Finding good designs for experiments

R. A. Bailey
University of St Andrews
QMUL (emerita)
Queen Mary University of London

Worcester Polytechnic Institute,
Colloquium in Mathematical Sciences,
22 March 2019

Abstract

I have N experimental units available for an experiment to compare v treatments. The experimental units may be all alike, or they make be partitioned into blocks, or there may be rows and columns.

Abstract

I have N experimental units available for an experiment to compare v treatments. The experimental units may be all alike, or they make be partitioned into blocks,
or there may be rows and columns.

The design is the function allocating treatments to units.
It is optimal if it minimizes the average value of the variance of the estimator of the difference between two treatments.

Abstract

I have N experimental units available for an experiment to compare v treatments. The experimental units may be all alike, or they make be partitioned into blocks,
or there may be rows and columns.

The design is the function allocating treatments to units.
It is optimal if it minimizes the average value of the variance of the estimator of the difference between two treatments.

How do I find an optimal design for my given situation?

1. Find a design that satisfies the conditions of a theorem guaranteeing optimality?

Abstract

I have N experimental units available for an experiment to compare v treatments. The experimental units may be all alike, or they make be partitioned into blocks,
or there may be rows and columns.

The design is the function allocating treatments to units.
It is optimal if it minimizes the average value of the variance of the estimator of the difference between two treatments.

How do I find an optimal design for my given situation?

1. Find a design that satisfies the conditions of a theorem guaranteeing optimality?
2. Use folklore that suggests that being close to those conditions implies being close to optimal?

Abstract

I have N experimental units available for an experiment to compare v treatments. The experimental units may be all alike, or they make be partitioned into blocks,
or there may be rows and columns.

The design is the function allocating treatments to units.
It is optimal if it minimizes the average value of the variance of the estimator of the difference between two treatments.

How do I find an optimal design for my given situation?

1. Find a design that satisfies the conditions of a theorem guaranteeing optimality?
2. Use folklore that suggests that being close to those conditions implies being close to optimal?
3. Use patterns, to find a design that is regular in the sense of looking the same from the viewpoint of each treatment?

Abstract

I have N experimental units available for an experiment to compare v treatments. The experimental units may be all alike, or they make be partitioned into blocks,
or there may be rows and columns.

The design is the function allocating treatments to units.
It is optimal if it minimizes the average value of the variance of the estimator of the difference between two treatments.

How do I find an optimal design for my given situation?

1. Find a design that satisfies the conditions of a theorem guaranteeing optimality?
2. Use folklore that suggests that being close to those conditions implies being close to optimal?
3. Use patterns, to find a design that is regular in the sense of looking the same from the viewpoint of each treatment?
4. Use symmetries, to find a design with a large group of automorphsims?

Abstract

I have N experimental units available for an experiment to compare v treatments. The experimental units may be all alike, or they make be partitioned into blocks,
or there may be rows and columns.

The design is the function allocating treatments to units.
It is optimal if it minimizes the average value of the variance of the estimator of the difference between two treatments.

How do I find an optimal design for my given situation?

1. Find a design that satisfies the conditions of a theorem guaranteeing optimality?
2. Use folklore that suggests that being close to those conditions implies being close to optimal?
3. Use patterns, to find a design that is regular in the sense of looking the same from the viewpoint of each treatment?
4. Use symmetries, to find a design with a large group of automorphsims?
5. Computer search?

The set-up: treatments and experimental units

I have v treatments that I want to compare.
I have N experimental units that I can use, where $N > v$.
One treatment can be applied to each.

Case 1. The experimental units are all alike.

The set-up: treatments and experimental units

I have v treatments that I want to compare.
I have N experimental units that I can use, where $N > v$.
One treatment can be applied to each.

Case 1. The experimental units are all alike.

Case 2. The experimental units are divided into b blocks, with k experimental units in each block (so $N = bk$).
We expect there to be inherent differences between blocks.

The set-up: treatments and experimental units

I have v treatments that I want to compare.
I have N experimental units that I can use, where $N > v$.
One treatment can be applied to each.

Case 1. The experimental units are all alike.

Case 2. The experimental units are divided into b blocks, with k experimental units in each block (so $N = bk$).
We expect there to be inherent differences between blocks.

Case 3. There are two or more systems of blocks, such as rows and columns.

The set-up: treatments and experimental units

I have v treatments that I want to compare.
I have N experimental units that I can use, where $N > v$.
One treatment can be applied to each.

Case 1. The experimental units are all alike.

Case 2. The experimental units are divided into b blocks, with k experimental units in each block (so $N = bk$).
We expect there to be inherent differences between blocks.

Case 3. There are two or more systems of blocks, such as rows and columns.

The set-up: treatments and experimental units

I have v treatments that I want to compare.
I have N experimental units that I can use, where $N > v$.
One treatment can be applied to each.

Case 1. The experimental units are all alike.

Case 2. The experimental units are divided into b blocks, with k experimental units in each block (so $N = bk$).
We expect there to be inherent differences between blocks.

Case 3. There are two or more systems of blocks, such as rows and columns.

How should I choose a design?

Case 1

The experimental units are all alike.

Estimation and variance

We measure the response Y on each unit.

If that unit has treatment i then we assume that

$$Y = \tau_i + \text{random noise}.$$

We want to estimate all the simple differences $\tau_i - \tau_j$.

Put $V_{ij} =$ variance of the best linear unbiased estimator for $\tau_i - \tau_j$.

We want all the V_{ij} to be small.

The design is A-optimal if it minimizes $\sum_{i=1}^{v} \sum_{j=i+1}^{v} V_{ij}$.

How do we calculate variance?

The replication r_i of treatment i is its number of occurrences.
So one constraint is

$$\sum_{i=1}^{v} r_i = N.$$

Theorem

Assume that all the noise is independent, with variance σ^2. Then

$$V_{ij} = \left(\frac{1}{r_i} + \frac{1}{r_j}\right)\sigma^2.$$

How do we calculate variance?

The replication r_i of treatment i is its number of occurrences. So one constraint is

$$\sum_{i=1}^{v} r_i = N.$$

Theorem

Assume that all the noise is independent, with variance σ^2. Then

$$V_{ij} = \left(\frac{1}{r_i} + \frac{1}{r_j}\right)\sigma^2.$$

Put $\bar{V} =$ average value of the V_{ij}. Then $\bar{V} = \frac{2}{v}\left(\sum_{i=1}^{v}\frac{1}{r_i}\right)\sigma^2.$

How do we calculate variance?

The replication r_i of treatment i is its number of occurrences.
So one constraint is

$$\sum_{i=1}^{v} r_i = N.$$

Theorem

Assume that all the noise is independent, with variance σ^2. Then

$$V_{ij} = \left(\frac{1}{r_i} + \frac{1}{r_j}\right)\sigma^2.$$

Put $\bar{V} =$ average value of the V_{ij}. Then $\bar{V} = \frac{2}{v}\left(\sum_{i=1}^{v} \frac{1}{r_i}\right)\sigma^2$.

Theorem

$\bar{V}$ is minimized when the replications are as equal as possible, in the sense that no pair differ by more than 1.

Proof.

I set this to my undergraduates. □

Case 2

The experimental units are divided into b blocks of k units each.

Model when there are blocks

We measure the response Y on each unit in each block.

If that unit has treatment i and block m, then we assume that

$$Y = \tau_i + \beta_m + \text{random noise}.$$

To get rid of the β parameters, we look at $(I - P)Y$,
where P is the $N \times N$ matrix of orthogonal projection onto the space spanned by the characteristic vectors of the blocks.

Model when there are blocks

We measure the response Y on each unit in each block.

If that unit has treatment i and block m, then we assume that

$$Y = \tau_i + \beta_m + \text{random noise.}$$

To get rid of the β parameters, we look at $(I - P)Y$, where P is the $N \times N$ matrix of orthogonal projection onto the space spanned by the characteristic vectors of the blocks.

Let X be the $N \times v$ incidence matrix of treatments in experimental units.
The information matrix is $X^\top(I - P)X$.

Design $\to$ graph

If $i \neq j$, the concurrence λ_{ij} of treatments i and j is the number of occurrences of the pair $\{i, j\}$ in blocks, counted according to multiplicity.

Design $\rightarrow$ graph

If $i \neq j$, the concurrence λ_{ij} of treatments i and j is the number of occurrences of the pair $\{i, j\}$ in blocks, counted according to multiplicity.

The concurrence graph G of the design has the treatments as vertices.
There are no loops.
If $i \neq j$ then there are λ_{ij} edges between i and j.
So the valency d_i of vertex i is

$$d_i = \sum_{j \neq i} \lambda_{ij}.$$

Graph $\to$ matrix

The Laplacian matrix L of this graph has

(i,i)-entry equal to $d_i = \sum_{j \neq i} \lambda_{ij}$

(i,j)-entry equal to $-\lambda_{ij}$ if $i \neq j$.

So the row sums of L are all zero.

Graph $\to$ matrix

The Laplacian matrix L of this graph has

(i,i)-entry equal to $d_i = \sum_{j \neq i} \lambda_{ij}$

(i,j)-entry equal to $-\lambda_{ij}$ if $i \neq j$.

So the row sums of L are all zero.

Hence L has eigenvalue 0 on the all-1 vector.

Graph $\to$ matrix

The Laplacian matrix L of this graph has

(i,i)-entry equal to $d_i = \sum_{j \neq i} \lambda_{ij}$

(i,j)-entry equal to $-\lambda_{ij}$ if $i \neq j$.

So the row sums of L are all zero.

Hence L has eigenvalue 0 on the all-1 vector.

This *trivial* eigenvalue has multiplicity 1

$\iff$ the graph G is connected

$\iff$ all contrasts between treatment parameters are estimable.

Graph $\rightarrow$ matrix

The Laplacian matrix L of this graph has

(i,i)-entry equal to $d_i = \sum_{j \neq i} \lambda_{ij}$

(i,j)-entry equal to $-\lambda_{ij}$ if $i \neq j$.

So the row sums of L are all zero.

Hence L has eigenvalue 0 on the all-1 vector.

This *trivial* eigenvalue has multiplicity 1

$\iff$ the graph G is connected

$\iff$ all contrasts between treatment parameters are estimable.

Call the remaining eigenvalues *nontrivial*. They are all non-negative.

Graph $\rightarrow$ matrix

The Laplacian matrix L of this graph has

(i,i)-entry equal to $d_i = \sum_{j \neq i} \lambda_{ij}$

(i,j)-entry equal to $-\lambda_{ij}$ if $i \neq j$.

So the row sums of L are all zero.

Hence L has eigenvalue 0 on the all-1 vector.

This *trivial* eigenvalue has multiplicity 1

$\iff$ the graph G is connected

$\iff$ all contrasts between treatment parameters are estimable.

Call the remaining eigenvalues *nontrivial*. They are all non-negative.

The information matrix is precisely $k^{-1}L$.

Estimation and variance when there are blocks

Theorem

Assume that all the noise is independent, with variance σ^2. Then the variance of the best linear unbiased estimator of the simple difference $\tau_i - \tau_j$ is

$$V_{ij} = \left(L^-_{ii} + L^-_{jj} - 2L^-_{ij}\right) k\sigma^2,$$

where L^- is any generalized inverse of L.

Estimation and variance when there are blocks

Theorem

Assume that all the noise is independent, with variance σ^2. Then the variance of the best linear unbiased estimator of the simple difference $\tau_i - \tau_j$ is

$$V_{ij} = \left(L^-_{ii} + L^-_{jj} - 2L^-_{ij}\right) k\sigma^2,$$

where L^- is any generalized inverse of L.

Put $\bar{V} =$ average value of the V_{ij}. Then

$$\bar{V} = \frac{2k\sigma^2 \operatorname{Tr}(L^-)}{v-1} = 2k\sigma^2 \times \frac{1}{\text{harmonic mean of } \theta_1, \ldots, \theta_{v-1}},$$

where $\theta_1, \ldots, \theta_{v-1}$ are the nontrivial eigenvalues of L.

Estimation and variance when there are blocks

Theorem

Assume that all the noise is independent, with variance σ^2. Then the variance of the best linear unbiased estimator of the simple difference $\tau_i - \tau_j$ is

$$V_{ij} = \left(L^-_{ii} + L^-_{jj} - 2L^-_{ij}\right) k\sigma^2,$$

where L^- is any generalized inverse of L.

Put $\bar{V} =$ average value of the V_{ij}. Then

$$\bar{V} = \frac{2k\sigma^2 \operatorname{Tr}(L^-)}{v-1} = 2k\sigma^2 \times \frac{1}{\text{harmonic mean of } \theta_1, \ldots, \theta_{v-1}},$$

where $\theta_1, \ldots, \theta_{v-1}$ are the nontrivial eigenvalues of L.

$$\begin{aligned} \text{A-optimal} \quad &\Longleftrightarrow \quad \text{minimize } \bar{V} \\ &\Longleftrightarrow \quad \text{maximize harmonic mean of } \theta_1, \ldots, \theta_{v-1}. \end{aligned}$$

First folklore for block designs

Theorem

If v divides N and there are no blocks then the A-optimal designs are precisely the equireplicate ones, that is, those where all treatments have equal replication.

First folklore for block designs

Theorem

If v divides N and there are no blocks then the A-optimal designs are precisely the equireplicate ones, that is, those where all treatments have equal replication.

Folklore surrogate

If v divides N and there are b blocks of size k where $b > 1$ then the set of equireplicate designs contains some A-optimal designs.

First folklore for block designs

Theorem

If v divides N and there are no blocks then the A-optimal designs are precisely the equireplicate ones, that is, those where all treatments have equal replication.

Folklore surrogate

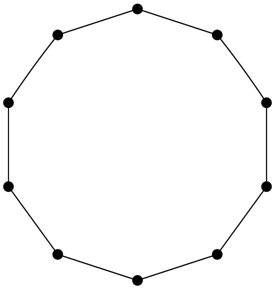
If v divides N and there are b blocks of size k where $b > 1$ then the set of equireplicate designs contains some A-optimal designs.

This was believed from the introduction of incomplete-block designs in the 1930s, so the search for good designs was restricted to equireplicate ones.

By the 1990s, it had been shown to be false in general.

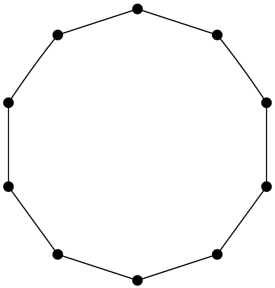
Designs for $k = 2$ when $b = v$ (blocks shown as edges)

The only connected equireplicate design is the cycle.



Designs for $k = 2$ when $b = v$ (blocks shown as edges)

The only connected equireplicate design is the cycle.

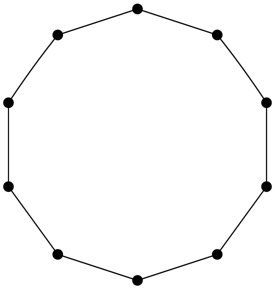


If the distance between i and j is w

$$V_{ij} = \frac{2w(v-w)}{v}\sigma^2,$$

Designs for $k = 2$ when $b = v$ (blocks shown as edges)

The only connected equireplicate design is the cycle.



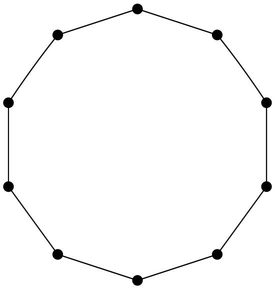
If the distance between i and j is w

$$V_{ij} = \frac{2w(v-w)}{v}\sigma^2,$$

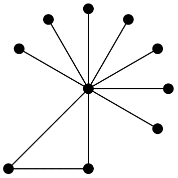
$> 4\sigma^2$ if $v \geq 10$ and $3 \leq w$.

Designs for $k = 2$ when $b = v$ (blocks shown as edges)

The only connected equireplicate design is the cycle.



Here is an alternative design.



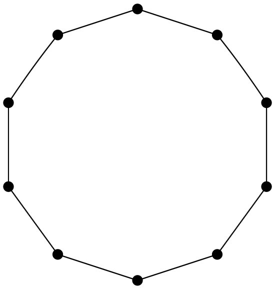
If the distance between i and j is w

$$V_{ij} = \frac{2w(v-w)}{v}\sigma^2,$$

$> 4\sigma^2$ if $v \geq 10$ and $3 \leq w$.

Designs for $k = 2$ when $b = v$ (blocks shown as edges)

The only connected equireplicate design is the cycle.

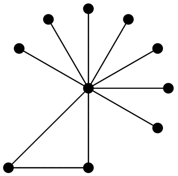


If the distance between i and j is w

$$V_{ij} = \frac{2w(v-w)}{v}\sigma^2,$$

$> 4\sigma^2$ if $v \geq 10$ and $3 \leq w$.

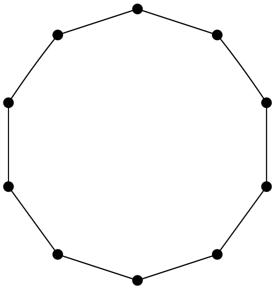
Here is an alternative design.



$V_{ij} \leq 4\sigma^2$ for all i, j.

Designs for $k = 2$ when $b = v$ (blocks shown as edges)

The only connected equireplicate design is the cycle.

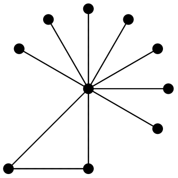


If the distance between i and j is w

$$V_{ij} = \frac{2w(v-w)}{v}\sigma^2,$$

$> 4\sigma^2$ if $v \geq 10$ and $3 \leq w$.

Here is an alternative design.



$V_{ij} \leq 4\sigma^2$ for all i, j.

A star attached to a triangle is A-optimal for all $v \geq 12$.

Reactions

Statistician: that result cannot be correct, because we know that equal replication is best.

Reactions

Statistician: that result cannot be correct, because we know that equal replication is best.

Biologist: the second design should be used, because we know that we should compare all treatments with the same thing.

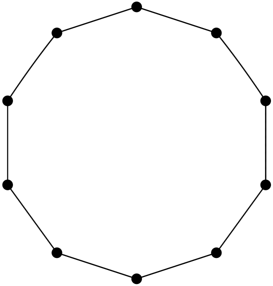
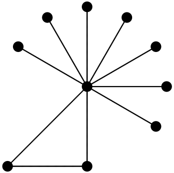
Reactions

Statistician: that result cannot be correct, because we know that equal replication is best.

Biologist: the second design should be used, because we know that we should compare all treatments with the same thing.

Producer of one of the compared treatments: that's not fair! My treatment has replication only one, so the variances of its comparisons with other treatments will be too large.

What about symmetry and regularity?

Design		
Automorphisms	2×10	$2 \times 7!$
	regular	more symmetries

Some history

In 1980, Jones and Eccleston published a short paper in JRSSB on the results of a computer search for A-optimal designs with
$k = 2$ and $v = b \leq 10$
(so average replication $= \bar{r} = 2$);
when $v = 9$ and $v = 10$ the optimal design is a star attached to a square.

Some history

In 1980, Jones and Eccleston published a short paper in JRSSB on the results of a computer search for A-optimal designs with $k = 2$ and $v = b \leq 10$
(so average replication $= \bar{r} = 2$);
when $v = 9$ and $v = 10$ the optimal design is a star attached to a square.

Their work was ignored by most statisticians, because we were so sure that equireplicate designs are best that we assumed that there was an error in the computation.

Balance

Definition

A balanced incomplete-block design (BIBD) is a block design with $k < v$ in which no treatment occurs more than once in any block and all treatment concurrences are equal.

Balance

Definition

A balanced incomplete-block design (BIBD) is a block design with $k < v$ in which no treatment occurs more than once in any block and all treatment concurrences are equal.

Theorem

Balanced incomplete-block designs are A-optimal.

Balance

Definition

A balanced incomplete-block design (BIBD) is a block design with $k < v$ in which no treatment occurs more than once in any block and all treatment concurrences are equal.

Theorem

Balanced incomplete-block designs are A-optimal.

Folklore surrogate

A BIBD is optimal even if it does not use all the available blocks.

Balance

Definition

A balanced incomplete-block design (BIBD) is a block design with $k < v$ in which no treatment occurs more than once in any block and all treatment concurrences are equal.

Theorem

Balanced incomplete-block designs are A-optimal.

Folklore surrogate

A BIBD is optimal even if it does not use all the available blocks.

This is nonsense: the theorem is comparing designs using the same number of experimental units.

A comparison

Folklore surrogate

If k divides v and there is a BIBD for v treatments in $b - (v/k)$ blocks of size k, then the best thing to do is to use that BIBD and make the extra blocks out of any partition of the treatments into sets of size k.

A comparison

Folklore surrogate

If k divides v and there is a BIBD for v treatments in $b - (v/k)$ blocks of size k, then the best thing to do is to use that BIBD and make the extra blocks out of any partition of the treatments into sets of size k.

The false reasoning in this is more subtle.

A comparison

Folklore surrogate

If k divides v and there is a BIBD for v treatments in $b-(v/k)$ blocks of size k, then the best thing to do is to use that BIBD and make the extra blocks out of any partition of the treatments into sets of size k.

The false reasoning in this is more subtle.

Example

Suppose that $v=6$, $b=12$ and $k=3$.

Design	$\bar{V}/\sigma^2$
BIBD with 10 blocks	0.5
That BIBD with two more blocks	0.42
Develop $\{0,1,2\}$ and $\{0,1,3\}$ modulo 6	0.4196...

How should we relax the BIBD condition?

Recall: the concurrence matrix Λ has entries λ_{ij},
where λ_{ij} is the number of blocks containing treatments i and j.

How should we relax the BIBD condition?

Recall: the concurrence matrix Λ has entries λ_{ij},
where λ_{ij} is the number of blocks containing treatments i and j.

$$\text{BIBD} \qquad \Lambda = rI + \lambda(J - I)$$

How should we relax the BIBD condition?

Recall: the concurrence matrix Λ has entries λ_{ij},
where λ_{ij} is the number of blocks containing treatments i and j.

BIBD	Λ	$=$	$rI + \lambda(J - I)$
Relax	Λ	$=$	$rI + (\lambda + 1)A + \lambda(J - A - I)$
			where A^2 is a linear combination of I, A and J

How should we relax the BIBD condition?

Recall: the concurrence matrix Λ has entries λ_{ij},
where λ_{ij} is the number of blocks containing treatments i and j.

BIBD	Λ	$=$	$rI + \lambda(J - I)$
Relax	Λ	$=$	$rI + (\lambda + 1)A + \lambda(J - A - I)$
			where A^2 is a linear combination of I, A and J
PBIBD(2)	Λ	$=$	$rI + \lambda_1 A + \lambda_2(J - A - I)$
			where A^2 is a linear combination of I, A and J

Relax $\Rightarrow$ Partially Balanced IBD
with 2 associate classes

How should we relax the BIBD condition?

Recall: the concurrence matrix Λ has entries λ_{ij},
where λ_{ij} is the number of blocks containing treatments i and j.

BIBD	Λ	$=$	$rI + \lambda(J - I)$
Relax	Λ	$=$	$rI + (\lambda+1)A + \lambda(J - A - I)$
			where A^2 is a linear combination of I, A and J
PBIBD(2)	Λ	$=$	$rI + \lambda_1 A + \lambda_2(J - A - I)$
			where A^2 is a linear combination of I, A and J
RGD	Λ	$=$	$rI + (\lambda+1)A + \lambda(J - A - I)$

Relax $\Rightarrow$ Partially Balanced IBD with 2 associate classes

$\Downarrow$

Regular Graph Design

How should we relax the BIBD condition?

Recall: the concurrence matrix Λ has entries λ_{ij},
where λ_{ij} is the number of blocks containing treatments i and j.

BIBD	Λ	$=$	$rI + \lambda(J - I)$
Relax	Λ	$=$	$rI + (\lambda+1)A + \lambda(J - A - I)$
			where A^2 is a linear combination of I, A and J
PBIBD(2)	Λ	$=$	$rI + \lambda_1 A + \lambda_2(J - A - I)$
			where A^2 is a linear combination of I, A and J
RGD	Λ	$=$	$rI + (\lambda+1)A + \lambda(J - A - I)$
PBIBD(s)	Λ	$=$	$rI + \lambda_1 A_1 + \cdots + \lambda_s A_s$
			where $I + A_1 + \cdots + A_s = J$ and
			$A_i A_j$ is a linear combination of $I, A_1, \ldots, A_s$.

Relax $\Rightarrow$ Partially Balanced IBD with 2 associate classes

$\Downarrow$

Regular Graph Design

Variance and concurrence

Folklore surrogate

If there are any regular graph designs, all optimal designs are RGDs.

Folklore surrogate

Variance V_{ij} *is a decreasing function of* λ_{ij}.

Variance and concurrence

Folklore surrogate

If there are any regular graph designs, all optimal designs are RGDs.

Folklore surrogate

Variance V_{ij} *is a decreasing function of* λ_{ij}*.*

Neither is true in general. Counterexamples exist.

Variance and concurrence

Folklore surrogate

If there are any regular graph designs, all optimal designs are RGDs.

Folklore surrogate

Variance V_{ij} *is a decreasing function of* λ_{ij}.

Neither is true in general. Counterexamples exist.

Theorem

If the information matrix has only two distinct non-trivial eigenvalues then V_{ij} *decreases linearly as* λ_{ij} *increases.*

Variance and concurrence

Folklore surrogate

If there are any regular graph designs, all optimal designs are RGDs.

Folklore surrogate

Variance V_{ij} is a decreasing function of λ_{ij}.

Neither is true in general. Counterexamples exist.

Theorem

If the information matrix has only two distinct non-trivial eigenvalues then V_{ij} decreases linearly as λ_{ij} increases.

In particular, this is true if the design is partially balanced with two associate classes, which means that the information matrix is in the Bose–Mesner algebra of a strongly regular graph.

Variance and concurrence

Folklore surrogate

If there are any regular graph designs, all optimal designs are RGDs.

Folklore surrogate

Variance V_{ij} is a decreasing function of λ_{ij}.

Neither is true in general. Counterexamples exist.

Theorem

If the information matrix has only two distinct non-trivial eigenvalues then V_{ij} decreases linearly as λ_{ij} increases.

In particular, this is true if the design is partially balanced with two associate classes, which means that the information matrix is in the Bose–Mesner algebra of a strongly regular graph.

Theorem

If the design is partially balanced with two associate classes, and the concurrences differ by 1, and one of those eigenvalues is equal to r, then the block design is A-optimal.

What about distance in the concurrence graph G?

Recall: G has one vertex for each treatment
and λ_{ij} edges between vertices i and j.

What about distance in the concurrence graph G?

Recall: G has one vertex for each treatment
and λ_{ij} edges between vertices i and j.

The distance between i and j in G is
the length of the shortest path between vertices i and j.

What about distance in the concurrence graph G?

Recall: G has one vertex for each treatment
and λ_{ij} edges between vertices i and j.

The distance between i and j in G is
the length of the shortest path between vertices i and j.

Folklore surrogate

Variance increases with distance in the concurrence graph.

What about distance in the concurrence graph G?

Recall: G has one vertex for each treatment
and λ_{ij} edges between vertices i and j.

The distance between i and j in G is
the length of the shortest path between vertices i and j.

Folklore surrogate

Variance increases with distance in the concurrence graph.

This is not true in general.

Electrical networks

We can consider the concurrence graph as an electrical network with a 1-ohm resistance in each edge. Connect a 1-volt battery between vertices i and j. Current flows in the network, according to these rules.

1. Ohm's Law:
   In every edge, voltage drop = current $\times$ resistance = current.
2. Kirchhoff's Voltage Law:
   The total voltage drop from one vertex to any other vertex is the same no matter which path we take from one to the other.
3. Kirchhoff's Current Law:
   At every vertex which is not connected to the battery, the total current coming in is equal to the total current going out.

Find the total current I from i to j, then use Ohm's Law to define the effective resistance R_{ij} between i and j as $1/I$.

Electrical networks: resistance distance

Theorem

The effective resistance R_{ij} between vertices i and j is

$$R_{ij} = \left(L^-_{ii} + L^-_{jj} - 2L^-_{ij} \right).$$

Electrical networks: resistance distance

Theorem

The effective resistance R_{ij} between vertices i and j is

$$R_{ij} = \left(L^-_{ii} + L^-_{jj} - 2L^-_{ij} \right).$$

So

$$V_{ij} = R_{ij} \times k\sigma^2.$$

Electrical networks: resistance distance

Theorem

The effective resistance R_{ij} between vertices i and j is

$$R_{ij} = \left(L^-_{ii} + L^-_{jj} - 2L^-_{ij}\right).$$

So

$$V_{ij} = R_{ij} \times k\sigma^2.$$

In other words, variance is proportional to resistance distance.

Electrical networks: resistance distance

Theorem

The effective resistance R_{ij} between vertices i and j is

$$R_{ij} = \left(L^{-}_{ii} + L^{-}_{jj} - 2L^{-}_{ij} \right).$$

So

$$V_{ij} = R_{ij} \times k\sigma^2.$$

In other words, variance is proportional to resistance distance.
Effective resistances are easy to calculate without matrix inversion if the graph is sparse.

So how do we find good designs?

The numbers v, b and k are specified to us.
It is rather rare for these to fit one of the theorems that guarantees a design to be A-optimal.
So how do we find a good design?

So how do we find good designs?

The numbers v, b and k are specified to us.
It is rather rare for these to fit one of the theorems that guarantees a design to be A-optimal.
So how do we find a good design?

1. Computer search.
2. Use patterns.
3. Accident.

Computer search

Except for very small designs,
exhaustive search is not usually feasible.
Here is one common approach.

1. Start with a random design.
2. Search among “close” designs
   (for example, swap a pair of treatments between blocks).
3. If a neighbouring design is better, move to it,
   and repeat from Step 2.
4. If no neighbouring design is better, record this design.
5. Repeat from Step 1 many times.
   Then choose the best of the recorded designs.

Computer search

Except for very small designs,
exhaustive search is not usually feasible.
Here is one common approach.

1. Start with a random design.
2. Search among "close" designs
(for example, swap a pair of treatments between blocks).
3. If a neighbouring design is better, move to it,
and repeat from Step 2.
4. If no neighbouring design is better, record this design.
5. Repeat from Step 1 many times.
Then choose the best of the recorded designs.

The purpose of the last step is to avoid being stuck in a local optimum.

How successful is computer search?

It usually finds fairly good designs.

How successful is computer search?

It usually finds fairly good designs.

However, if the optimal design has a high degree of symmetry, then it is often sitting on the top of a mountain with very steep sides, and so this approach will not find it.

Use patterns

I typically start with a combinatorial object with v points
which is either highly regular or highly symmetric,
and then see if I can use the patterns in that
to construct a design with the specified parameters.

Use patterns

I typically start with a combinatorial object with v points
which is either highly regular or highly symmetric,
and then see if I can use the patterns in that
to construct a design with the specified parameters.

For example:

if $v = 7$ begin with the points of the Fano plane;

Use patterns

I typically start with a combinatorial object with v points which is either highly regular or highly symmetric, and then see if I can use the patterns in that to construct a design with the specified parameters.

For example:

if $v = 7$ begin with the points of the Fano plane;

if $v = 8$ start with the vertices of a cube;

Use patterns

I typically start with a combinatorial object with v points which is either highly regular or highly symmetric, and then see if I can use the patterns in that to construct a design with the specified parameters.

For example:

if $v = 7$ begin with the points of the Fano plane;

if $v = 8$ start with the vertices of a cube;

if $v = 10$ think about all pairs from $\{1, 2, 3, 4, 5\}$;

Use patterns

I typically start with a combinatorial object with v points which is either highly regular or highly symmetric, and then see if I can use the patterns in that to construct a design with the specified parameters.

For example:

if $v = 7$ begin with the points of the Fano plane;

if $v = 8$ start with the vertices of a cube;

if $v = 10$ think about all pairs from $\{1, 2, 3, 4, 5\}$;

if $v = 12$ use the faces of a regular dodecahedron;

Use patterns

I typically start with a combinatorial object with v points which is either highly regular or highly symmetric, and then see if I can use the patterns in that to construct a design with the specified parameters.

For example:

- if $v = 7$ begin with the points of the Fano plane;
- if $v = 8$ start with the vertices of a cube;
- if $v = 10$ think about all pairs from $\{1, 2, 3, 4, 5\}$;
- if $v = 12$ use the faces of a regular dodecahedron;
- if $v = 21$ use the points of the projective plane over the finite field with 4 elements.

How successful is the pattern method?

If the optimal design is highly symmetric,
this method can find it when computer search does not.

How successful is the pattern method?

If the optimal design is highly symmetric,
this method can find it when computer search does not.

It usually finds good designs, but will not find the optimal one if none of the optimal ones is highly symmetric.

An example: $v = 10$, $b = 30$, $k = 2$ $(A = 2\sigma^2/(r\bar{V}))$

Method	Patterns	Search
Design	Treatments are all pairs from $\{1,2,3,4,5\}$. Two pairs form a block if they overlap.	Treatments are the vertices of a 6-cycle and 4 more points. Blocks are the edges of the 6-cycle, and all duos with one from the 6 and one from the 4.
$\bar{V}/\sigma^2$	0.63333	0.62698
A	0.52632	0.53165
Auto-morphisms	$5! = 120$	$12 \times 4! = 288$
	regular	more symmetries

Accident: an example

Trials of new crop varieties typically have a large number of varieties.

Accident: an example

Trials of new crop varieties typically have a large number of varieties.

Even at a well-run testing centre,
inhomogeneity among the plots (experimental units) makes it desirable to group the plots into homogeneous blocks,
usually too small to contain all the varieties.

Accident: an example

Trials of new crop varieties typically have a large number of varieties.

Even at a well-run testing centre,
inhomogeneity among the plots (experimental units) makes it desirable to group the plots into homogeneous blocks,
usually too small to contain all the varieties.

For management reasons, it is often convenient if the blocks can themselves be grouped into replicates, in such a way that each variety occurs exactly once in each replicate. Such a block design is called resolvable.

Accident: an example

Trials of new crop varieties typically have a large number of varieties.

Even at a well-run testing centre,
inhomogeneity among the plots (experimental units) makes it desirable to group the plots into homogeneous blocks,
usually too small to contain all the varieties.

For management reasons, it is often convenient if the blocks can themselves be grouped into replicates, in such a way that each variety occurs exactly once in each replicate. Such a block design is called resolvable.

A block design is A-optimal if it minimizes the sum of the variances of the estimators of differences between varieties.

Square lattice designs

Yates (Rothamsted Experimental Station: 1936, 1937) introduced square lattice designs for this purpose.
The number of varieties has the form n^2 for some integer n, and each replicate consists of n blocks of n plots.

Square lattice designs

Yates (Rothamsted Experimental Station: 1936, 1937) introduced square lattice designs for this purpose.
The number of varieties has the form n^2 for some integer n, and each replicate consists of n blocks of n plots.
Imagine the varieties listed in an abstract $n \times n$ square array.
The rows of this array form the blocks of the first replicate, and the columns of this array form the blocks of the second replicate.

Square lattice designs

Yates (Rothamsted Experimental Station: 1936, 1937) introduced square lattice designs for this purpose.
The number of varieties has the form n^2 for some integer n, and each replicate consists of n blocks of n plots.
Imagine the varieties listed in an abstract $n \times n$ square array.
The rows of this array form the blocks of the first replicate, and the columns of this array form the blocks of the second replicate.

Let r be the number of replicates. If $r > 2$ then $r - 2$ mutually orthogonal Latin squares of order n are needed. For each of these Latin squares, each letter determines a block of size n.

Mutually orthogonal Latin squares

Definition

A pair of Latin squares of order n are orthogonal to each other if, when they are superposed, each letter of one occurs exactly once with each letter of the other.

Mutually orthogonal Latin squares

Definition

A pair of Latin squares of order n are orthogonal to each other if, when they are superposed, each letter of one occurs exactly once with each letter of the other.

Here are a pair of orthogonal Latin squares of order 4.

A	B	C	D
B	A	D	C
C	D	A	B
D	C	B	A

α	β	γ	δ
γ	δ	α	β
δ	γ	β	α
β	α	δ	γ

Mutually orthogonal Latin squares

Definition

A pair of Latin squares of order n are orthogonal to each other if, when they are superposed, each letter of one occurs exactly once with each letter of the other.

Here are a pair of orthogonal Latin squares of order 4.

A	B	C	D
B	A	D	C
C	D	A	B
D	C	B	A

α	β	γ	δ
γ	δ	α	β
δ	γ	β	α
β	α	δ	γ

Mutually orthogonal Latin squares

Definition

A pair of Latin squares of order n are orthogonal to each other if, when they are superposed, each letter of one occurs exactly once with each letter of the other.

Here are a pair of orthogonal Latin squares of order 4.

A	B	C	D
B	A	D	C
C	D	A	B
D	C	B	A

α	β	γ	δ
γ	δ	α	β
δ	γ	β	α
β	α	δ	γ

Definition

A collection of Latin squares of the same order is mutually orthogonal if every pair is orthogonal.

Square lattice designs for 16 varieties in 2–4 replicates

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

A	B	C	D
B	A	D	C
C	D	A	B
D	C	B	A

α	β	γ	δ
γ	δ	α	β
δ	γ	β	α
β	α	δ	γ

Square lattice designs for 16 varieties in 2–4 replicates

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

A	B	C	D
B	A	D	C
C	D	A	B
D	C	B	A

α	β	γ	δ
γ	δ	α	β
δ	γ	β	α
β	α	δ	γ

Replicate 1

1	5	9	13
2	6	10	14
3	7	11	15
4	8	12	16

Replicate 2

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Square lattice designs for 16 varieties in 2–4 replicates

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

A	B	C	D
B	A	D	C
C	D	A	B
D	C	B	A

α	β	γ	δ
γ	δ	α	β
δ	γ	β	α
β	α	δ	γ

Replicate 1

1	5	9	13
2	6	10	14
3	7	11	15
4	8	12	16

Replicate 2

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Replicate 3

1	2	3	4
6	5	8	7
11	12	9	10
16	15	14	13

Square lattice designs for 16 varieties in 2–4 replicates

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

A	B	C	D
B	A	D	C
C	D	A	B
D	C	B	A

α	β	γ	δ
γ	δ	α	β
δ	γ	β	α
β	α	δ	γ

Replicate 1

1	5	9	13
2	6	10	14
3	7	11	15
4	8	12	16

Replicate 2

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Replicate 3

1	2	3	4
6	5	8	7
11	12	9	10
16	15	14	13

Replicate 4

1	2	3	4
7	8	5	6
12	11	10	9
14	13	16	15

Square lattice designs for 16 varieties in 2–4 replicates

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

A	B	C	D
B	A	D	C
C	D	A	B
D	C	B	A

α	β	γ	δ
γ	δ	α	β
δ	γ	β	α
β	α	δ	γ

Replicate 1

1	5	9	13
2	6	10	14
3	7	11	15
4	8	12	16

Replicate 2

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Replicate 3

1	2	3	4
6	5	8	7
11	12	9	10
16	15	14	13

Replicate 4

1	2	3	4
7	8	5	6
12	11	10	9
14	13	16	15

Using a third Latin square orthogonal to the previous two Latin squares gives a fifth replicate, if required.

Square lattice designs for 16 varieties in 2–4 replicates

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

A	B	C	D
B	A	D	C
C	D	A	B
D	C	B	A

α	β	γ	δ
γ	δ	α	β
δ	γ	β	α
β	α	δ	γ

Replicate 1				Replicate 2				Replicate 3				Replicate 4			
1	5	9	13	1	2	3	4	1	2	3	4	1	2	3	4
2	6	10	14	5	6	7	8	6	5	8	7	7	8	5	6
3	7	11	15	9	10	11	12	11	12	9	10	12	11	10	9
4	8	12	16	13	14	15	16	16	15	14	13	14	13	16	15

Using a third Latin square orthogonal to the previous two Latin squares gives a fifth replicate, if required.

Square lattice designs are resolvable and A-optimal.
All pairwise variety concurrences are in $\{0, 1\}$.

We have a problem when $n = 6$

If $n \in \{2, 3, 4, 5, 7, 8, 9\}$ then there is a complete set of $n - 1$ mutually orthogonal Latin squares of order n.

We have a problem when $n = 6$

If $n \in \{2, 3, 4, 5, 7, 8, 9\}$ then there is a complete set of $n - 1$ mutually orthogonal Latin squares of order n.

Using these gives a square lattice design
for n^2 treatments in $n(n + 1)$ blocks of size n,
which is a balanced incomplete-block design.

We have a problem when $n = 6$

If $n \in \{2, 3, 4, 5, 7, 8, 9\}$ then there is a complete set of $n - 1$ mutually orthogonal Latin squares of order n.

Using these gives a square lattice design
for n^2 treatments in $n(n+1)$ blocks of size n,
which is a balanced incomplete-block design.

There is not even a pair of mutually orthogonal Latin squares of order 6, so square lattice designs for 36 treatments are available for 2 or 3 replicates only.

We have a problem when $n = 6$

If $n \in \{2, 3, 4, 5, 7, 8, 9\}$ then there is a complete set of $n - 1$ mutually orthogonal Latin squares of order n.

Using these gives a square lattice design
for n^2 treatments in $n(n+1)$ blocks of size n,
which is a balanced incomplete-block design.

There is not even a pair of mutually orthogonal Latin squares of order 6, so square lattice designs for 36 treatments are available for 2 or 3 replicates only.

Patterson and Williams (University of Edinburgh: 1976)
used computer search to find a design for
36 treatments in 4 replicates of blocks of size 6
with all concurrences in $\{0, 1, 2\}$.
The average variance is very little more than the unachievable lower bound.

A new design problem: sesqui-arrays

A sesqui-array of order n is an allocation of $n(n+1)$ letters to the cells of rectangle with $n+1$ rows and n^2 columns, satisfying conditions (i) and (ii) below.

A new design problem: sesqui-arrays

A sesqui-array of order n is an allocation of $n(n+1)$ letters to the cells of rectangle with $n+1$ rows and n^2 columns, satisfying conditions (i) and (ii) below.

Example with $n = 3$

D	*H*	*F*	*L*	*E*	*K*	*I*	*G*	*J*
A	*K*	*I*	*B*	*J*	*G*	*C*	*L*	*H*
J	*A*	*L*	*D*	*B*	*F*	*K*	*E*	*C*
G	*E*	*A*	*H*	*I*	*B*	*D*	*C*	*F*

A new design problem: sesqui-arrays

A sesqui-array of order n is an allocation of $n(n+1)$ letters to the cells of rectangle with $n+1$ rows and n^2 columns, satisfying conditions (i) and (ii) below.

Example with $n = 3$

D	H	F	L	E	K	I	G	J
A	K	I	B	J	G	C	L	H
J	A	L	D	B	F	K	E	C
G	E	A	H	I	B	D	C	F

Condition (i) Each letter occurs in all rows except one.

A new design problem: sesqui-arrays

A sesqui-array of order n is an allocation of $n(n+1)$ letters to the cells of rectangle with $n+1$ rows and n^2 columns, satisfying conditions (i) and (ii) below.

Example with $n = 3$

D	H	F	L	E	K	I	G	J
A	K	I	B	J	G	C	L	H
J	A	L	D	B	F	K	E	C
G	E	A	H	I	B	D	C	F

Condition (i) Each letter occurs in all rows except one.

Condition (ii) Each row has n letters in common with each column.

Constructing sesqui-arrays

Tomas Nilson (University of mid-Sweden) and
Peter Cameron (University of St Andrews) hoped to give a general construction of sesqui-arrays for all $n \geq 3$.

Constructing sesqui-arrays

Tomas Nilson (University of mid-Sweden) and
Peter Cameron (University of St Andrews) hoped to give a general construction of sesqui-arrays for all $n \geq 3$.

TN found a general construction, using a pair of mutually orthogonal Latin squares of order n. So this works for all positive integers n except for $n \in \{1, 2, 6\}$.

Constructing sesqui-arrays

Tomas Nilson (University of mid-Sweden) and
Peter Cameron (University of St Andrews) hoped to give a general construction of sesqui-arrays for all $n \geq 3$.

TN found a general construction, using a pair of mutually orthogonal Latin squares of order n. So this works for all positive integers n except for $n \in \{1, 2, 6\}$.

This motivated PJC to find a sesqui-array for $n = 6$.

Constructing sesqui-arrays

Tomas Nilson (University of mid-Sweden) and
Peter Cameron (University of St Andrews) hoped to give a general construction of sesqui-arrays for all $n \geq 3$.

TN found a general construction, using a pair of mutually orthogonal Latin squares of order n. So this works for all positive integers n except for $n \in \{1, 2, 6\}$.

This motivated PJC to find a sesqui-array for $n = 6$.

Later, RAB found a simpler version of TN's construction, that needs a Latin square of order n but not orthogonal Latin squares. So $n = 6$ is covered. If this had been known earlier, PJC would not have found the nice design for $n = 6$.

Naughty but nice

6 is uniquely **BAD** amongst positive integers in that
it is big enough to have a pair of orthogonal Latin squares
but there are no such squares.

Naughty but nice

6 is uniquely **BAD** amongst positive integers in that it is big enough to have a pair of orthogonal Latin squares but there are no such squares.

6 is uniquely **GOOD** amongst positive integers in that the symmetric group S_6 of all permutations of $\{1, 2, 3, 4, 5, 6\}$ has an automorphism σ which is not of the form $\sigma(g) = h^{-1}gh$.

Naughty but nice

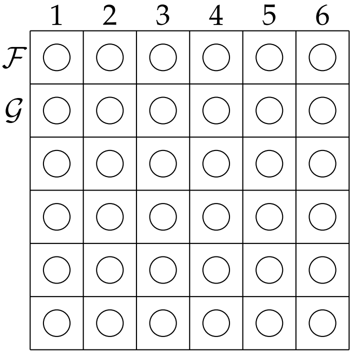
6 is uniquely **BAD** amongst positive integers in that
it is big enough to have a pair of orthogonal Latin squares
but there are no such squares.

6 is uniquely **GOOD** amongst positive integers in that
the symmetric group S_6 of all permutations of $\{1, 2, 3, 4, 5, 6\}$
has an automorphism σ which is not of the form $\sigma(g) = h^{-1}gh$.

This can be used to construct the Sylvester graph,
which has 36 vertices, all with valency 5.

The Sylvester graph

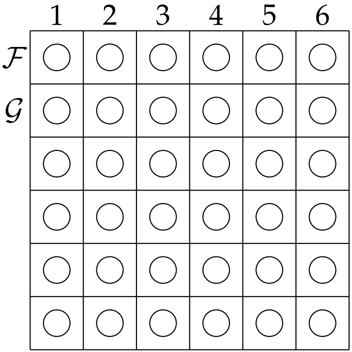
The vertices can be thought of as the cells of a 6×6 grid.



Rows are labelled by the one-factorizations (edge-colourings) of K_6.

The Sylvester graph

The vertices can be thought of as the cells of a 6×6 grid.



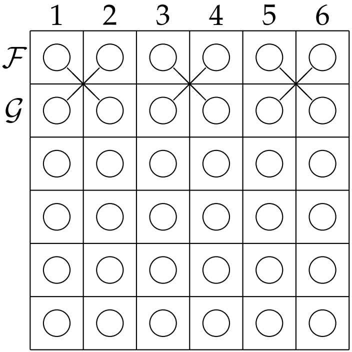
Rows are labelled by the one-factorizations (edge-colourings) of K_6.

$$\mathcal{F} = ||12|34|56||13|25|46||14|26|35||15|24|36||16|23|45||$$

$$\mathcal{G} = ||12|34|56||23|15|46||24|16|35||25|14|36||26|13|45|| = \mathcal{F}^{(12)}$$

The Sylvester graph

The vertices can be thought of as the cells of a 6×6 grid.



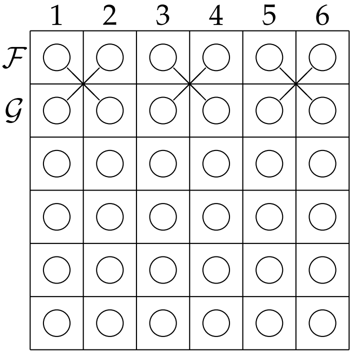
Rows are labelled by the one-factorizations (edge-colourings) of K_6.

$$\mathcal{F} = ||12|34|56||13|25|46||14|26|35||15|24|36||16|23|45||$$

$$\mathcal{G} = ||12|34|56||23|15|46||24|16|35||25|14|36||26|13|45|| = \mathcal{F}^{(12)}$$

The Sylvester graph

The vertices can be thought of as the cells of a 6×6 grid.



Rows are labelled by the one-factorizations (edge-colourings) of K_6.

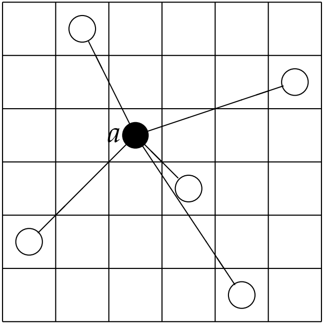
$$\mathcal{F} = ||12|34|56||13|25|46||14|26|35||15|24|36||16|23|45||$$

$$\mathcal{G} = ||12|34|56||23|15|46||24|16|35||25|14|36||26|13|45|| = \mathcal{F}^{(12)}$$

Automorphisms: S_6 on rows and on columns at the same time; the outer automorphism of S_6 swaps rows with columns.

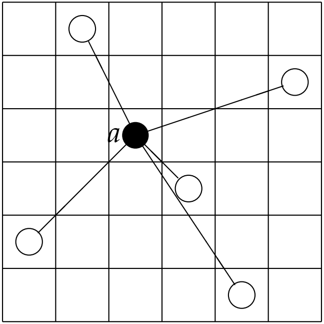
The Sylvester graph and its starfish

At each vertex a, the *starfish* $S(a)$ defined by the 5 edges at a has 6 vertices, one in each row and one in each column.



The Sylvester graph and its starfish

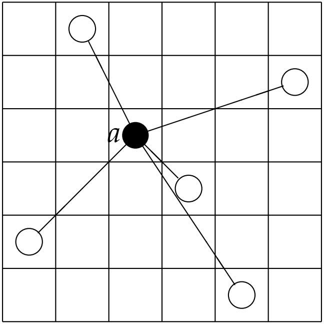
At each vertex a, the *starfish* $S(a)$ defined by the 5 edges at a has 6 vertices, one in each row and one in each column.



Use each starfish as a block of size 6.

The Sylvester graph and its starfish

At each vertex a, the *starfish* $S(a)$ defined by the 5 edges at a has 6 vertices, one in each row and one in each column.

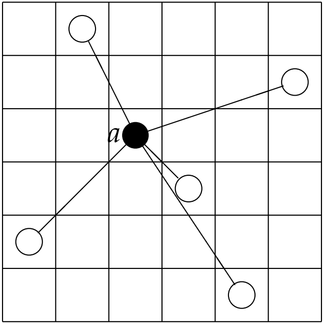


Use each starfish as a block of size 6.

The *galaxy* of starfish with centres in a single column give a single replicate.

The Sylvester graph and its starfish

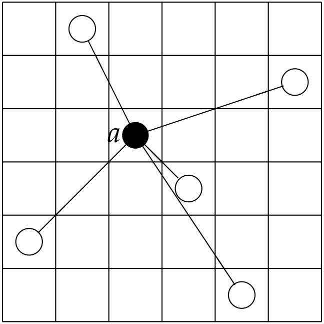
At each vertex a, the *starfish* $S(a)$ defined by the 5 edges at a has 6 vertices, one in each row and one in each column.



Use each starfish as a block of size 6.
The *galaxy* of starfish with centres in a single column give a single replicate. Hence up to six replicates.

The Sylvester graph and its starfish

At each vertex a, the *starfish* $S(a)$ defined by the 5 edges at a has 6 vertices, one in each row and one in each column.



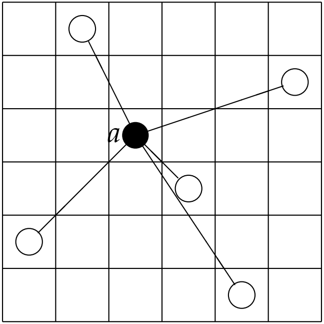
Use each starfish as a block of size 6.

The *galaxy* of starfish with centres in a single column give a single replicate. Hence up to six replicates.

Rows and columns give two further replicates, if needed.

The Sylvester graph and its starfish

At each vertex a, the *starfish* $S(a)$ defined by the 5 edges at a has 6 vertices, one in each row and one in each column.



Use each starfish as a block of size 6.
The *galaxy* of starfish with centres in a single column give a single replicate. Hence up to six replicates.
Rows and columns give two further replicates, if needed.
All these designs have average variance very close to the unachievable lower bound.

Personal communication from Emlyn Williams

I gave a talk about these designs in August 2017
at the meeting on *Latest advances in the theory and applications of design and analysis of experiments*
in the Banff International Research Station in Canada.

Personal communication from Emlyn Williams

I gave a talk about these designs in August 2017
at the meeting on *Latest advances in the theory and applications of design and analysis of experiments*
in the Banff International Research Station in Canada.

They video all lectures, and make them available on the web.

Personal communication from Emlyn Williams

I gave a talk about these designs in August 2017
at the meeting on *Latest advances in the theory and applications of design and analysis of experiments*
in the Banff International Research Station in Canada.

They video all lectures, and make them available on the web.

Emlyn Williams learnt about this,
and watched the video of my lecture.

Personal communication from Emlyn Williams

I gave a talk about these designs in August 2017
at the meeting on *Latest advances in the theory and applications of design and analysis of experiments*
in the Banff International Research Station in Canada.

They video all lectures, and make them available on the web.

Emlyn Williams learnt about this,
and watched the video of my lecture.

This motivated him to re-run that computer search from the 1970s with a more up-to-date version of his search program on a more up-to date computer.

Personal communication from Emlyn Williams

I gave a talk about these designs in August 2017
at the meeting on *Latest advances in the theory and applications of design and analysis of experiments*
in the Banff International Research Station in Canada.

They video all lectures, and make them available on the web.

Emlyn Williams learnt about this,
and watched the video of my lecture.

This motivated him to re-run that computer search from the 1970s with a more up-to-date version of his search program on a more up-to date computer.

Thus he found resolvable designs for 36 varieties in up to eight replicates of blocks of size six.

All concurrences are in $\{0, 1, 2\}$.

Personal communication from Emlyn Williams

I gave a talk about these designs in August 2017
at the meeting on *Latest advances in the theory and applications of design and analysis of experiments*
in the Banff International Research Station in Canada.

They video all lectures, and make them available on the web.

Emlyn Williams learnt about this,
and watched the video of my lecture.

This motivated him to re-run that computer search from the 1970s with a more up-to-date version of his search program on a more up-to date computer.

Thus he found resolvable designs for 36 varieties in up to eight replicates of blocks of size six.

All concurrences are in $\{0, 1, 2\}$.

He emailed me these results in September 2017.

Two resolvable designs with $v = 36$, $k = 6$, $r = 8$ and $b = 48$

These two designs have exactly the same value of $\bar{V}$,
so they are equally good.

Two resolvable designs with $v = 36$, $k = 6$, $r = 8$ and $b = 48$

These two designs have exactly the same value of $\bar{V}$, so they are equally good.

They are not isomorphic.
Our design has an automorphism group of order $2 \times 6! = 1440$ while Emlyn Williams' design has no automorphisms other than the identity.

Two resolvable designs with $v = 36$, $k = 6$, $r = 8$ and $b = 48$

These two designs have exactly the same value of $\bar{V}$,
so they are equally good.

They are not isomorphic.
Our design has an automorphism group of order $2 \times 6! = 1440$
while Emlyn Williams' design has no automorphisms other than the identity.

But there is a permutation of the varieties taking one concurrence matrix to the other,
which explains why they have exactly the same value of $\bar{V}$.

Case 3

There are two or more systems of blocks.

An example of a two-phase experiment

The treatments are 10 varieties of common beans.

An example of a two-phase experiment

The treatments are 10 varieties of common beans.

In Phase I, these are grown in a field, in 10 blocks of size 6.

An example of a two-phase experiment

The treatments are 10 varieties of common beans.

In Phase I, these are grown in a field, in 10 blocks of size 6.

In Phase II, a sample of beans is taken from each plot. Each sample is cooked in a special machine. The measured response is the time taken to properly cook the beans.

An example of a two-phase experiment

The treatments are 10 varieties of common beans.

In Phase I, these are grown in a field, in 10 blocks of size 6.

In Phase II, a sample of beans is taken from each plot.
Each sample is cooked in a special machine. The measured response is the time taken to properly cook the beans.

In Phase II, only four samples can be processed per day.
So we should treat days as 15 blocks of size 4.

An example of a two-phase experiment

The treatments are 10 varieties of common beans.

In Phase I, these are grown in a field, in 10 blocks of size 6.

In Phase II, a sample of beans is taken from each plot. Each sample is cooked in a special machine. The measured response is the time taken to properly cook the beans.

In Phase II, only four samples can be processed per day. So we should treat days as 15 blocks of size 4.

Now the design consists of one function allocating bean varieties to plots in the field, and another function allocating each plot to a run of the cooking machine.

Model when there are two systems of blocks

We measure the response Y on each sample.

If that sample is from a plot in block m with treatment i in Phase I and it is allocated to day n in Phase II, then we assume that

$$Y = \tau_i + \beta_m + \gamma_n + \text{random noise}.$$

To get rid of the β parameters and the γ parameters, we look at $(I - P_*)Y$, where P_* is the $N \times N$ matrix of orthogonal projection onto the space spanned by the characteristic vectors of the blocks in Phase I and the characteristic vectors of the days in Phase II.

Model when there are two systems of blocks

We measure the response Y on each sample.

If that sample is from a plot in block m with treatment i in Phase I and it is allocated to day n in Phase II, then we assume that

$$Y = \tau_i + \beta_m + \gamma_n + \text{random noise.}$$

To get rid of the β parameters and the γ parameters, we look at $(I - P_*)Y$, where P_* is the $N \times N$ matrix of orthogonal projection onto the space spanned by the characteristic vectors of the blocks in Phase I and the characteristic vectors of the days in Phase II.

Let X be the $N \times v$ incidence matrix of treatments in experimental units.
The information matrix is $X^\top (I - P_*) X$.

Computer search

At a conference on variety-testing in Słupia Wielka, Poland, in June 2018, Nha Vo-Thanh (Universität Hohenheim) gave a talk explaining his work with Hans-Peter Piepho on several different methods of computer search to find a good design for this experiment.

Computer search

At a conference on variety-testing in Słupia Wielka, Poland, in June 2018, Nha Vo-Thanh (Universität Hohenheim) gave a talk explaining his work with Hans-Peter Piepho on several different methods of computer search to find a good design for this experiment.

That evening, I got out some paper and a pen, and scribbled down some ideas, using my pattern approach. Very soon, I had a design with a smaller value of $\bar{V}$ than he had found.

Principle: Consider the smaller blocks first

The blocks in Phase II are smaller than those in Phase I,
so they will have more effect on increasing the variance.
So it makes sense to consider the design for Phase II first.

Principle: Consider the smaller blocks first

The blocks in Phase II are smaller than those in Phase I, so they will have more effect on increasing the variance. So it makes sense to consider the design for Phase II first.

There are 10 treatments in 15 blocks of size 4.
Think of the treatments as all pairs from $\{1, 2, 3, 4, 5\}$.
An obvious way to make 15 blocks of size 4 is to use the 4-cycles in the complete graph K_5 on 5 vertices.

Principle: Consider the smaller blocks first

The blocks in Phase II are smaller than those in Phase I, so they will have more effect on increasing the variance. So it makes sense to consider the design for Phase II first.

There are 10 treatments in 15 blocks of size 4.
Think of the treatments as all pairs from $\{1, 2, 3, 4, 5\}$.
An obvious way to make 15 blocks of size 4 is to use the 4-cycles in the complete graph K_5 on 5 vertices.

In fact, this design is balanced (all concurrences are 2), so it is best possible for Phase II.

Principle: Consider the smaller blocks first

The blocks in Phase II are smaller than those in Phase I, so they will have more effect on increasing the variance. So it makes sense to consider the design for Phase II first.

There are 10 treatments in 15 blocks of size 4. Think of the treatments as all pairs from $\{1, 2, 3, 4, 5\}$. An obvious way to make 15 blocks of size 4 is to use the 4-cycles in the complete graph K_5 on 5 vertices.

In fact, this design is balanced (all concurrences are 2), so it is best possible for Phase II.

A	B	C	E	F	G	H	D	B	D	A	F	A	E	B
E	H	F	H	J	I	J	I	C	J	C	G	D	G	I
C	A	B	G	E	F	B	C	I	A	G	C	E	B	A
H	F	E	J	I	H	D	H	J	F	J	D	I	D	G

Principle: Consider the smaller blocks first

The blocks in Phase II are smaller than those in Phase I, so they will have more effect on increasing the variance. So it makes sense to consider the design for Phase II first.

There are 10 treatments in 15 blocks of size 4.
Think of the treatments as all pairs from $\{1, 2, 3, 4, 5\}$.
An obvious way to make 15 blocks of size 4 is to use the 4-cycles in the complete graph K_5 on 5 vertices.

In fact, this design is balanced (all concurrences are 2), so it is best possible for Phase II.

A	B	C	E	F	G	H	D	B	D	A	F	A	E	B
E	H	F	H	J	I	J	I	C	J	C	G	D	G	I
C	A	B	G	E	F	B	C	I	A	G	C	E	B	A
H	F	E	J	I	H	D	H	J	F	J	D	I	D	G

Principle: Consider the smaller blocks first

The blocks in Phase II are smaller than those in Phase I,
so they will have more effect on increasing the variance.
So it makes sense to consider the design for Phase II first.

There are 10 treatments in 15 blocks of size 4.
Think of the treatments as all pairs from $\{1, 2, 3, 4, 5\}$.
An obvious way to make 15 blocks of size 4 is to use the 4-cycles in the complete graph K_5 on 5 vertices.

In fact, this design is balanced (all concurrences are 2),
so it is best possible for Phase II.

A	B	C	E	F	G	H	D	B	D	A	F	A	E	B
E	H	F	H	J	I	J	I	C	J	C	G	D	G	I
C	A	B	G	E	F	B	C	I	A	G	C	E	B	A
H	F	E	J	I	H	D	H	J	F	J	D	I	D	G

The non-intuitive step

The Phase II design has the property that we can group its days into five groups of three days, in such a way that every treatment in a group occurs twice in that group.

The non-intuitive step

The Phase II design has the property that we can group its days into five groups of three days, in such a way that every treatment in a group occurs twice in that group.

Arrange each group as a $(2 \times 3)/2$ rectangle,
in such a way that days are columns
and each treatment in the group occurs in both rows.

The non-intuitive step

The Phase II design has the property that we can group its days into five groups of three days, in such a way that every treatment in a group occurs twice in that group.

Arrange each group as a $(2 \times 3)/2$ rectangle,
in such a way that days are columns
and each treatment in the group occurs in both rows.

A	B	C	E	F	G	H	D	B	D	A	F	A	E	B
E	H	F	H	J	I	J	I	C	J	C	G	D	G	I
C	A	B	G	E	F	B	C	I	A	G	C	E	B	A
H	F	E	J	I	H	D	H	J	F	J	D	I	D	G

The non-intuitive step

The Phase II design has the property that we can group its days into five groups of three days, in such a way that every treatment in a group occurs twice in that group.

Arrange each group as a $(2 \times 3)/2$ rectangle,
in such a way that days are columns
and each treatment in the group occurs in both rows.

A	*B*	*C*	*E*	*F*	*G*	*H*	*D*	*B*	*D*	*A*	*F*	*A*	*E*	*B*
E	*H*	*F*	*H*	*J*	*I*	*J*	*I*	*C*	*J*	*C*	*G*	*D*	*G*	*I*
C	*A*	*B*	*G*	*E*	*F*	*B*	*C*	*I*	*A*	*G*	*C*	*E*	*B*	*A*
H	*F*	*E*	*J*	*I*	*H*	*D*	*H*	*J*	*F*	*J*	*D*	*I*	*D*	*G*

Use each row as a field block in Phase I.

The non-intuitive step

The Phase II design has the property that we can group its days into five groups of three days, in such a way that every treatment in a group occurs twice in that group.

Arrange each group as a $(2 \times 3)/2$ rectangle, in such a way that days are columns and each treatment in the group occurs in both rows.

A	B	C	E	F	G	H	D	B	D	A	F	A	E	B
E	H	F	H	J	I	J	I	C	J	C	G	D	G	I
C	A	B	G	E	F	B	C	I	A	G	C	E	B	A
H	F	E	J	I	H	D	H	J	F	J	D	I	D	G

Use each row as a field block in Phase I.
The treatment information lost to field blocks is the same as the information lost to rectangles, which is part of the information already lost to days, so no further information is lost in Phase I.

A surprising theorem

Theorem

In a nested row-column design,
if the rows within each rectangle have exactly the same treatments
then the loss of information on treatment differences is the same
as it is in the block design obtained by ignoring rectangles and rows.

A surprising theorem

Theorem

*In a nested row-column design,
if the rows within each rectangle have exactly the same treatments
then the loss of information on treatment differences is the same
as it is in the block design obtained by ignoring rectangles and rows.*

In this example, the best design for Phase I alone cannot be arranged as a nested row-column design with this property.

Comparison of designs

Design	computer search	patterns
A	0.80896	0.83333

So how did I spot that grouping?

If you take a BIBD for 10 treatments in 15 blocks of size 4 off the shelf, it may not be easy to find that rearrangement in five rectangles.

So how did I spot that grouping?

If you take a BIBD for 10 treatments in 15 blocks of size 4 off the shelf, it may not be easy to find that rearrangement in five rectangles.

The pattern approach suggests making one rectangle by using the six pairs which avoid 5.

If you take a BIBD for 10 treatments in 15 blocks of size 4 off the shelf, it may not be easy to find that rearrangement in five rectangles.

The pattern approach suggests making one rectangle by using the six pairs which avoid 5.

Two possibilities come to mind immediately.

12	13	14		12	13	14
34	24	23		23	34	24
14	12	13	and	14	12	13
23	34	24		34	24	23

How do those guys talk to each other?

Theorems

How do those guys talk to each other?

Theorems

↓

Folklore

How do those guys talk to each other?

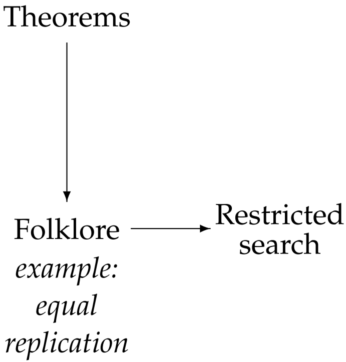
Theorems

↓

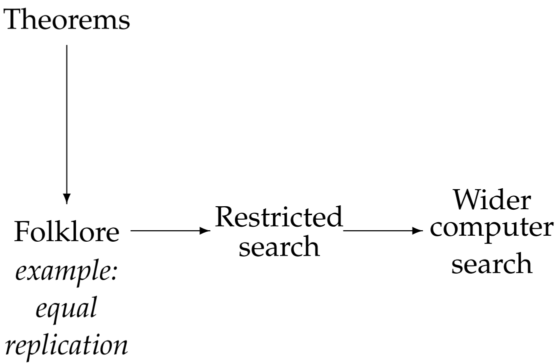
Folklore

example:
equal
replication

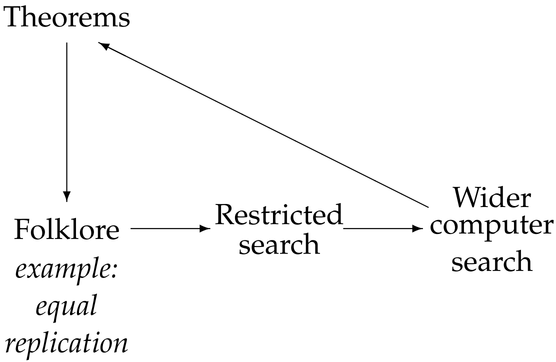
How do those guys talk to each other?



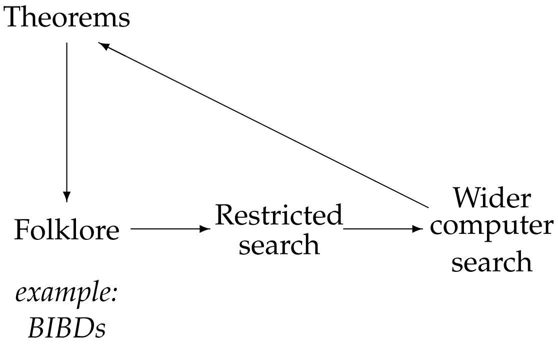
How do those guys talk to each other?



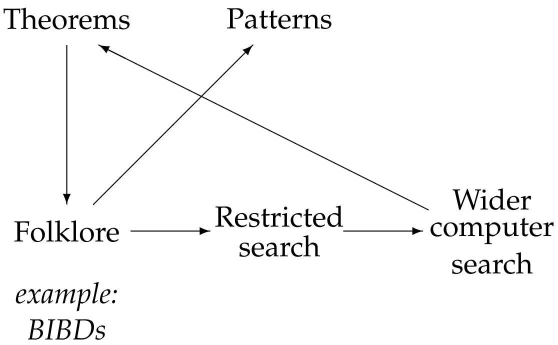
How do those guys talk to each other?



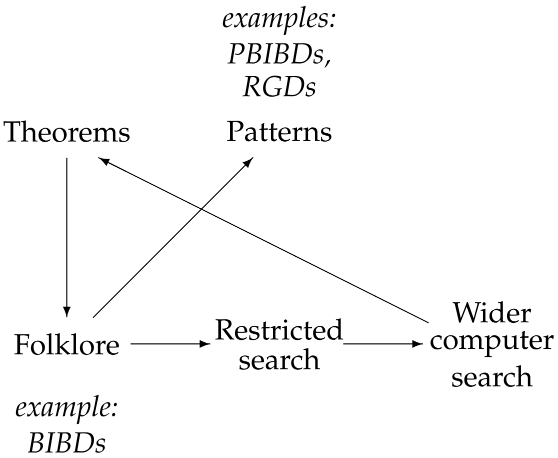
How do those guys talk to each other?



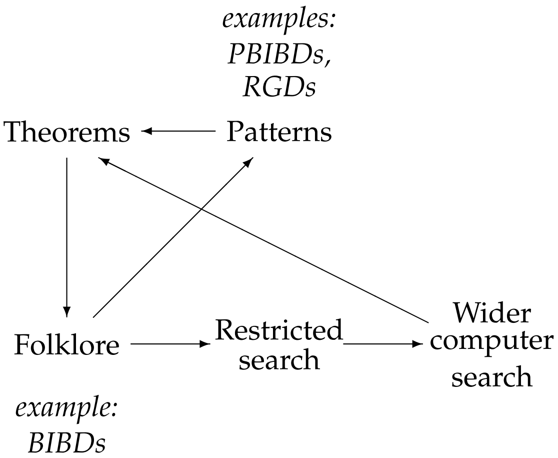
How do those guys talk to each other?



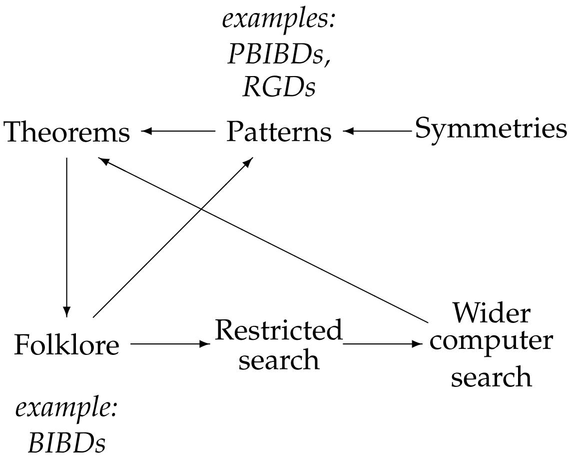
How do those guys talk to each other?



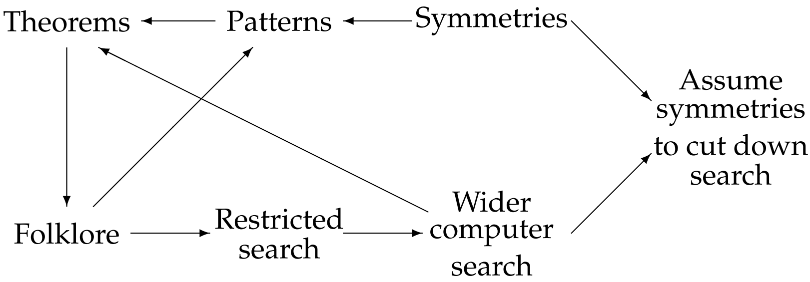
How do those guys talk to each other?



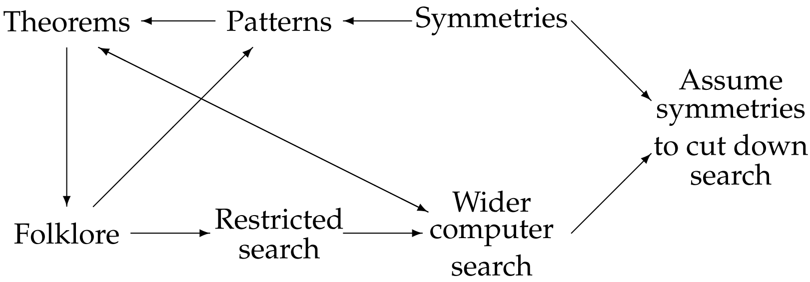
How do those guys talk to each other?



How do those guys talk to each other?



How do those guys talk to each other?



After I showed my design to NVT and HPP, they adapted their search method to incorporate that theorem

Conclusion

So—good luck with your search for good designs!

Which method will you use?