Generalized Covering and Packing Designs

Robert Bailey
University of Regina

Joint work with Andrea Burgess (MUN), Michael Cavers (UofC),
Karen Meagher (UofR)

CMS Winter Meeting
4th December 2010

Generalized t-designs

- In a 2009 paper, Peter Cameron obtained a common generalization of various classes of combinatorial design, including t-designs and orthogonal arrays (among many others).

Generalized t-designs

- In a 2009 paper, Peter Cameron obtained a common generalization of various classes of combinatorial design, including t-designs and orthogonal arrays (among many others).
- Several people are now working on these "generalized t-designs".

Generalized t-designs

- In a 2009 paper, Peter Cameron obtained a common generalization of various classes of combinatorial design, including t-designs and orthogonal arrays (among many others).
- Several people are now working on these "generalized t-designs".
- Question: Is there a similar generalization for covering designs and covering arrays?

Generalized t-designs

- In a 2009 paper, Peter Cameron obtained a common generalization of various classes of combinatorial design, including t-designs and orthogonal arrays (among many others).
- Several people are now working on these "generalized t-designs".
- Question: Is there a similar generalization for covering designs and covering arrays?
- Same question, but for packing designs and packing arrays?

Generalized t-designs

- In a 2009 paper, Peter Cameron obtained a common generalization of various classes of combinatorial design, including t-designs and orthogonal arrays (among many others).
- Several people are now working on these "generalized t-designs".
- Question: Is there a similar generalization for covering designs and covering arrays?
- Same question, but for packing designs and packing arrays?
- Answer: Yes.....

Notation

- Suppose v, k, t, λ, m are positive integers, where $v \geq k \geq t$.

Notation

- Suppose v, k, t, λ, m are positive integers, where $v \geq k \geq t$.
- Let $\mathbf{v} = (v_1, v_2, \ldots, v_m)$ be an m-tuple of positive integers with sum v, and let $\mathbf{k} = (k_1, k_2, \ldots, k_m)$ be an m-tuple of positive integers with sum k, and where each $k_i \leq v_i$.

Notation

- Suppose v, k, t, λ, m are positive integers, where $v \geq k \geq t$.
- Let $\mathbf{v} = (v_1, v_2, \ldots, v_m)$ be an m-tuple of positive integers with sum v, and let $\mathbf{k} = (k_1, k_2, \ldots, k_m)$ be an m-tuple of positive integers with sum k, and where each $k_i \leq v_i$.
- Now let $\mathbf{X} = (X_1, X_2, \ldots, X_m)$ be an m-tuple of pairwise disjoint sets, where $|X_i| = v_i$.

Notation

- Suppose v, k, t, λ, m are positive integers, where $v \geq k \geq t$.
- Let $\mathbf{v} = (v_1, v_2, \ldots, v_m)$ be an m-tuple of positive integers with sum v, and let $\mathbf{k} = (k_1, k_2, \ldots, k_m)$ be an m-tuple of positive integers with sum k, and where each $k_i \leq v_i$.
- Now let $\mathbf{X} = (X_1, X_2, \ldots, X_m)$ be an m-tuple of pairwise disjoint sets, where $|X_i| = v_i$.
- An m-tuple $\mathbf{t} = (t_1, t_2, \ldots, t_m)$ of *non-negative* integers is called *admissible*, if they sum to t and each $t_i \leq k_i$.

Notation

- Suppose v, k, t, λ, m are positive integers, where $v \geq k \geq t$.
- Let $\mathbf{v} = (v_1, v_2, \ldots, v_m)$ be an m-tuple of positive integers with sum v, and let $\mathbf{k} = (k_1, k_2, \ldots, k_m)$ be an m-tuple of positive integers with sum k, and where each $k_i \leq v_i$.
- Now let $\mathbf{X} = (X_1, X_2, \ldots, X_m)$ be an m-tuple of pairwise disjoint sets, where $|X_i| = v_i$.
- An m-tuple $\mathbf{t} = (t_1, t_2, \ldots, t_m)$ of *non-negative* integers is called *admissible*, if they sum to t and each $t_i \leq k_i$.
- Similarly, an m-tuple $\mathbf{T} = (T_1, T_2, \ldots, T_m)$ of disjoint sets is called *admissible*, if each $T_i \subseteq X_i$ and $|T_i| = t_i$, where $\mathbf{t} = (t_1, t_2, \ldots, t_m)$ is admissible.

Definition

- A *generalized* t-$(\mathbf{v},\mathbf{k},\lambda)$ *design* is a collection of *blocks*

  $$\mathbf{B} = (B_1, B_2, \ldots, B_m),$$

  where:

Definition

- A *generalized* t-$(\mathbf{v}, \mathbf{k}, \lambda)$ *design* is a collection of *blocks*

  $$\mathbf{B} = (B_1, B_2, \ldots, B_m),$$

  where:

  - each $B_i \subseteq X_i$, and $|B_i| = k_i$;

Definition

- A *generalized* t-$(\mathbf{v}, \mathbf{k}, \lambda)$ *design* is a collection of *blocks*

  $$\mathbf{B} = (B_1, B_2, \ldots, B_m),$$

  where:
  - each $B_i \subseteq X_i$, and $|B_i| = k_i$;
  - every admissible $\mathbf{T} = (T_1, T_2, \ldots, T_m)$ is a contained in exactly λ blocks.

Motivating examples

- If $\mathbf{v} = (v)$ and $\mathbf{k} = (k)$, we obtain an (ordinary) t-(v, k, λ) design.

Motivating examples

- If $\mathbf{v} = (v)$ and $\mathbf{k} = (k)$, we obtain an (ordinary) t-(v, k, λ) design.
- If $\mathbf{v} = (s, s, \ldots, s)$ and $\mathbf{k} = (1, 1, \ldots, 1)$ (both vectors of length k), we obtain an orthogonal array, $OA_\lambda(k, s, t)$.

Motivating examples

- If $\mathbf{v} = (v)$ and $\mathbf{k} = (k)$, we obtain an (ordinary) t-(v, k, λ) design.
- If $\mathbf{v} = (s, s, \ldots, s)$ and $\mathbf{k} = (1, 1, \ldots, 1)$ (both vectors of length k), we obtain an orthogonal array, $OA_\lambda(k, s, t)$.
- Other classes of designs that can be obtained include 1-factorizations of complete graphs (where $\mathbf{k} = (2, 1)$, $t = 2$, $\lambda = 1$) and resolvable designs (where $\mathbf{k} = (k - 1, 1)$).

Motivating examples

- If $\mathbf{v} = (v)$ and $\mathbf{k} = (k)$, we obtain an (ordinary) t-(v, k, λ) design.
- If $\mathbf{v} = (s, s, \ldots, s)$ and $\mathbf{k} = (1, 1, \ldots, 1)$ (both vectors of length k), we obtain an orthogonal array, $OA_\lambda(k, s, t)$.
- Other classes of designs that can be obtained include 1-factorizations of complete graphs (where $\mathbf{k} = (2, 1)$, $t = 2$, $\lambda = 1$) and resolvable designs (where $\mathbf{k} = (k - 1, 1)$).
- How can we similarly generalize covering designs/arrays or packing designs/arrays?

Generalized covering designs

- A *generalized covering design* $GC_\lambda(\mathbf{v}, \mathbf{k}, t)$ is a collection of blocks

  $$\mathbf{B} = (B_1, B_2, \ldots, B_m),$$

  where:

Generalized covering designs

- A *generalized covering design* $GC_\lambda(\mathbf{v}, \mathbf{k}, t)$ is a collection of blocks
$$\mathbf{B} = (B_1, B_2, \ldots, B_m),$$
where:
    - each $B_i \subseteq X_i$, and $|B_i| = k_i$;

Generalized covering designs

- A *generalized covering design* $GC_\lambda(\mathbf{v}, \mathbf{k}, t)$ is a collection of blocks

  $$\mathbf{B} = (B_1, B_2, \ldots, B_m),$$

  where:

  - each $B_i \subseteq X_i$, and $|B_i| = k_i$;
  - every admissible $\mathbf{T} = (T_1, T_2, \ldots, T_m)$ is a subset of **at least** λ blocks.

Generalized covering designs

- A *generalized covering design* $GC_{\lambda}(\mathbf{v}, \mathbf{k}, t)$ is a collection of blocks

  $$\mathbf{B} = (B_1, B_2, \ldots, B_m),$$

  where:

    - each $B_i \subseteq X_i$, and $|B_i| = k_i$;
    - every admissible $\mathbf{T} = (T_1, T_2, \ldots, T_m)$ is a subset of **at least** λ blocks.

- Usually, we are only concerned with the case $\lambda = 1$, and omit the subscript λ.

Generalized covering designs

- A *generalized covering design* $GC_\lambda(\mathbf{v}, \mathbf{k}, t)$ is a collection of blocks
$$\mathbf{B} = (B_1, B_2, \ldots, B_m),$$
where:
  - each $B_i \subseteq X_i$, and $|B_i| = k_i$;
  - every admissible $\mathbf{T} = (T_1, T_2, \ldots, T_m)$ is a subset of **at least** λ blocks.
- Usually, we are only concerned with the case $\lambda = 1$, and omit the subscript λ.
- By taking $\mathbf{v} = (v)$ and $\mathbf{k} = (k)$ we obtain (ordinary) covering designs, and by taking $\mathbf{v} = (s, s, \ldots, s)$ and $\mathbf{k} = (1, 1, \ldots, 1)$ we obtain covering arrays.

Generalized packing designs

- A *generalized packing design* $GP_\lambda(\mathbf{v}, \mathbf{k}, t)$ is a collection of blocks

  $$\mathbf{B} = (B_1, B_2, \ldots, B_m),$$

  where:
  - each $B_i \subseteq X_i$, and $|B_i| = k_i$;
  - every admissible $\mathbf{T} = (T_1, T_2, \ldots, T_m)$ is a subset of **at most** λ blocks.
- Usually, we are only concerned with the case $\lambda = 1$, and omit the subscript λ.
- By taking $\mathbf{v} = (v)$ and $\mathbf{k} = (k)$ we obtain (ordinary) packing designs, and by taking $\mathbf{v} = (s, s, \ldots, s)$ and $\mathbf{k} = (1, 1, \ldots, 1)$ we obtain packing arrays.

Aren't these the same thing?

- Here, the similarity between coverings and packings ends and the two diverge.

Aren't these the same thing?

- Here, the similarity between coverings and packings ends and the two diverge.
- For generalized covering designs, our work has been on general construction methods.

Aren't these the same thing?

- Here, the similarity between coverings and packings ends and the two diverge.
- For generalized covering designs, our work has been on general construction methods.
- See Bailey, Burgess, Cavers, Meagher, *Generalized covering designs and clique coverings*, preprint: `arXiv.org/abs/1011.3804`

Aren't these the same thing?

- Here, the similarity between coverings and packings ends and the two diverge.
- For generalized covering designs, our work has been on general construction methods.
- See Bailey, Burgess, Cavers, Meagher, *Generalized covering designs and clique coverings*, preprint: `arXiv.org/abs/1011.3804`
- For packings, we have been working on a census of optimal packings for small k and t. Many well-known classes of designs arise!

Aren't these the same thing?

- Here, the similarity between coverings and packings ends and the two diverge.
- For generalized covering designs, our work has been on general construction methods.
- See Bailey, Burgess, Cavers, Meagher, *Generalized covering designs and clique coverings*, preprint: `arXiv.org/abs/1011.3804`
- For packings, we have been working on a census of optimal packings for small k and t. Many well-known classes of designs arise!
- This is still ongoing.....

Clique coverings

- Where $t = 2$, we can describe generalized covering designs in terms of graphs.

Clique coverings

- Where $t = 2$, we can describe generalized covering designs in terms of graphs.
- For vectors $\mathbf{v} = (v_1, v_2, \ldots, v_m)$ and $\mathbf{k} = (k_1, k_2, \ldots, k_m)$, construct a graph $G = H_1 + H_2 + \cdots + H_m$, where

$$H_i = \begin{cases} \overline{K_{v_i}}, & \text{if } k_i = 1, \\ K_{v_i}, & \text{if } k_i \geq 2. \end{cases}$$

Clique coverings

- Where $t = 2$, we can describe generalized covering designs in terms of graphs.
- For vectors $\mathbf{v} = (v_1, v_2, \ldots, v_m)$ and $\mathbf{k} = (k_1, k_2, \ldots, k_m)$, construct a graph $G = H_1 + H_2 + \cdots + H_m$, where

$$H_i = \begin{cases} \overline{K_{v_i}}, & \text{if } k_i = 1, \\ K_{v_i}, & \text{if } k_i \geq 2. \end{cases}$$

- Then a $GC(\mathbf{v}, \mathbf{k}, 2)$ corresponds to a *clique covering* of G by copies of a complete graph K_k, where each clique contains k_i vertices of H_i.

A construction algorithm

- If, for each i, we have $k_i \geq 2$, there is an algorithm to construct a $GC(\mathbf{v}, \mathbf{k}, 2)$ starting from known (ordinary) covering designs:

A construction algorithm

- If, for each i, we have $k_i \geq 2$, there is an algorithm to construct a $GC(\mathbf{v}, \mathbf{k}, 2)$ starting from known (ordinary) covering designs:
  - Start with a $(v_{\max}, k_{\min}, 2)$ covering design $\mathcal{C}$.

A construction algorithm

- If, for each i, we have $k_i \geq 2$, there is an algorithm to construct a $GC(\mathbf{v}, \mathbf{k}, 2)$ starting from known (ordinary) covering designs:
  - Start with a $(v_{\max}, k_{\min}, 2)$ covering design $\mathcal{C}$.
  - Put a copy of $\mathcal{C}$ on each part.

A construction algorithm

- If, for each i, we have $k_i \geq 2$, there is an algorithm to construct a $GC(\mathbf{v}, \mathbf{k}, 2)$ starting from known (ordinary) covering designs:
  - Start with a $(v_{\max}, k_{\min}, 2)$ covering design $\mathcal{C}$.
  - Put a copy of $\mathcal{C}$ on each part.
  - In any part where $v_i < v_{\max}$, delete the extra points, replacing them with a “placeholder” symbol $\star$.

A construction algorithm

- If, for each i, we have $k_i \geq 2$, there is an algorithm to construct a $GC(\mathbf{v}, \mathbf{k}, 2)$ starting from known (ordinary) covering designs:
  - Start with a $(v_{\max}, k_{\min}, 2)$ covering design $\mathcal{C}$.
  - Put a copy of $\mathcal{C}$ on each part.
  - In any part where $v_i < v_{\max}$, delete the extra points, replacing them with a "placeholder" symbol $\star$.
  - In any part where $k_i > k_{\min}$, add $k_i - k_{\min}$ placeholders to each block.

A construction algorithm

- If, for each i, we have $k_i \geq 2$, there is an algorithm to construct a $GC(\mathbf{v}, \mathbf{k}, 2)$ starting from known (ordinary) covering designs:
    - Start with a $(v_{\max}, k_{\min}, 2)$ covering design $\mathcal{C}$.
    - Put a copy of $\mathcal{C}$ on each part.
    - In any part where $v_i < v_{\max}$, delete the extra points, replacing them with a "placeholder" symbol $\star$.
    - In any part where $k_i > k_{\min}$, add $k_i - k_{\min}$ placeholders to each block.
    - In each block, replace the placeholders greedily, ensuring that no symbol is repeated in a block.

A construction algorithm

- If, for each i, we have $k_i \geq 2$, there is an algorithm to construct a $GC(\mathbf{v}, \mathbf{k}, 2)$ starting from known (ordinary) covering designs:
  - Start with a $(v_{\max}, k_{\min}, 2)$ covering design $\mathcal{C}$.
  - Put a copy of $\mathcal{C}$ on each part.
  - In any part where $v_i < v_{\max}$, delete the extra points, replacing them with a "placeholder" symbol $\star$.
  - In any part where $k_i > k_{\min}$, add $k_i - k_{\min}$ placeholders to each block.
  - In each block, replace the placeholders greedily, ensuring that no symbol is repeated in a block.
  - Remove any repeated blocks.

A construction algorithm

- If, for each i, we have $k_i \geq 2$, there is an algorithm to construct a $GC(\mathbf{v}, \mathbf{k}, 2)$ starting from known (ordinary) covering designs:
    - Start with a $(v_{\max}, k_{\min}, 2)$ covering design $\mathcal{C}$.
    - Put a copy of $\mathcal{C}$ on each part.
    - In any part where $v_i < v_{\max}$, delete the extra points, replacing them with a "placeholder" symbol $\star$.
    - In any part where $k_i > k_{\min}$, add $k_i - k_{\min}$ placeholders to each block.
    - In each block, replace the placeholders greedily, ensuring that no symbol is repeated in a block.
    - Remove any repeated blocks.
- If $\mathcal{C}$ is optimal, and there is an index i where $v_i = v_{\max}$ and $k_i = k_{\min}$, then we are guaranteed our $GC(\mathbf{v}, \mathbf{k}, 2)$ is also optimal.

Example

- Suppose $\mathbf{v} = (5, 6, 7)$, $\mathbf{k} = (3, 4, 3)$ and $t = 2$; then $v_{\max} = 7$ and $k_{\min} = 3$.

Example

- Suppose $\mathbf{v} = (5, 6, 7)$, $\mathbf{k} = (3, 4, 3)$ and $t = 2$; then $v_{\max} = 7$ and $k_{\min} = 3$.
- Start with a $(7, 3, 2)$ covering design $\mathcal{C}$:

$$\begin{gathered} \{124\} \\ \{235\} \\ \{346\} \\ \{457\} \\ \{156\} \\ \{267\} \\ \{137\} \end{gathered}$$

Example, II

- Suppose $\mathbf{v} = (5, 6, 7)$, $\mathbf{k} = (3, 4, 3)$ and $t = 2$; then $v_{\max} = 7$ and $k_{\min} = 3$.
- Put a copy of $\mathcal{C}$ on each part:

$$\begin{array}{lll}
(\{124\}, & \{124\}, & \{124\}) \\
(\{235\}, & \{235\}, & \{235\}) \\
(\{346\}, & \{346\}, & \{346\}) \\
(\{457\}, & \{457\}, & \{457\}) \\
(\{156\}, & \{156\}, & \{156\}) \\
(\{267\}, & \{267\}, & \{267\}) \\
(\{137\}, & \{137\}, & \{137\})
\end{array}$$

Example, III

- Suppose $\mathbf{v} = (5, 6, 7)$, $\mathbf{k} = (3, 4, 3)$ and $t = 2$; then $v_{\max} = 7$ and $k_{\min} = 3$.
- In any part where $v_i < v_{\max}$, delete the extra points, replacing them with a "placeholder" symbol $\star$:

$$\begin{array}{lll}
(\{124\}, & \{124\}, & \{124\}) \\
(\{235\}, & \{235\}, & \{235\}) \\
(\{34\star\}, & \{346\}, & \{346\}) \\
(\{45\star\}, & \{45\star\}, & \{457\}) \\
(\{15\star\}, & \{156\}, & \{156\}) \\
(\{2\star\star\}, & \{26\star\}, & \{267\}) \\
(\{13\star\}, & \{13\star\}, & \{137\})
\end{array}$$

Example, IV

- Suppose $\mathbf{v} = (5, 6, 7)$, $\mathbf{k} = (3, 4, 3)$ and $t = 2$; then $v_{\max} = 7$ and $k_{\min} = 3$.
- In any part where $k_i > k_{\min}$, add $k_i - k_{\min}$ placeholders to each block:

$$\begin{array}{lll}
(\{124\}, & \{124\star\}, & \{124\}) \\
(\{235\}, & \{235\star\}, & \{235\}) \\
(\{34\star\}, & \{346\star\}, & \{346\}) \\
(\{45\star\}, & \{45\star\star\}, & \{457\}) \\
(\{15\star\}, & \{156\star\}, & \{156\}) \\
(\{2\star\star\}, & \{26\star\star\}, & \{267\}) \\
(\{13\star\}, & \{13\star\star\}, & \{137\})
\end{array}$$

Example, V

- Suppose $\mathbf{v} = (5, 6, 7)$, $\mathbf{k} = (3, 4, 3)$ and $t = 2$; then $v_{\max} = 7$ and $k_{\min} = 3$.
- Replace the placeholders greedily, ensuring that no symbol is repeated in a block:

$$\begin{array}{lll}
(\{124\}, & \{1234\}, & \{124\}) \\
(\{235\}, & \{1235\}, & \{235\}) \\
(\{134\}, & \{1346\}, & \{346\}) \\
(\{145\}, & \{1245\}, & \{457\}) \\
(\{125\}, & \{1256\}, & \{156\}) \\
(\{123\}, & \{1236\}, & \{267\}) \\
(\{123\}, & \{1234\}, & \{137\})
\end{array}$$

Optimal packings: $k = 3$, $t = 2$

- For $k = 3$ and $t = 2$, optimal generalized packing designs $GP(\mathbf{v}, \mathbf{k}, 2)$ can be found as follows:

Optimal packings: $k = 3$, $t = 2$

- For $k = 3$ and $t = 2$, optimal generalized packing designs $GP(\mathbf{v}, \mathbf{k}, 2)$ can be found as follows:
  - $\mathbf{k} = (3)$: maximum packings of triples (well-known)

Optimal packings: $k = 3$, $t = 2$

- For $k = 3$ and $t = 2$, optimal generalized packing designs $GP(\mathbf{v}, \mathbf{k}, 2)$ can be found as follows:
  - $\mathbf{k} = (3)$: maximum packings of triples (well-known)
  - $\mathbf{k} = (2, 1)$: (partial) (near) 1-factorizations of complete graphs

Optimal packings: $k = 3$, $t = 2$

- For $k = 3$ and $t = 2$, optimal generalized packing designs $GP(\mathbf{v}, \mathbf{k}, 2)$ can be found as follows:
    - $\mathbf{k} = (3)$: maximum packings of triples (well-known)
    - $\mathbf{k} = (2, 1)$: (partial) (near) 1-factorizations of complete graphs
    - $\mathbf{k} = (1, 1, 1)$: Latin rectangles

Optimal packings: $k = 4$, $t = 2$

- For $k = 4$ and $t = 2$, optimal generalized packing designs $\mathrm{GP}(\mathbf{v}, \mathbf{k}, 2)$ can be found as follows:

Optimal packings: $k = 4$, $t = 2$

- For $k = 4$ and $t = 2$, optimal generalized packing designs $\mathrm{GP}(\mathbf{v}, \mathbf{k}, 2)$ can be found as follows:
  - $\mathbf{k} = (4)$: maximum packings of quadruples

Optimal packings: $k = 4$, $t = 2$

- For $k = 4$ and $t = 2$, optimal generalized packing designs $\mathrm{GP}(\mathbf{v}, \mathbf{k}, 2)$ can be found as follows:
    - $\mathbf{k} = (4)$: maximum packings of quadruples
    - $\mathbf{k} = (3, 1)$: depending on congruence classes mod 6, we have *Kirkman triple systems*, *Hanani triple systems*, *strong incomplete Kirkman signal sets*, and designs derived from them

Optimal packings: $k = 4$, $t = 2$

- For $k = 4$ and $t = 2$, optimal generalized packing designs $\mathrm{GP}(\mathbf{v}, \mathbf{k}, 2)$ can be found as follows:
    - $\mathbf{k} = (4)$: maximum packings of quadruples
    - $\mathbf{k} = (3, 1)$: depending on congruence classes mod 6, we have *Kirkman triple systems*, *Hanani triple systems*, *strong incomplete Kirkman signal sets*, and designs derived from them
    - $\mathbf{k} = (2, 2)$: ??????

Optimal packings: $k = 4$, $t = 2$

- For $k = 4$ and $t = 2$, optimal generalized packing designs $\mathrm{GP}(\mathbf{v}, \mathbf{k}, 2)$ can be found as follows:
    - $\mathbf{k} = (4)$: maximum packings of quadruples
    - $\mathbf{k} = (3, 1)$: depending on congruence classes mod 6, we have *Kirkman triple systems*, *Hanani triple systems*, *strong incomplete Kirkman signal sets*, and designs derived from them
    - $\mathbf{k} = (2, 2)$: ??????
    - $\mathbf{k} = (2, 1, 1)$: Howell designs, Room squares, SOMAs, ...

Optimal packings: $k = 4$, $t = 2$

- For $k = 4$ and $t = 2$, optimal generalized packing designs $\mathrm{GP}(\mathbf{v}, \mathbf{k}, 2)$ can be found as follows:
    - $\mathbf{k} = (4)$: maximum packings of quadruples
    - $\mathbf{k} = (3, 1)$: depending on congruence classes mod 6, we have *Kirkman triple systems*, *Hanani triple systems*, *strong incomplete Kirkman signal sets*, and designs derived from them
    - $\mathbf{k} = (2, 2)$: ??????
    - $\mathbf{k} = (2, 1, 1)$: Howell designs, Room squares, SOMAs, ...
    - $\mathbf{k} = (1, 1, 1)$: mutually orthogonal Latin rectangles

THE END

Preprint: Bailey, Burgess, Cavers, Meagher, *Generalized covering designs and clique coverings*, `arXiv.org/abs/1011.3804`