An algorithmic approach to Tran Van Trung’s basic recursive construction of $t$-designs

Oscar Lopez

Department of Mathematical Sciences
Florida Atlantic University
Boca Raton, Florida, U.S.A.

31st Cumberland Conference
May 19, 2019
From now on, $v$, $k$, $\lambda$, and $t$ will denote positive integers, such that $v \geq k \geq t$.

A $t-(v, k, \lambda)$ design, is a pair $D = (V, B)$, where $V$ is a $v$-set of points and $B$ is a collection of $k$-subsets of $V$ called blocks, such that every $t$-subset of $V$ is contained in exactly $\lambda$ blocks. The general term $t$-design is used to indicate any $t-(v, k, \lambda)$ design.

A $t-(v, k, \lambda)$ design, $D$, is said to be simple if it has no repeated blocks.
From now on, \( v, k, \lambda, \) and \( t \) will denote positive integers, such that \( v \geq k \geq t \).

A \( t-(v, k, \lambda) \) design, is a pair \( D = (V, B) \), where \( V \) is a \( v \)-set of points and \( B \) is a collection of \( k \)-subsets of \( V \) called blocks, such that every \( t \)-subset of \( V \) is contained in exactly \( \lambda \) blocks. The general term \( t \)-design is used to indicate any \( t-(v, k, \lambda) \) design.

A \( t-(v, k, \lambda) \) design, \( D \), is said to be simple if it has no repeated blocks.
From now on, $v$, $k$, $\lambda$, and $t$ will denote positive integers, such that $v \geq k \geq t$.

A $t-(v, k, \lambda)$ design, is a pair $D = (V, B)$, where $V$ is a $v$-set of points and $B$ is a collection of $k$-subsets of $V$ called blocks, such that every $t$-subset of $V$ is contained in exactly $\lambda$ blocks. The general term $t$-design is used to indicate any $t-(v, k, \lambda)$ design.

A $t-(v, k, \lambda)$ design, $D$, is said to be simple if it has no repeated blocks.
A famous example of a simple $2 - (15, 3, 1)$ design is the solution to the famous Kirkman’s Schoolgirls Problem:

$$V = \{1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f\}$$

<table>
<thead>
<tr>
<th></th>
<th>1 8 f</th>
<th>2 9 f</th>
<th>3 a f</th>
<th>4 b f</th>
<th>5 c f</th>
<th>6 d f</th>
<th>7 e f</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8</td>
<td>f</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>a</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>b</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>c</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>d</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>e</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>f</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**OAL (FAU)**

An algorithmic approach to Tran Van Trung’s 31st Cumberland Conference May 19, 2019
For fixed values of $t$, $v$ and $k$, we can construct a simple $t-(v, k, \lambda)$ design by taking all possible $k$-subsets of a $v$-set as its blocks. This $t$-design is called a complete or trivial design.

In a complete $t-(v, k, \lambda)$ design, $\lambda = \binom{v-t}{k-t}$. This value of $\lambda$ is the largest value for which a $t-(v, k, \lambda)$ design exists and it is denoted by $\lambda_{\text{max}}$.

**Theorem** [Teirlinck (1987)] Given positive integers $t$ and $v$, with $v > t + 1$ and $v \equiv t \pmod{(t + 1)!^{2t+1}}$, then a simple $t-(v, t+1, (t+1)!^{2t+1})$ design exists. That is, simple non-trivial $t$-designs exist for all values of $t$. 

![Image of page content](image-url)
For fixed values of $t$, $v$ and $k$, we can construct a simple $t-(v, k, \lambda)$ design by taking all possible $k$-subsets of a $v$-set as its blocks. This $t$-design is called a complete or trivial design.

In a complete $t-(v, k, \lambda)$ design, $\lambda = \binom{v-t}{k-t}$. This value of $\lambda$ is the largest value for which a $t-(v, k, \lambda)$ design exists and it is denoted by $\lambda_{max}$.

**Theorem** [Teirlinck (1987)] Given positive integers $t$ and $v$, with $v > t + 1$ and $v \equiv t \pmod{(t + 1)!^{2t+1}}$, then a simple $t-(v, t+1, (t+1)!^{2t+1})$ design exists. That is, simple non-trivial $t$-designs exist for all values of $t$. 
For fixed values of $t$, $v$ and $k$, we can construct a simple $t-(v, k, \lambda)$ design by taking all possible $k$-subsets of a $v$-set as its blocks. This $t$-design is called a complete or trivial design.

In a complete $t-(v, k, \lambda)$ design, $\lambda = \binom{v - t}{k - t}$. This value of $\lambda$ is the largest value for which a $t-(v, k, \lambda)$ design exists and it is denoted by $\lambda_{\text{max}}$.

**Theorem** [Teirlinck (1987)] Given positive integers $t$ and $v$, with $v > t + 1$ and $v \equiv t \pmod{(t + 1)!^{2t+1}}$, then a simple $t-(v, t + 1, (t + 1)!^{2t+1})$ design exists. That is, simple non-trivial $t$-designs exist for all values of $t$. 
Theorem  Let $\mathcal{D}$ be a $t-(v, k, \lambda)$ design. If $S$ is any $s$-subset of $V$, with $0 \leq s \leq t$, then the number of blocks containing $S$ is:

$$\lambda_s := \lambda \frac{\binom{v - s}{t - s} \binom{t - s}{k - s}}{\binom{t - s}{t - s}}$$

Corollary  If $\mathcal{D} = (V, B)$ is a $t-(v, k, \lambda)$ design and $s$ is a positive integer, with $s \leq t$, then $\mathcal{D}$ is also an $s-(v, k, \lambda_s)$ design. If $s = t - 1$, we will call it the *reduced design* of $\mathcal{D}$. 
Theorem  Let $\mathcal{D}$ be a $t-(v, k, \lambda)$ design. If $S$ is any $s$-subset of $V$, with $0 \leq s \leq t$, then the number of blocks containing $S$ is:

$$\lambda_s := \lambda \frac{\binom{v-s}{t-s}}{\binom{k-s}{t-s}} \binom{t-s}{k-s} \quad (1)$$

Corollary  If $\mathcal{D} = (V, \mathcal{B})$ is a $t-(v, k, \lambda)$ design and $s$ is a positive integer, with $s \leq t$, then $\mathcal{D}$ is also an $s-(v, k, \lambda_s)$ design. If $s = t - 1$, we will call it the reduced design of $\mathcal{D}$. 

OAL (FAU)  An algorithmic approach to Tran Van Trung’s 31st Cumberland Conference  May 19, 2019  5/35
A set of necessary divisibility conditions for the existence of a $t-(v, k, \lambda)$ design is that
\[
\lambda \binom{v - s}{t - s} \equiv 0 \pmod{\binom{k - s}{t - s}}, \quad \text{for } 0 \leq s < t \tag{2}
\]

A quadruple $t-(v, k, \lambda)$, with $\lambda \leq \lambda_{\text{max}}$, will be called admissible if the above-mentioned divisibility conditions are met.
A set of necessary divisibility conditions for the existence of a $t-(v, k, \lambda)$ design is that

$$\lambda \binom{v-s}{t-s} \equiv 0 \pmod{\binom{k-s}{t-s}}, \text{ for } 0 \leq s < t$$

(2)

A quadruple $t-(v, k, \lambda)$, with $\lambda \leq \lambda_{\text{max}}$, will be called **admissible** if the above-mentioned divisibility conditions are met.
The smallest positive integer value of $\lambda$ for which $t-(v, k, \lambda)$ is admissible is denoted by $\lambda_{\text{min}}$, more exactly $\lambda_{\text{min}}(t, k, v)$. Then, for any admissible $t-(v, k, \lambda)$, $\lambda_{\text{min}}$ must divide $\lambda$.

An admissible $t-(v, k, \lambda)$ is said to be realizable if a $t-(v, k, \lambda)$ design exists. If a $t-(v, k, \lambda)$ design is yet not known to exist, then $t-(v, k, \lambda)$ will be called questionable.
The smallest positive integer value of $\lambda$ for which $t-(v, k, \lambda)$ is admissible is denoted by $\lambda_{min}$, more exactly $\lambda_{min}(t, k, v)$. Then, for any admissible $t-(v, k, \lambda)$, $\lambda_{min}$ must divide $\lambda$.

An admissible $t-(v, k, \lambda)$ is said to be \textit{realizable} if a $t-(v, k, \lambda)$ design exists. If a $t-(v, k, \lambda)$ design is yet not known to exist, then $t-(v, k, \lambda)$ will be called \textit{questionable}.
**Theorem** [Keevash] For fixed $t$, $k$, and $\lambda$, there exist $v_0(k, t, \lambda)$ such that if $v > v_0(k, t, \lambda)$ satisfies the divisibility conditions (2), then a $t-(v, k, \lambda)$ design exists.

If a $t$-design $\mathcal{D}$ is realizable, we can construct new designs, with smaller values of $t$, from $\mathcal{D}$:

(i) reduced design,
(ii) derived design, and
(iii) residual design.

Additionally, other new $t$-designs, with the same value of $t$, can be constructed from $\mathcal{D}$:

(i) supplementary design, and
(ii) complementary design.
Theorem [Keevash] For fixed $t$, $k$, and $\lambda$, there exist $v_0(k, t, \lambda)$ such that if $v > v_0(k, t, \lambda)$ satisfies the divisibility conditions (2), then a $t$-$(v, k, \lambda)$ design exists.

If a $t$-design $\mathcal{D}$ is realizable, we can construct new designs, with smaller values of $t$, from $\mathcal{D}$:

(i) reduced design,
(ii) derived design, and
(iii) residual design.

Additionally, other new $t$-designs, with the same value of $t$, can be constructed from $\mathcal{D}$:

(i) supplementary design, and
(ii) complementary design.
Theorem [Keevash] For fixed $t$, $k$, and $\lambda$, there exist $v_0(k, t, \lambda)$ such that if $v > v_0(k, t, \lambda)$ satisfies the divisibility conditions (2), then a $t-(v, k, \lambda)$ design exists.

If a $t$-design $\mathcal{D}$ is realizable, we can construct new designs, with smaller values of $t$, from $\mathcal{D}$:

(i) reduced design,
(ii) derived design, and
(iii) residual design.

Additionally, other new $t$-designs, with the same value of $t$, can be constructed from $\mathcal{D}$:

(i) supplementary design, and
(ii) complementary design.
Theorem [Keevash] For fixed $t$, $k$, and $\lambda$, there exist $v_0(k, t, \lambda)$ such that if $v > v_0(k, t, \lambda)$ satisfies the divisibility conditions (2), then a $t-(v, k, \lambda)$ design exists.

If a $t$-design $\mathcal{D}$ is realizable, we can construct new designs, with smaller values of $t$, from $\mathcal{D}$:

(i) reduced design,
(ii) derived design, and
(iii) residual design.

Additionally, other new $t$-designs, with the same value of $t$, can be constructed from $\mathcal{D}$:

(i) supplementary design, and
(ii) complementary design.
Theorem [Keevash] For fixed $t$, $k$, and $\lambda$, there exist $v_0(k, t, \lambda)$ such that if $v > v_0(k, t, \lambda)$ satisfies the divisibility conditions (2), then a $t-(v, k, \lambda)$ design exists.

If a $t$-design $\mathcal{D}$ is realizable, we can construct new designs, with smaller values of $t$, from $\mathcal{D}$:

(i) reduced design,
(ii) derived design, and
(iii) residual design.

Additionally, other new $t$-designs, with the same value of $t$, can be constructed from $\mathcal{D}$:

(i) supplementary design, and
(ii) complementary design.
Theorem [Keevash] For fixed $t$, $k$, and $\lambda$, there exist $v_0(k, t, \lambda)$ such that if $v > v_0(k, t, \lambda)$ satisfies the divisibility conditions (2), then a $t$-$(v, k, \lambda)$ design exists.

If a $t$-design $D$ is realizable, we can construct new designs, with smaller values of $t$, from $D$:

(i) reduced design,
(ii) derived design, and
(iii) residual design.

Additionally, other new $t$-designs, with the same value of $t$, can be constructed from $D$:

(i) supplementary design, and
(ii) complementary design.
Theorem [Keevash] For fixed $t$, $k$, and $\lambda$, there exist $v_0(k, t, \lambda)$ such that if $v > v_0(k, t, \lambda)$ satisfies the divisibility conditions (2), then a $t-(v, k, \lambda)$ design exists.

If a $t$-design $\mathcal{D}$ is realizable, we can construct new designs, with smaller values of $t$, from $\mathcal{D}$:

(i) reduced design,
(ii) derived design, and
(iii) residual design.

Additionally, other new $t$-designs, with the same value of $t$, can be constructed from $\mathcal{D}$:

(i) supplementary design, and
(ii) complementary design.
**Theorem** [Keevash] For fixed $t$, $k$, and $\lambda$, there exist $v_0(k, t, \lambda)$ such that if $v > v_0(k, t, \lambda)$ satisfies the divisibility conditions (2), then a $t-(v, k, \lambda)$ design exists.

If a $t$-design $\mathcal{D}$ is realizable, we can construct new designs, with smaller values of $t$, from $\mathcal{D}$:

(i) reduced design,
(ii) derived design, and
(iii) residual design.

Additionally, other new $t$-designs, with the same value of $t$, can be constructed from $\mathcal{D}$:

(i) supplementary design, and
(ii) complementary design.
Notation

- Let $V$ be a $v$-set, and let $V = V_1 \cup V_2$ be a partition of $V$ (i.e. $V_1 \cap V_2 = \emptyset$) with $|V_1| = v_1$ and $|V_2| = v_2$.

- The parameter set $t - \left(v_2, j, \lambda_t^{(j)} \right)$ for a design will indicate that the point set of the design is $V_2$. A design defined on that point set will be denoted by $D = \left(V_2, B \right)$.

- For $i = 0, \ldots, t$, let $D_i = \left(V_1, B^{(i)} \right)$ be the complete $i - (v_1, i, 1)$ design. For $i = t + 1, \ldots, k$, let $D_i = \left(V_1, B^{(i)} \right)$ be a simple $t - \left(v_1, i, \lambda_t^{(i)} \right)$ design.
Tran Van Trung’s basic construction of $t$-designs

Notation

- Identically, for $i = 0, \ldots, t$, let $\overline{D}_i = \left( V_2, \overline{B}^{(i)} \right)$ be the complete $i-(v_2, i, 1)$ design. For $i = t + 1, \ldots, k$, let $\overline{D}_i = \left( V_2, \overline{B}^{(i)} \right)$ be a simple $t-(v_2, i, \lambda_t^{(i)})$ design.

- There are two degenerate cases that will arise commonly in the construction. One of them is the case $v = k = t = 0$, which gives an ”empty” design, denoted by $\emptyset$, but we use the convention that its number of blocks is 1 (since the unique block is the empty block).

- The other one is the case $v = k$, which gives a degenerate $k$-design having only 1 block containing all $v$ points.
Notation

- For the "undefined cases" $s > i$ and $t - s > k - i$, we will consider $\lambda_s^{(i)} = 0$ and $\lambda_t^{(k-i)} = 0$, respectively.

- $T(s, t-s)$ will denote a $t$-subset $T$ of $V$ with $|T \cap V_1| = s$ and $|T \cap V_2| = t - s$, for $s = 0, ..., t$. Clearly, any $t$-subset of $V$ is then a $T(s, t-s)$ set for some $s \in \{0, ..., t\}$.

- Let $\mathcal{A}$ be a finite set, and let $u \in \{0, 1\}$. We will use the notation $\mathcal{A} \times [u]$ as follows: $\mathcal{A} \times [0] = \emptyset$, and $\mathcal{A} \times [1] = \mathcal{A}$. 
Notation

For $i = 0, \ldots, k$, we define

$$B_{i,k-i} := \left\{ B = B_i \cup \overline{B}_{k-i} : B_i \in B^{(i)}, \overline{B}_{k-i} \in \overline{B}^{(k-i)} \right\}$$

Note that $B_{i,k-i}$ and $B_{j,k-j}$ are pairwise disjoint, for $i \neq j$ and $i, j = 0, \ldots, k$.

We define

$$B := B_{(0,k)} \times [u_0] \cup B_{(1,k-1)} \times [u_1] \cup \ldots \cup B_{(k-1,1)} \times [u_{k-1}] \cup B_{(k,0)} \times [u_k]$$

with $u_i \in \{0, 1\}$, for $i = 0, \ldots, k$. 
**Theorem**  [Tran Van Trung - 2017]  (The ”Basic Construction”)  With the previously defined notation, let

\[ L_{s,t-s} = \sum_{i=0}^{k} u_i \lambda_s^{(i)} \lambda_{t-s}^{(k-i)} \]  

for \( s = 0, \ldots, t \). If

\[ L_0,t = L_1,t-1 = L_2,t-2 = \ldots = L_{t-1,1} = L_{t,0} = \Lambda \]

for some positive integer \( \Lambda \), then \( D = (V, B) \) is a simple  \( t-(v, k, \Lambda) \) design.
Major approaches to the problem of discovering new $t$-designs rely on

(i) the construction of large sets of $t$-designs,
(ii) using prescribed automorphism groups,
(iii) recursive construction methods.

This construction is of purely combinatorial nature and it requires finding solutions for the indices of smaller designs, which we call ingredient designs, that satisfy the above-mentioned equalities.
In order to take advantage of this procedure, we need to have at hand a complete and updated database of realizable simple t-designs.

We constructed a bank of realizable simple t-designs using the information from the "Handbook of Combinatorial Designs", and the database of t-designs published by Betten, Haberberger, Laue and Wassermann at http://www.algorithm.uni-bayreuth.de/en/research/discreta/disc_database.html

We used this bank as a source of ingredient designs. The bank contains 212,253 realizable non-trivial simple t-designs.
Our interest lies in constructing new $t$-designs using this procedure. For specific values of $t$, $v$, and $k$, we would be able to construct a $t-(v, k, \Lambda)$ design using this method if for some value of $v_1$, with $v_1 \in \{1, 2, \ldots, v - 1\}$, the following nonlinear system of $t + 1$ equations on $(k + 1)(2t + 3)$ integer unknowns has “realizable” solutions on the the $u_i$, $\lambda_s^{(i)}$ and $\overline{\lambda}_{t-s}^{(k-i)}$, for $s = 0, 1, \ldots, t$ and $i = 0, 1, \ldots, k$:

$$
\sum_{i=0}^{k} u_i \lambda_s^{(i)} \overline{\lambda}_{t-s}^{(k-i)} = \Lambda, s \in \{0, 1, \ldots, t\}
$$
For \( s = 0, 1, \ldots, t \) and \( i = 0, 1, \ldots, k \), and with the exception of the "undefined cases", we have from equation (1) that

\[
\lambda_s^{(i)} = \lambda^{(i)} \frac{v_1 - s}{t - s} \frac{t - s}{i - s} \quad \text{and} \quad \overline{\lambda}^{(k-i)}_{t-s} = \overline{\lambda}^{(k-i)} \frac{v_2 - t + s}{s} \frac{s}{i - t + s}.
\]

We know that \( \lambda^{(i)} = \lambda^{(i)}_{\text{min}} m^{(i)} \) and \( \overline{\lambda}^{(k-i)} = \overline{\lambda}^{(k-i)}_{\text{min}} \overline{m}^{(k-i)} \), for some \( m^{(i)} \in \{1, \ldots, m^{(i)}_{\text{max}}\} \) and \( \overline{m}^{(k-i)} \in \{1, \ldots, \overline{m}^{(k-i)}_{\text{max}}\} \), where

\[
m^{(i)}_{\text{max}} = \frac{\lambda^{(i)}_{\text{max}}}{\lambda^{(i)}_{\text{min}}} \quad \text{and} \quad \overline{m}^{(k-i)}_{\text{max}} = \frac{\overline{\lambda}^{(k-i)}_{\text{max}}}{\overline{\lambda}^{(k-i)}_{\text{min}}}.
\]
Our approach

For \( s = 0, 1, \ldots, t \) and \( i = 0, 1, \ldots, k \), and with the exception of the "undefined cases" defined previously, we rename

\[
\begin{align*}
c_s(i) &= \lambda_{\min}^{(i)} \left( \frac{v_1 - s}{t - s} \right) \left( \frac{t - s}{i - s} \right) \quad \text{and} \quad c_{(k-i)}^{(i)} = \lambda_{\min}^{(k-i)} \left( \frac{v_2 - t + s}{s} \right) \left( \frac{s}{i - t + s} \right).
\end{align*}
\]

Then, we can rewrite

\[
\lambda_s^{(i)} = c_s^{(i)} m^{(i)} \quad \text{and} \quad \lambda_{(k-i)}^{(k-i)} = c_{(k-i)}^{(i)} m^{(k-i)}.
\]

We consider a "new" value for the parameter \( \lambda \) in the following way: \( \lambda^{(i)} = 0 \) and \( \overline{\lambda}^{(k-i)} = 0 \) denote that \( B_{(i,k-i)} \) is not being used in the construction.
For $i = 0, 1, ..., k$, we then define the variable $x^{(i)} = m^{(i)} \overline{m}^{(k-i)}$. Then, $x^{(i)} \in \left\{1, ..., m_{\text{max}}^{(i)}\right\} \cdot \left\{1, ..., \overline{m}_{\text{max}}^{(k-i)}\right\}$, where $x^{(i)} = 0$ means that $\mathcal{B}_{(i,k-i)}$ is not being used in the construction.

Let’s define $R^{(i)} = \left\{m^{(i)} \in \left\{1, ..., m_{\text{max}}^{(i)}\right\} : \text{there is a realizable} \right. ~ t - \left( v_1, i, \lambda_{\text{min}}^{(i)} m^{(i)} \right) \text{ design} \},$ and \[ R^{(k-i)} = \left\{\overline{m}^{(k-i)} \in \left\{1, ..., \overline{m}_{\text{max}}^{(k-i)}\right\} : \text{there is a realizable} \right. ~ t - \left( v_2, k - i, \lambda_{\text{min}}^{(k-i)} \overline{m}^{(k-i)} \right) \text{ design} \}, \text{ for } i = 0, 1, ..., k. \]
In order to construct a $t-\left( v, k, \Lambda \right)$ design we must use only ingredient designs that are realizable, so the value of $x^{(i)}$ will have to be restricted to the set $\{0\} \cup R^{(i)} \cdot \overline{R}^{(k-i)}$. The problem of constructing a $t-\left( v, k, \Lambda \right)$ design using Tran Van Trung’s Theorem is now reduced to finding a solution of the linear system

$$A \cdot x = M\Lambda_{min} \cdot j$$

(4)

where $A = \begin{pmatrix} c^{(i)}_s \overline{c}_{t-s} \end{pmatrix}_{0 \leq s \leq t, 0 \leq i \leq k}$ is of dimension $(t + 1) \times (k + 1)$, $x = \left( x^{(0)}, x^{(1)}, \ldots, x^{(k)} \right)$ with $x^{(i)} \in \{0\} \cup R^{(i)} \cdot \overline{R}^{(k-i)}$ for $i = 0, 1, \ldots, k$, $M \in \{1, 2, \ldots, M_{max}\}$, $M_{max} = \frac{\Lambda_{max}}{\Lambda_{min}}$, and $j$ is the $(t + 1)$-dimensional vector of all ones.
Since all the entries of the matrix $A$ in the linear system (4) are rational numbers, multiplying the $i$th row of $A$ by an appropriate integer $n_i$, for $1 \leq i \leq t + 1$, will transform $A$ into a matrix with integer entries. Let's rename this new matrix as $A'$ and the new vector on the right side of the matrix equation as $Mb$, where $b$ is the column vector obtained by multiplying the entry $i$ of $\Lambda_{min} j$ by $n_i$. Then, our focus turns to solving the following linear system

$$A' \cdot x = Mb$$  (5)
An **integral lattice** is a discrete additive subgroup of $\mathbb{Z}^n$.

The domain space of equations (5) is then a bounded integral lattice that, unfortunately, will almost always contain a significant number of “holes” (missing points), which increases notably the difficulty of finding such solutions.
To add insult to injury, we are interested in looking for solutions of those equations for every value of $M$ between 1 and $M_{\text{max}}$.
Using "brute force" to search for new realizable $t$-$(v, k, \Lambda)$ designs using this method results to be computationally intractable. For instance, let's say that we want to construct $4 - (37, 11, \Lambda)$ designs with $v_1 = 23$ and $v_2 = 14$, using this method.

The number of points in the domain of Equations (4) is then:

$$2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 4 \cdot 2 \cdot 7 \cdot 20 \cdot 4 \cdot 58 \cdot 2 \cdot 970 \cdot 2 \cdot 1293 \cdot 2 \cdot 647 \cdot 2 \cdot 643 \cdot 2 \cdot 6723 \cdot 2 = 7.4669729008197 \times 10^{24}$$
Let $c$ be an $n$-dimensional real coefficient vector, let $A$ be an $m \times n$ real coefficient matrix, and let $b$ be an $m$-dimensional real vector. The problem

$$\min \quad c^T \cdot x$$

subject to

$$A \cdot x = b$$

for

$$x \geq 0$$

and

$$x \in \mathbb{Z}^n$$

is called an **Integer Linear Programming (ILP) problem**. The function $c^T \cdot x$ is the **objective function**. The matrix $A$ is the **constraint coefficient matrix**, and $b$ is the **vector of constraint limits**. The set of the solutions of $A \cdot x = b$, for $x \geq 0$ and $x \in \mathbb{Z}^n$, is called the **feasible region**. If some of the entries of $x$ are not integers, then we say this is a **Mixed Integer Linear Programming (MILP) problem**.
An ILP problem is considered NP-complete. However, there are ILP solvers available that work very efficiently for problems of the size we are considering. We particularly directed our attention to one ILP solver called GUROBI.

The GUROBI Optimizer is a commercial optimization solver for mathematical programming problems. GUROBI generally solves ILP and MILP problems using a linear-programming based branch-and-bound algorithm. As an additional feature, GUROBI has the ability to directly solve ILP problems with piecewise-linear (PWL) objective functions. Each piecewise-linear objective function is associated with a model variable. For each variable, a PWL function will be defined by its breakpoints.
We define
\[
f : \prod_{i=0}^{k} \{0, 1, \ldots, m_{\text{max}}^{(i)} \cdot m_{\text{max}}^{(k-i)}\} \to \mathbb{Z}^+ \cup \{0\} \tag{6}
\]
by
\[
f = \sum_{i=0}^{k} f_i \tag{7}
\]
where for \(i = 0, 1, \ldots, k\), \(f_i\) is defined in the following way:
\[
f_i (x^{(i)}) = \begin{cases} 0, & x^{(i)} \in \{0\} \cup R^{(i)} \cdot \overline{R}^{(k-i)} \\ 1, & \text{otherwise} \end{cases} \tag{8}
\]
For $i = 0, 1, \ldots, k$, if we think of $f_i$ as the function of a real variable comprised of line segments joining the points

$$(x^{(i)}, f_i(x^{(i)})),$$

for $x^{(i)} \in \left\{0, 1, \ldots, m_{\text{max}}^{(i)} \cdot \overline{m}_{\text{max}}^{(k-i)}\right\}$, then $f_i$ would be a PWL objective function of a real variable. Hence, $f$ is the sum of the PWL objective functions associated to each variable of the problem.
Now, the problem

\[
\begin{align*}
\min & \quad f \\
\text{subject to} & \quad A' \cdot x = M b \\
\text{and} & \quad x \in \prod_{i=0}^{k} \left\{ 0, 1, \ldots, m_{\max}^{(i)} \cdot \overline{m}_{\max}^{(k-i)} \right\}
\end{align*}
\]  

becomes an ILP and can be solved through GUROBI.
The ILP approach

- If the minimum value for $f$ is 0, then we have a vector $x$ solution of the original system

$$A \cdot x = M\Lambda_{min}j$$

with $x^{(i)} \in \{0\} \cup R^{(i)} \cdot \overline{R}^{(k-i)}$ for $i = 0, 1, ..., k$, and through a fast search we can find the appropriate $m^{(i)}$ and $\overline{m}^{(k-i)}$ from the equation $x^{(i)} = m^{(i)} \overline{m}^{(k-i)}$.

- If the minimum value of $f$ is greater than 0 or the feasible region is empty, then the original system does not have a solution using the realizable ingredient designs from our database. Thus the $t-(\nu, k, \Lambda)$ design cannot be constructed using Tran Van Trung’s Theorem.
We focused our construction on values for $t \in \{3, 4, 5, 6, 7\}$, $v \leq 52$, and $\Lambda < 1000000$, and only on questionable $t-(v, k, \Lambda)$ designs.

We performed our construction starting from the highest value of $t$ we selected. In this way, every time we constructed new $t$-designs, we could generate some other new ones with lower values of $t$, through reduced, derived and residual designs.

We designed our algorithm to be able to run in parallel through different cores. For fixed values of $t, k$ and $v$, we would start looking for solutions of equations (5) from $M = 1$ to $M_{\text{max}}$, skipping those values of $M$ for which a $t-(v, k, M\Lambda_{\text{min}})$ design had already been constructed.
Even though we are still performing our constructions, we have been able to create 327722 new simple non-trivial $t$-designs, more than doubling the original number of simple non-trivial $t$-designs.
We are still running our algorithm to construct new $t$-$(v, k, \Lambda)$ designs. This list will continue growing.

We are thinking of how we could construct new $t$-designs using "ingredient" designs with lower values of $t$, by extending Tran Van Trung approach.

Tran Van Trung published a second recursive technique of this kind that uses resolutions. We are working on implementing an efficient algorithm to construct more new $t$-designs under this second approach.
Questions?
Thank you