Generalized Covering Designs

Robert Bailey
University of Regina

Ottawa–Carleton Discrete Mathematics Days
14–15 May 2010

Covering designs

- Let v, k, t, λ be positive integers with $v \geq k \geq t$.

Covering designs

- Let v, k, t, λ be positive integers with $v \geq k \geq t$.
- A $(v, k, t)_\lambda$ *covering design* is a family $\mathcal{C}$ of k-subsets of a v-set X, where any t-subset of X is contained in at least λ members of $\mathcal{C}$.

Covering designs

- Let v, k, t, λ be positive integers with $v \geq k \geq t$.
- A $(v, k, t)_\lambda$ *covering design* is a family $\mathcal{C}$ of k-subsets of a v-set X, where any t-subset of X is contained in at least λ members of $\mathcal{C}$.
- This is weakening of the notion of a t-*design*, requiring only "at least λ" blocks rather than "exactly λ".

Covering designs

- Let v, k, t, λ be positive integers with $v \geq k \geq t$.
- A $(v, k, t)_\lambda$ *covering design* is a family $\mathcal{C}$ of k-subsets of a v-set X, where any t-subset of X is contained in at least λ members of $\mathcal{C}$.
- This is weakening of the notion of a t-*design*, requiring only "at least λ" blocks rather than "exactly λ".
- Usually, we are only concerned with the case $\lambda = 1$, and omit the subscript λ.

Covering designs

- Let v, k, t, λ be positive integers with $v \geq k \geq t$.
- A $(v, k, t)_\lambda$ *covering design* is a family $\mathcal{C}$ of k-subsets of a v-set X, where any t-subset of X is contained in at least λ members of $\mathcal{C}$.
- This is weakening of the notion of a t-*design*, requiring only "at least λ" blocks rather than "exactly λ".
- Usually, we are only concerned with the case $\lambda = 1$, and omit the subscript λ.
- The size of the smallest possible (v, k, t) covering design is denoted by $C(v, k, t)$.

Covering designs: an example

- An $(8, 5, 2)$ covering design:

$$\begin{array}{c} 1\,2\,3\,4\,5 \\ 1\,5\,6\,7\,8 \\ 2\,3\,6\,7\,8 \\ 4\,5\,6\,7\,8 \end{array}$$

Covering designs: an example

- An $(8, 5, 2)$ covering design:

$$\begin{array}{c} 1\,2\,3\,4\,5 \\ 1\,5\,6\,7\,8 \\ 2\,3\,6\,7\,8 \\ 4\,5\,6\,7\,8 \end{array}$$

- Each pair chosen from $\{1, \ldots, 8\}$ is contained in at least one of the 5-sets given.

Covering arrays

- Let N, t, k, v, λ be positive integers.

Covering arrays

- Let N, t, k, v, λ be positive integers.
- A *covering array* $CA_\lambda(N; k, s, t)$ is an $N \times k$ array, where:
  - the entries are from an alphabet of size s;
  - in every set of t columns, each t-tuple occurs in at least λ rows.

Covering arrays

- Let N, t, k, v, λ be positive integers.
- A *covering array* $CA_\lambda(N; k, s, t)$ is an $N \times k$ array, where:
  - the entries are from an alphabet of size s;
  - in every set of t columns, each t-tuple occurs in at least λ rows.
- This is weakening of the notion of an *orthogonal array*, requiring only “at least λ” rows rather than “exactly λ”.

Covering arrays

- Let N, t, k, v, λ be positive integers.
- A *covering array* $CA_\lambda(N; k, s, t)$ is an $N \times k$ array, where:
  - the entries are from an alphabet of size s;
  - in every set of t columns, each t-tuple occurs in at least λ rows.
- This is weakening of the notion of an *orthogonal array*, requiring only "at least λ" rows rather than "exactly λ".
- Usually, we are only concerned with the case $\lambda = 1$, and omit the subscript λ.

Covering arrays: an example

- A covering array $CA(5;4,2,2)$:

$$\begin{array}{cccc}
0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 \\
1 & 1 & 0 & 1 \\
1 & 0 & 1 & 1 \\
0 & 1 & 1 & 1
\end{array}$$

Covering arrays: an example

- A covering array $CA(5;4,2,2)$:

$$\begin{array}{cccc}
0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 \\
1 & 1 & 0 & 1 \\
1 & 0 & 1 & 1 \\
0 & 1 & 1 & 1
\end{array}$$

- In each pair of columns, each of the 2^2 possible combinations $00, 01, 10, 11$ appears in at least one row.

A common generalization?

- 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).

A common generalization?

- 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".

A common generalization?

- 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?

A common generalization?

- 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?
- Answer: Yes.....

A common generalization: notation

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

A common generalization: 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 $k_i \leq v_i$.

A common generalization: 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 $|X_i| = v_i$.

A common generalization: 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 $t_i \leq k_i$.

A common generalization: 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.

A common generalization: definition

- 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:

A common generalization: definition

- 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 $|B_i| = k_i$;

A common generalization: definition

- 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.

A common generalization: definition

- 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.

A common generalization: definition

- 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.
- Cameron's generalized t-designs are defined similarly, but strengthened to require "exactly λ" blocks rather than "at least λ".

A common generalization: motivating examples

- A (v, k, t) covering design is a $GC(\mathbf{v}, \mathbf{k}, t)$, where $\mathbf{v} = (v)$ and $\mathbf{k} = (k)$.

A common generalization: motivating examples

- A (v, k, t) covering design is a $GC(\mathbf{v}, \mathbf{k}, t)$, where $\mathbf{v} = (v)$ and $\mathbf{k} = (k)$.
- A covering array $CA(N; k, s, t)$ is a $GC(\mathbf{v}, \mathbf{k}, t)$, where $\mathbf{v} = (s, s, \ldots, s)$ and $\mathbf{k} = (1, 1, \ldots, 1)$ (both vectors of length k), with N blocks.

A common generalization: motivating examples

- A (v,k,t) covering design is a $GC(\mathbf{v},\mathbf{k},t)$, where $\mathbf{v}=(v)$ and $\mathbf{k}=(k)$.
- A covering array $CA(N;k,s,t)$ is a $GC(\mathbf{v},\mathbf{k},t)$, where $\mathbf{v}=(s,s,\ldots,s)$ and $\mathbf{k}=(1,1,\ldots,1)$ (both vectors of length k), with N blocks.
- So we do indeed have a common generalization!

A common generalization: motivating examples

- A (v, k, t) covering design is a $GC(\mathbf{v}, \mathbf{k}, t)$, where $\mathbf{v} = (v)$ and $\mathbf{k} = (k)$.
- A covering array $CA(N; k, s, t)$ is a $GC(\mathbf{v}, \mathbf{k}, t)$, where $\mathbf{v} = (s, s, \ldots, s)$ and $\mathbf{k} = (1, 1, \ldots, 1)$ (both vectors of length k), with N blocks.
- So we do indeed have a common generalization!
- Both t-designs and orthogonal arrays appear as motivating examples for Cameron's generalized t-designs in the same way.

A common generalization: another example

- Suppose $\mathbf{v} = (5, 6, 7)$, $\mathbf{k} = (3, 4, 3)$ and $t = 2$.

A common generalization: another example

- Suppose $\mathbf{v} = (5, 6, 7)$, $\mathbf{k} = (3, 4, 3)$ and $t = 2$.
- The following is a $GC(\mathbf{v}, \mathbf{k}, 2)$:

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

Clique coverings

- In the case where $t = 2$, we can describe generalized covering designs in terms of graphs.

Clique coverings

- In the case 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

- In the case 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.

Bounds

- The main question in the study of coverings is often to bound the smallest possible size of a design.

Bounds

- The main question in the study of coverings is often to bound the smallest possible size of a design.
- Borrowing the notation from (ordinary) covering designs, we denote the smallest possible size of a $GC(\mathbf{v}, \mathbf{k}, t)$ by $C(\mathbf{v}, \mathbf{k}, t)$.

Bounds

- The main question in the study of coverings is often to bound the smallest possible size of a design.
- Borrowing the notation from (ordinary) covering designs, we denote the smallest possible size of a $GC(\mathbf{v}, \mathbf{k}, t)$ by $C(\mathbf{v}, \mathbf{k}, t)$.
- A naïve bound can be obtained as follows:

$$C(\mathbf{v}, \mathbf{k}, 2) \geq \left\lceil \frac{|E(G)|}{|E(K_k)|} \right\rceil = \left\lceil \frac{\binom{v}{2} - \sum_{k_i=1} \binom{v_i}{2}}{\binom{k}{2}} \right\rceil.$$

Restriction

- If $I \subseteq \{1, \ldots, m\}$, the *restriction* of $\mathbf{v}$ to I, denoted $\mathbf{v}^I$, consists of those entries of $\mathbf{v}$ indexed by I.

Restriction

- If $I \subseteq \{1, \ldots, m\}$, the *restriction* of $\mathbf{v}$ to I, denoted $\mathbf{v}^I$, consists of those entries of $\mathbf{v}$ indexed by I.
- Likewise, we can define $\mathbf{k}^I$.

Restriction

- If $I \subseteq \{1, \ldots, m\}$, the *restriction* of $\mathbf{v}$ to I, denoted $\mathbf{v}^I$, consists of those entries of $\mathbf{v}$ indexed by I.
- Likewise, we can define $\mathbf{k}^I$.
- It is easy to see that

$$C(\mathbf{v}, \mathbf{k}, 2) \geq C(\mathbf{v}^I, \mathbf{k}^I, 2)$$

for any set of indices I.

Restriction

- If $I \subseteq \{1, \ldots, m\}$, the *restriction* of $\mathbf{v}$ to I, denoted $\mathbf{v}^I$, consists of those entries of $\mathbf{v}$ indexed by I.
- Likewise, we can define $\mathbf{k}^I$.
- It is easy to see that

  $$C(\mathbf{v}, \mathbf{k}, 2) \geq C(\mathbf{v}^I, \mathbf{k}^I, 2)$$

  for any set of indices I.
- If each $k_i \geq 2$, this gives us a bound in terms of "ordinary" covering numbers:

  $$C(\mathbf{v}, \mathbf{k}, 2) \geq \max\{C(v_i, k_i, 2)\}.$$

Restriction

- If $I \subseteq \{1, \ldots, m\}$, the *restriction* of $\mathbf{v}$ to I, denoted $\mathbf{v}^I$, consists of those entries of $\mathbf{v}$ indexed by I.
- Likewise, we can define $\mathbf{k}^I$.
- It is easy to see that

  $$C(\mathbf{v}, \mathbf{k}, 2) \geq C(\mathbf{v}^I, \mathbf{k}^I, 2)$$

  for any set of indices I.
- If each $k_i \geq 2$, this gives us a bound in terms of "ordinary" covering numbers:

  $$C(\mathbf{v}, \mathbf{k}, 2) \geq \max\{C(v_i, k_i, 2)\}.$$

- This allows us to use known values and bounds for "ordinary" covering designs to obtain bounds for generalized covering designs.

Equivalence

- We can define an equivalence relation $\sim$ on the indices, by $i \sim j$ iff $v_i = v_j$ and $k_i = k_j \geq 2$.

Equivalence

- We can define an equivalence relation $\sim$ on the indices, by $i \sim j$ iff $v_i = v_j$ and $k_i = k_j \geq 2$.
- If R is a set of equivalence class representatives, then we have

$$C(\mathbf{v}, \mathbf{k}, 2) = C(\mathbf{v}^R, \mathbf{k}^R, 2).$$

Equivalence

- We can define an equivalence relation $\sim$ on the indices, by $i \sim j$ iff $v_i = v_j$ and $k_i = k_j \geq 2$.
- If R is a set of equivalence class representatives, then we have

$$C(\mathbf{v}, \mathbf{k}, 2) = C(\mathbf{v}^R, \mathbf{k}^R, 2).$$

- This works because we can simply "repeat" equivalent parts without needing more blocks.

Equivalence

- We can define an equivalence relation $\sim$ on the indices, by $i \sim j$ iff $v_i = v_j$ and $k_i = k_j \geq 2$.
- If R is a set of equivalence class representatives, then we have
$$C(\mathbf{v}, \mathbf{k}, 2) = C(\mathbf{v}^R, \mathbf{k}^R, 2).$$
- This works because we can simply "repeat" equivalent parts without needing more blocks.
- So it suffices (if each $k_i \geq 2$) to consider equivalence classes under $\sim$.

Point deletion and block expansion

- Within a part, we can reduce v_i and obtain a $GC(\mathbf{v}', \mathbf{k}, 2)$ with the same number of blocks. This operation is called *point deletion*.

Point deletion and block expansion

- Within a part, we can reduce v_i and obtain a $GC(\mathbf{v}', \mathbf{k}, 2)$ with the same number of blocks. This operation is called *point deletion*.
- Also, we can increase k_i and obtain a $GC(\mathbf{v}, \mathbf{k}', 2)$ with the same number of blocks. This operation is called *block expansion*.

Point deletion and block expansion

- Within a part, we can reduce v_i and obtain a $GC(\mathbf{v}', \mathbf{k}, 2)$ with the same number of blocks. This operation is called *point deletion*.
- Also, we can increase k_i and obtain a $GC(\mathbf{v}, \mathbf{k}', 2)$ with the same number of blocks. This operation is called *block expansion*.
- Assuming each $k_i \geq 2$, if $v_{\max}$ is the largest v_i and $k_{\min}$ the smallest k_i, then repeating these operations (and using equivalence) gives the bound

$$C(\mathbf{v}, \mathbf{k}, 2) \leq C(v_{\max}, k_{\min}, 2).$$

Point deletion and block expansion

- Within a part, we can reduce v_i and obtain a $GC(\mathbf{v}', \mathbf{k}, 2)$ with the same number of blocks. This operation is called *point deletion*.
- Also, we can increase k_i and obtain a $GC(\mathbf{v}, \mathbf{k}', 2)$ with the same number of blocks. This operation is called *block expansion*.
- Assuming each $k_i \geq 2$, if $v_{\max}$ is the largest v_i and $k_{\min}$ the smallest k_i, then repeating these operations (and using equivalence) gives the bound

$$C(\mathbf{v}, \mathbf{k}, 2) \leq C(v_{\max}, k_{\min}, 2).$$

- Furthermore, if there is an index i where $v_i = v_{\max}$ and $k_i = k_{\min}$, restricting to that part gives an exact result:

$$C(\mathbf{v}, \mathbf{k}, 2) = C(v_{\max}, k_{\min}, 2).$$

An algorithm

- The ideas on the last few slides are all constructive, and (if each $k_i \geq 2$) they give rise to an algorithm to construct a $GC(\mathbf{v}, \mathbf{k}, 2)$ starting from known (ordinary) covering designs.

An algorithm

- The ideas on the last few slides are all constructive, and (if each $k_i \geq 2$) they give rise to 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 $\mathcal{C}$.

An algorithm

- The ideas on the last few slides are all constructive, and (if each $k_i \geq 2$) they give rise to 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.

An algorithm

- The ideas on the last few slides are all constructive, and (if each $k_i \geq 2$) they give rise to 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 $\star$.

An algorithm

- The ideas on the last few slides are all constructive, and (if each $k_i \geq 2$) they give rise to 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.

An algorithm

- The ideas on the last few slides are all constructive, and (if each $k_i \geq 2$) they give rise to 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.

An algorithm

- The ideas on the last few slides are all constructive, and (if each $k_i \geq 2$) they give rise to 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.

An algorithm

- The ideas on the last few slides are all constructive, and (if each $k_i \geq 2$) they give rise to 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.

That example again

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

That example again

- 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 $\mathcal{C}$:

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

That example again

- 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:

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

That example again

- 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 $\star$:

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

That example again

- 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:

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

That example again

- 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:

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

THE END