.* 33 (1987), 96–105.

[87] D.R. Stinson and L. Zhu, "On the existence of three MOLS with equal–sized holes", *Austral. J. Combin.* 4 (1991), 33–47.

[88] K. Szajowski, "The number of orthogonal latin squares", *Applicationes Math.* 15 (1976), 85–102.

[89] G. Tarry, "Le problème de 36 officiers", *Ass. Franc. Av. Sci.* 29 (1900), 170–203.

[90] D.T. Todorov, "Three mutually orthogonal latin squares of order 14", *Ars Combinat.* 20 (1985), 45–48.

[91] D.T. Todorov, "Four mutually orthogonal latin squares of order 20", *Ars Combinat.* 27 (1989), 63–65.

[92] W.D. Wallis, "Three orthogonal latin squares", *Congr. Numer.* 42 (1984), 69–86.

[93] W.D. Wallis and L. Zhu, "Orthogonal latin squares with small subsquares", *Lect. Notes Math.* 1036 (1983), 398–409.

[94] S.P. Wang, *On self-orthogonal latin squares and partial transversals of latin squares*, Ph.D. thesis, Ohio State University, 1978.

[95] R.M. Wilson, "A few more squares", *Proc. Fifth Southeastern Conf. Combin. Graph Theory Computing* (1974), pp. 675–680.

[96] R.M. Wilson, "Concerning the number of mutually orthogonal latin squares", *Discrete Math.* 9 (1974), 181–198.

[97] R.M. Wilson, "Constructions and uses of pairwise balanced designs", in: *Combinatorics*, Math. Centrum Amsterdam Tracts 55 (1974), 18–41.

[98] M. Wojtas, "On seven mutually orthogonal latin squares", *Discrete Math.* 20 (1977), 193–201.

[99] M. Wojtas, "The construction of mutually orthogonal latin squares", Kommunikat 172, Inst. Math. Politechniki Wroclaw, 1978.

[100] M. Wojtas, "A note on mutually orthogonal latin squares", Kommunikat 236, Inst. Math. Politechniki Wroclaw, 1978.

[101] M. Wojtas, "New Wilson–type constructions of mutually orthogonal latin squares II", Inst. Math. Politechniki Wroclaw, 1979.

[102] M. Wojtas, "New Wilson–type constructions of mutually orthogonal latin squares", *Discrete Math.* 32 (1980), 191–199.

[103] M. Wojtas, "Some new matrices–minus–diagonal and MOLS", *Discrete Math.* 76 (1989), 291–292.

[104] M. Wojtas, "Five mutually orthogonal latin squares of orders 24 and 40", preprint.

[105] K. Yamamoto, "Generation principles of latin squares", *Bull. Inst. Internat. Stat.* 38 (1961), 73–76.

[106] X. Zhang and H. Zhang, "Three mutually orthogonal idempotent latin squares of order 18", preprint.

[107] L. Zhu, "A short disproof of Euler's conjecture concerning orthogonal latin squares", *Ars Combin.* 14 (1982), 47–55.

[108] L. Zhu, "Pairwise orthogonal latin squares with orthogonal small subsquares", Research Report CORR 83–19, University of Waterloo, 1983.

[109] L. Zhu, "Some results on orthogonal latin squares with orthogonal subsquares", *Util. Math.* 25 (1984), 241–248.

[110] L. Zhu, "Orthogonal latin squares with subsquares", *Discrete Math.* 48 (1984), 315–321.

[111] L. Zhu, "Six pairwise orthogonal latin squares of order 69", *J. Austral. Math. Soc.* A37 (1984), 1–3.

[112] L. Zhu, "Incomplete transversal designs with block size five", *Congr. Numer.* 69 (1989), 13–20.

The following references are sections of the *CRC Handbook of Combinatorial Designs*, (C.J. Colbourn and J.H. Dinitz, editors), CRC Press, Boca Raton FL, 1996.

[113] R.J.R. Abel, "Difference families".

[114] R.J.R. Abel, A.E. Brouwer, C.J. Colbourn and J.H. Dinitz, "Mutually orthogonal latin squares".

[115] R.J.R. Abel, C.J. Colbourn and J.H. Dinitz, "Incomplete MOLS".

[116] R.J.R. Abel and S.C. Furino, "Resolvable and near–resolvable designs".

[117] R.A. Mathon and A. Rosa, "Balanced incomplete block designs".