CS 147:
Computer Systems Performance Analysis
Higher Designs and Other Considerations

CS 147:
Computer Systems Performance Analysis
Higher Designs and Other Considerations

Overview

Larger Designs
- Price of More Levels
- Extending Confounding
- Fractionating Using Confounding Algebra
- Example
- Higher and Mixed Levels

Block Designs

Informal Methods

Other Considerations
- Record-Keeping
- Randomization of Experimental Order
- Digression on PRNGs
- Types of Randomization

Overview

Larger Designs
Price of More Levels
Extending Confounding
Fractionating Using Confounding Algebra
Example
Higher and Mixed Levels
Block Designs
Informal Methods
Other Considerations
Record-Keeping
Randomization of Experimental Order
Digression on PRNGs
Types of Randomization

The Price of More Levels

- Using more levels increases no. of runs
  - 50% for raising a single parameter from 2 to 3
  - 125% for two parameters
- Extra runs could be used in other ways
  - Examine more parameters
  - Reduce variance (more replications)
- Extra levels complicate experimentation

The Price of More Levels

- Using more levels increases no. of runs
  - 50% for raising a single parameter from 2 to 3
  - 125% for two parameters
- Extra runs could be used in other ways
  - Examine more parameters
  - Reduce variance (more replications)
- Extra levels complicate experimentation

Deciding to Use More Levels

- Must balance cost of extra runs against extra information gained
- Is response likely to be nonlinear?
- Are extreme responses sufficient?
- Does curve have minimum/maximum between extremes?

Deciding to Use More Levels

- Must balance cost of extra runs against extra information gained
- Is response likely to be nonlinear?
- Are extreme responses sufficient?
- Does curve have minimum/maximum between extremes?

Extending the Algebra of Confounding

- Standard 2-level confounding algebra is based on exponentiation modulo 2: $A^2 = A^0 = I$
- This is trivial to extend to level n: $A^x A^y = A^{(x+y) \bmod n}$

Extending the Algebra of Confounding

- Standard 2-level confounding algebra is based on exponentiation modulo 2: $A^2 = A^0 = I$
- This is trivial to extend to level n: $A^x A^y = A^{(x+y) \bmod n}$

Rules of Extended Confounding Algebra

- First letter should have unit exponent
  - Achieved by raising to powers
  - Example (mod 3): $A^2B = (A^2B)^2 = A^4B^2 = AB^2$
- This works because of fractionating method
  - Sum of exponents modulo n is constant

Rules of Extended Confounding Algebra

- First letter should have unit exponent
  - Achieved by raising to powers
  - Example (mod 3): $A^2B = (A^2B)^2 = A^4B^2 = AB^2$
- This works because of fractionating method
  - Sum of exponents modulo n is constant

Fractionating Using Confounding Algebra

- Choose a confounding
- Divide experiment into blocks based on confounding
- Choose a particular block at random
- Execute experiments in random order

Fractionating Using Confounding Algebra

- Choose a confounding
- Divide experiment into blocks based on confounding
- Choose a particular block at random
- Execute experiments in random order

Using Confounding to Create 3^{k-p} Blocks

- Levels of each factor chosen from $\{0, 1, 2\}$
- Confounding exponents indicate multipliers
- Sum of multiplied levels modulo 3^p gives block number
- Example: AB^2 converts to $a + 2b \mod 3 = i$ where i is block number

Using Confounding to Create 3^{k-p} Blocks

- Levels of each factor chosen from $\{0, 1, 2\}$
- Confounding exponents indicate multipliers
- Sum of multiplied levels modulo 3^p gives block number
- Example: AB^2 converts to $a + 2b \mod 3 = i$ where i is block number

Example of a 3^{2-1} Fraction

- Choose confounding: $I = AB^2$
- Divide combinations into blocks:

a	b	$a + 2b \mod 3$
0	0	0
0	1	2
0	2	1
1	0	1
1	1	0
1	2	2
2	0	2
2	1	1
2	2	0

Example of a 3^{2-1} Fraction

- Choose confounding: $I = AB^2$
- Divide combinations into blocks:

a	b	$a + 2b \mod 3$
0	0	0
0	1	2
0	2	1
1	0	1
1	1	0
1	2	2
2	0	2
2	1	1
2	2	0

Running the 3^{2-1} Fraction

- Choose block 2 by rolling dice
- Run combinations (a,b) = (0,1);(1,2);(2,0)
- Complete confoundings:
  - $I = AB^2$
  - $A = AB = B$
- Calculating effects is beyond scope of lecture
  - Even in this simple case

Running the 3^{2-1} Fraction

- Choose block 2 by rolling dice
- Run combinations (a,b) = (0,1);(1,2);(2,0)
- Complete confoundings:
  - $I = AB^2$
  - $A = AB = B$
- Calculating effects is beyond scope of lecture
  - Even in this simple case

Higher Prime Levels

- Same algebra can be used for higher numbers of levels
  - So long as prime
- Complexity rapidly becomes prohibitive
  - Normally use computers to do hard stuff
- Often simpler to use 2-level experiments
  - Figure out which effects are major
  - Then use 1-factor tests to examine closely

Higher Prime Levels

- Same algebra can be used for higher numbers of levels
  - So long as prime
- Complexity rapidly becomes prohibitive
  - Normally use computers to do hard stuff
- Often simpler to use 2-level experiments
  - Figure out which effects are major
  - Then use 1-factor tests to examine closely

Mixed Levels

- Factors may have different numbers of levels
  - E.g., 2 CPUs, 3 disk drives, 4 memory sizes
- Possible to do fractional experiments here, too
  - Complexity is remarkable
  - No simple way to select fraction
  - Consult catalogs or software for fraction tables

Mixed Levels

- Factors may have different numbers of levels
  - E.g., 2 CPUs, 3 disk drives, 4 memory sizes
- Possible to do fractional experiments here, too
  - Complexity is remarkable
  - No simple way to select fraction
  - Consult catalogs or software for fraction tables

Block Designs

- Causes of blocking
- Example of blocking
- Confounding between blocks
- Special types of blocks

Block Designs
- Causes of blocking
- Example of blocking
- Confounding between blocks
- Special types of blocks

Causes of Blocking

- Physical constraints
  - n probes, $m > n$ signals to measure
- Time constraints
  - Only n experiments per day, with other activity between
- Subject constraints
  - Need to install new hardware between runs
  - Multiple disks on each machine

Causes of Blocking

- Physical constraints
  - n probes, $m > n$ signals to measure
- Time constraints
  - Only n experiments per day, with other activity between
- Subject constraints
  - Need to install new hardware between runs
  - Multiple disks on each machine

Example of Blocking

- Need to run 2 benchmarks under 2 network loads
- Can only do 2 runs per day
- Other conditions may vary from day to day
- Which pair to do first?

Example of Blocking

- Need to run 2 benchmarks under 2 network loads
- Can only do 2 runs per day
- Other conditions may vary from day to day
- Which pair to do first?

Confounding Between Blocks

Example of the problem:

- Three different ways to divide runs:

Day 1:	00	01	00	10	00	11
Day 2:	10	11	01	11	10	01

- Each choice confounds something with the day effect:
  - Factor a (level takes a day to change)
  - Factor b (same)
  - Interaction (equal levels one day, unequal other)

Confounding Between Blocks

Example of the problem:
- Three different ways to divide runs:
Day 1: 00 01 | 00 10 | 00 11
Day 2: 10 11 | 01 11 | 10 01
- Each choice confounds something with the day effect:
  - Factor a (level takes a day to change)
  - Factor b (same)
  - Interaction (equal levels one day, unequal other)

Special Types of Blocks

- Split plot
  - Group by one factor, vary others randomly
  - Useful when expensive to change that factor
- Nested or hierarchical
  - One factor varies within another
  - E.g., 5 computers, each with one NIC from each of 3 manufacturers
- ANOVA is different here: watch out!

Special Types of Blocks

- Split plot
  - Group by one factor, vary others randomly
  - Useful when expensive to change that factor
- Nested or hierarchical
  - One factor varies within another
  - E.g., 5 computers, each with one NIC from each of 3 manufacturers
- ANOVA is different here: watch out!

Informal Methods

- Often only want best performance
- Can simply pick combination that does best
- Better choice: sort by performance
  - Identify which factors are common to top entries
  - Eliminate any that aren't consistent

Informal Methods
- Often only want best performance
- Can simply pick combination that does best
- Better choice: sort by performance
  - Identify which factors are common to top entries
  - Eliminate any that aren't consistent

Record-Keeping Principles

- Never throw away data
- Be able to reproduce any experiment
- Parameterize your software
  - Don't create experiments by editing source
  - Leads to irreproducibility
- Use version control!

Record-Keeping Principles

- Never throw away data
- Be able to reproduce any experiment
- Parameterize your software
  - Don't create experiments by editing source
  - Leads to irreproducibility
- Use version control!

The Need for Randomization

- Uncontrollable parameters may vary during experimentation
- Plotting error vs. experiment number detects systematic variation
  - But doesn't control it
- Randomization controls the problem
  - Turns it into error parameter

The Need for Randomization
- Uncontrollable parameters may vary during experimentation
- Plotting error vs. experiment number detects systematic variation
  - But doesn't control it
- Randomization controls the problem
  - Turns it into error parameter

Example of External Trends

- Consider measuring disk performance:
  - Benchmark creates 1000 small files, 10 large ones, writes them, then deletes them
  - File size is varied as experimental parameter
  - One run takes several hours
  - Other people use system daily

Example of External Trends
- Consider measuring disk performance:
  - Benchmark creates 1000 small files, 10 large ones, writes them, then deletes them
  - File size is varied as experimental parameter
  - One run takes several hours
  - Other people use system daily

Slide contains animations.

Example of External Trends

- Consider measuring disk performance:
  - Benchmark creates 1000 small files, 10 large ones, writes them, then deletes them
  - File size is varied as experimental parameter
  - One run takes several hours
  - Other people use system daily
- Disk fragmentation may increase over time, changing results

Example of External Trends

- Consider measuring disk performance:
  - Benchmark creates 1000 small files, 10 large ones, writes them, then deletes them
  - File size is varied as experimental parameter
  - One run takes several hours
  - Other people use system daily
- Disk fragmentation may increase over time, changing results

Slide contains animations.

(Pseudo-)Random Number Generation

- Not all PRNGs are equal
- Most common is linear congruential
  - E.g., `rand`, `random`, `drand48`
  - *Don't* use low bits (modulo) to adjust range
  - Best to use floating result in $[0, 1)$ and multiply by range
- Prefer longer periods
  - E.g., Mersenne Twist (see Google)

(Pseudo-)Random Number Generation

- Not all PRNGs are equal
- Most common is linear congruential
  - E.g., rand, random, drand48
  - Don't use low bits (modulo) to adjust range
  - Best to use floating result in [0, 1) and multiply by range
- Prefer longer periods
  - E.g., Mersenne Twist (see Google)

Random Seeding

- Risky to use "random" seeds (e.g., `/dev/random`)
  - Irreproducible (uncheckable) results
  - Risk of getting into middle of sequence from different experiment
    - Produces correlation
  - Especially with linear congruential
- Better to use different parts of long sequence
  - Note that many PRNGs only take a 32-bit seed
  - ⇒ Only 2^{32} different sequences
  - If period is 2^{32}, then you're always diving into the middle of a sequence
- In any case, remember your seeds

Random Seeding

- Risky to use "random" seeds (e.g., `/dev/random`)
  - Irreproducible (uncheckable) results
  - Risk of getting into middle of sequence from different experiment
    - Produces correlation
  - Especially with linear congruential
- Better to use different parts of long sequence
  - Note that many PRNGs only take a 32-bit seed
  - ⇒ Only 2^{32} different sequences
  - If period is 2^{32}, then you're always diving into the middle of a sequence
- In any case, remember your seeds

Complete Randomization

- Plan experiment first
  - Levels of each parameter
  - Number of replications
- List experiments by levels and replication number
- Choose experiments from list randomly
  - Use selection without replacement
  - Equivalently, shuffle list and use shuffled order

Complete Randomization

- Plan experiment first
  - Levels of each parameter
  - Number of replications
- List experiments by levels and replication number
- Choose experiments from list randomly
  - Use selection without replacement
  - Equivalently, shuffle list and use shuffled order

Constrained Randomization

- Complete randomization sometimes impossible
  - One experiment might destroy others
  - Lengthy setup times
  - Need to send computer back to manufacturer
- Must divide experiments into blocks
  - Randomize within each block
- Block effect confounded with true effect

Constrained Randomization

- Complete randomization sometimes impossible
  - One experiment might destroy others
  - Lengthy setup times
  - Need to send computer back to manufacturer
- Must divide experiments into blocks
  - Randomize within each block
- Block effect confounded with true effect