Introduction to Experiment Design

Refs: Chap 16-17 of Raj Jain’s book

Shiv Kalyanaraman
Rensselaer Polytechnic Institute
shivkuma@ecse.rpi.edu
http://www.ecse.rpi.edu/Homepages/shivkuma

Adapted from Prof. Raj Jain’s slides
Why Experiment design?

- Problem: validation and results only as good as your test cases!
- How to design a critical set of test cases?
- Idea: Parameterize and use black-box strategy

Parameters or Factors → System → Metrics

Rensselaer Polytechnic Institute

Shivkumar Kalyanaraman
Performance, Metrics and Parameters

- **Performance questions:**
  - **Absolute:** How fast does computer A run MY program?
  - **Relative:** Is machine A faster than machine B, and if so, how much faster?

- **Parameters:** factors or inputs
  - Eg: clock rate, poisson inter-arrivals

- **Metrics:** functions of factors or parameters
  - Eg: throughput, response time, queue length ...
  - Metric should characterize the design tradeoffs adequately
  - Metrics are usually functions of many factors. Use of one factor alone may be misleading.
Choose metrics to reflect appropriate tradeoffs

- **Network users**: services and performance that their applications need, e.g., guarantee that each message it sends will be delivered without error within a certain amount of time.

- **Network designers**: cost-effective design, e.g., that network resources are efficiently utilized and fairly allocated to different users.
  - Users + designer perspectives enough to meet 3 factors of interface/architecture design. But ...

- **Network providers**: system that is easy to administer and manage, e.g., that faults can be easily isolated and it is easy to account for usage.
  - Require management tools and interfaces.
  - Often considered to the basic protocol interface design.
Goals of Experiment Design

- Design a proper *set* of (simulation or measurement) experiments: maximize information gained with minimum experiments!
- Develop a *(regression)* model that best describes the data obtained
- Estimate the *contribution* of each factor and its values (i.e. alternatives) to the *performance variation*
- Isolate measurement *errors*
- Estimate *confidence intervals* for model *parameters*
- Check if the *alternatives* are significantly *different*
- Check if the model is *adequate*
Example: TCP/AQM Design Problem

- TCP versions: Reno or SACK
- Max Segment sizes: 100 B vs 1000 B
- Buffer Size: 10 pkts vs 100 pkts
- AQM: Drop tail vs RED vs REM
  - AQM parameters …
- Workload: FTP
- Configuration parameters: 3 flows vs 10 flows
- Metrics: total throughput (efficiency), C.o.V of per-flow throughputs (fairness)
Terminology

- **Response Variable**: Outcome. Eg: response time
- **Factors**: Variables that affect the response variable
  - Eg: buffer size, RED parameters
- **Levels**: The values that a factor can assume
  - Eg: TCP type has 2 levels: SACK or Reno
- **Primary Factors**: The “interesting” factors whose effects need to be quantified.
- **Secondary Factors**: Factors whose impact need not be quantified
  - Eg: We have excluded minor factors like receiver window size, or delay ack (whose effects we roughly know and/or consider minor). Some workload parameters also are secondary factors
- **Replication**: Repetition of all or some experiments
- **Design**: The number of experiments, the factor level, and number of replications for each experiment
  - Eg: Full Factorial Design with replications
- **Interaction**: Effect of one factor depends upon other factors!
Experiment Design Problem

- A system with \( k \) parameters and \( n_i \) level for parameter \( x_i \) and a performance metric \( f \)
- \( f = s(x_1, x_2, \ldots, x_k) \) can be evaluated by an experiment, i.e. simulation or measurement or analytical formula
- Find the effect of each parameter or parameter interaction on the system performance, the best parameter combination, etc.

Parameters or Factors → System → Metrics
Simple Experiment Design

- Start with a typical configuration
- Vary one parameter at a time
- Number of experiments:

\[ 1 + \sum_{i=1}^{k} n_i - 1 \]

- Disadvantage: only good for the system with *no interaction* between parameters
  - Wrong conclusions if the factors have interaction
- Not statistically efficient
Full factorial Design

- Try every possible combination of all the parameters.
- Number of experiments

\[ \prod_{i=1}^{k} n_i \]

- Disadvantage: too many experiments
Reduce experiment number?

- Reduce the number of levels for each parameter, e.g., reduce $n_i$ to 2 levels for each parameter
- Reduce the number of parameters
- Use *fractional factorial design*:
  - Try a subset of the all possible parameter combinations
  - Less information
  - May not get all interactions
  - Not a problem if negligible interactions
$2^k$ Full Factorial Design

- Using a *nonlinear* regression model (assuming 2 parameters, $x_1, x_2$)

\[ y = q_0 + q_1x_1 + q_2x_2 + q_3x_3 + q_{12}x_1x_2 + q_{23}x_2x_3 + q_{13}x_1x_3 + q_{123}x_1x_2x_3 \]

- Goal: Run $2^k$ experiment (e.g., take the *two extreme values* of each parameter) and solve the above equation for the regression parameters: $q_0, q_1, q_2, q_{12}, q_{23}, q_{13}, q_{123}$.

- Disadvantage: only good for systems where effects of parameters or interactions are monotonous
  - i.e., the system has to be consistent with the model
2\(^{k-p}\) Fractional Factorial Design

- Use a simplified regression model, ignore some interactions (especially high-order interactions since their effect are usually small)

- For example, 2\(^{3-1}\) design:

\[ y = q_0 + q_1 x_1 + q_2 x_2 + q_3 x_3 \]

All interactions are ignored, only 4 unknowns, only 2\(^{3-1}\) experiments are needed to solve them.
Simple Full Factorial Problem

Two factors, each at two levels.

<table>
<thead>
<tr>
<th>Cache Size</th>
<th>Memory Size 4M Bytes</th>
<th>Memory Size 16M Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1K</td>
<td>15</td>
<td>45</td>
</tr>
<tr>
<td>2K</td>
<td>25</td>
<td>75</td>
</tr>
</tbody>
</table>

$x_A = \begin{cases} -1 & \text{if 4M bytes memory} \\ 1 & \text{if 16M bytes memory} \end{cases}$

$x_B = \begin{cases} -1 & \text{if 1K bytes cache} \\ 1 & \text{if 2K bytes cache} \end{cases}$

Qn: What is the effect of memory & cache on workstation performance?
Underlying Regression Model

$$y = q_0 + q_A x_A + q_B x_B + q_{AB} x_A x_B$$

$$15 = q_0 - q_A - q_B + q_{AB}$$
$$45 = q_0 + q_A - q_B - q_{AB}$$
$$25 = q_0 - q_A + q_B - q_{AB}$$
$$75 = q_0 + q_A + q_B + q_{AB}$$

$$y = 40 + 20x_A + 10x_B + 5x_A x_B$$

- **Interpretation**: Mean performance = 40 MIPS
- Effect of memory = 20 MIPS
- Effect of cache = 10 MIPS
- **Interaction** between memory and cache = 5 MIPS
Computation of Effects (i.e. coeffs)

<table>
<thead>
<tr>
<th>Experiment</th>
<th>A</th>
<th>B</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>y_1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>-1</td>
<td>y_2</td>
</tr>
<tr>
<td>3</td>
<td>-1</td>
<td>1</td>
<td>y_3</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>y_4</td>
</tr>
</tbody>
</table>

Model:

\[ y = q_0 + q_A x_A + q_B x_B + q_{AB} x_A x_B \]

Substitution:

\[ y_1 = q_0 - q_A - q_B + q_{AB} \]
\[ y_2 = q_0 + q_A - q_B - q_{AB} \]
\[ y_3 = q_0 - q_A + q_B - q_{AB} \]
\[ y_4 = q_0 + q_A + q_B + q_{AB} \]
Solution:

\[ q_0 = \frac{1}{4}(y_1 + y_2 + y_3 + y_4) \]
\[ q_A = \frac{1}{4}(-y_1 + y_2 - y_3 + y_4) \]
\[ q_B = \frac{1}{4}(-y_1 - y_2 + y_3 + y_4) \]
\[ q_{AB} = \frac{1}{4}(y_1 - y_2 - y_3 + y_4) \]

Notice:

- $q_A = \text{Column } A \times \text{Column } y$
- $q_B = \text{Column } B \times \text{Column } y$
- $q_{AB} = \text{Column } A \times \text{Column } B \times \text{Column } y$

- Note: effects ($q_i$) are **linear** combinations of responses ($y_i$)
- Sum of the coefficients is zero: aka “contrasts”
- Note that the above equations can be expressed in terms of column A, B etc!
  => Sign table method!
Sign Table Method

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>AB</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td>15</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>45</td>
</tr>
<tr>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>25</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>75</td>
</tr>
<tr>
<td></td>
<td>160</td>
<td>80</td>
<td>40</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>20</td>
<td>10</td>
<td>5</td>
</tr>
</tbody>
</table>

- $q_0$
- $q_A$
- $q_B$
- $q_{AB}$

- Each column multiplies the responses ($y_i$);
- Sum the multiples, and
- Divide the sums by $2^k$
Allocation of Variation

- A.k.a. which factors and alternatives matter!
- Importance of a factor: proportion of variation explained by that factor
- Note: variation is not variance!

Sample Variance of \( y = s^2_y = \frac{\sum_{i=1}^{2^2} (y_i - \bar{y})^2}{2^2 - 1} \)

Variation of \( y \triangleq Numerator \)

\[ = \frac{2^2}{\sum_{i=1}^{2^2} (y_i - \bar{y})^2} \]

\[ = \text{sum of squares total (SST)} \]
Allocation of Variation: $2^2$ design

For a $2^2$ design:

$$\text{SST} = 2^2 q_A^2 + 2^2 q_B^2 + 2^2 q_{AB}^2$$

Variation due to $A = \text{SSA} = 2^2 q_A^2$

Variation due to $B = \text{SSB} = 2^2 q_B^2$

Variation due to interaction $= \text{SSAB} = 2^2 q_{AB}^2$

$$\text{SST} = \text{SSA} + \text{SSB} + \text{SSAB}$$

Fraction explained by $A = \frac{\text{SSA}}{\text{SST}}$

- Fractions explained by $B$ and interaction between $A$ & $B$ can be obtained similarly
Allocation of Variation: Example 2² design

Memory-cache study:

\[ \bar{y} = \frac{1}{4}(15 + 55 + 25 + 75) = 40 \]

Total Variation

\[ \sum_{i=1}^{4} (y_i - \bar{y})^2 \]

\[ = (25^2 + 15^2 + 15^2 + 35^2) \]

\[ = 2100 \]

\[ = 4 \times 20^2 + 4 \times 10^2 + 4 \times 5^2 \]

Total variation = 2100
Variation due to Memory = 1600 (76%)
Variation due to cache = 400 (19%)
Variation due to interaction = 100 (5%)

Note: A is most imppt factor!
General $2^k$ Factorial Designs

$k$ factors at two levels each.
$2^k$ experiments.
$2^k$ effects:
- $k$ main effects
- $\binom{k}{2}$ two factor interactions
- $\binom{k}{3}$ three factor interactions...

**EXAMPLE**

<table>
<thead>
<tr>
<th>Factor</th>
<th>Level -1</th>
<th>Level 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A$ Memory Size</td>
<td>4MB</td>
<td>16MB</td>
</tr>
<tr>
<td>$B$ Cache Size</td>
<td>1kB</td>
<td>2kB</td>
</tr>
<tr>
<td>$C$ Number of Processors</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>
### 2\(^k\) Factorial Design Example

#### 2\(^k\) Design Example (cont)

<table>
<thead>
<tr>
<th>Cache Size</th>
<th>4M Bytes</th>
<th></th>
<th></th>
<th>16M Bytes</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 Proc</td>
<td>2 Proc</td>
<td></td>
<td>1 Proc</td>
<td>2 Proc</td>
<td></td>
</tr>
<tr>
<td>1K Byte</td>
<td>14</td>
<td>46</td>
<td></td>
<td>22</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>2K Byte</td>
<td>10</td>
<td>50</td>
<td></td>
<td>34</td>
<td>86</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>I</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>AB</th>
<th>AC</th>
<th>BC</th>
<th>ABC</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>14</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>22</td>
</tr>
<tr>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>34</td>
</tr>
<tr>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td>46</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>58</td>
</tr>
<tr>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>50</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>86</td>
</tr>
</tbody>
</table>

| Total/8 | 320 | 80  | 40  | 160 | 40  | 16  | 24   | 9   |
| Total/8 | 40  | 10  | 5   | 20  | 5   | 2   | 3    | 1   |
Allocation of Variation

$$SST = 2^3(q_A^2 + q_B^2 + q_C^2 + q_{AB}^2 + q_{AC}^2 + q_{BC}^2 + q_{ABC}^2)$$
$$= 8(10^2 + 5^2 + 20^2 + 5^2 + 2^2 + 3^2 + 1^2)$$
$$= 800 + 200 + 3200 + 200 + 32 + 72 + 8 = 4512$$
$$= 18\% + 4\% + 71\% + 4\% + 1\% + 2\% + 0\%$$
$$= 100\%$$

- Factor C (i.e. number of processors) is the most important (accounts for 71% of performance variation!)
Exercise

Analyze the $2^3$ design:

<table>
<thead>
<tr>
<th></th>
<th>$A_1$</th>
<th>$A_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_1$</td>
<td>100</td>
<td>120</td>
</tr>
<tr>
<td>$C_2$</td>
<td>15</td>
<td>10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>$B_1$</th>
<th>$B_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_1$</td>
<td>40</td>
<td>20</td>
</tr>
<tr>
<td>$C_2$</td>
<td>30</td>
<td>50</td>
</tr>
</tbody>
</table>

a. Quantify main effects and all interactions.
b. Quantify percentages of variation explained.
c. Sort the variables in the order of decreasing importance.
Experiment Designer Package

- Run $2^k$ full factorial or $2^{(k-p)}$ fractional factorial design
- Executable: edesign –s <configuration_file>
- Experiment results are in designer.res
Example Configuration File

ScriptFile  my_script.tcl
LogOn       y
Designer Factorial
P         0

[parameters]
#name  min  max
x1   -2     2
x2   -2     2

[/parameters]
Example Configuration File

Farmer localhost:6666:farmer:any
Worker localhost:worker
[deployer]
./get_effects.pl designer.res
[/deployer]
Example Experiment Script

```bash
#SETPARAMETERS
return_result [expr $x1*$x1+$x2*$x2]
```

Original Experiment Script

Generate a set of parameter values

Experiment Script

Return experiment result

Farmer-worker System
Example Output (design.res)

- Format of designer.res
  - 1st parameter, 2nd parameter, ..., metric

- Example
  - -1,-1,-1,-1,-0.999
  - 1,-1,-1,-1,1.001
  - -1,1,-1,-1,1.001

- get-effects.pl will process this file and output the effect and variation percentage of each parameter (interaction).
Example get-effects.pl Output

The parameter index represents:

<table>
<thead>
<tr>
<th>Parameter(s)</th>
<th>Effect</th>
<th>Variation Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td>40</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>20</td>
<td>76.19%</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>19.05%</td>
</tr>
<tr>
<td>1 2</td>
<td>5</td>
<td>4.76%</td>
</tr>
</tbody>
</table>
What you need to do?

- Finish the tcl script
- Open a terminal window, run `start_system.sh` (don’t repeat even you need to do multiple edesign)
- Open another terminal window, run:
  `edesign -s ex_ff.conf`
- Sit and wait to see the output of `get-effects.pl`
- Any problem, ask Yong or me.
Experiment Designer Package

- Run $2^k$ full factorial or $2^{(k-p)}$ fractional factorial design
- Executable: edesign <configuration_file>
- Experiment results are in designer.res
Example Configuration File

- ExpCountLimit 100
- Experiment ns my.script
- LogOn y
- Designer Factorial
Example Configuration File

- [parameters]
- #name  min  max  default  step  is_int?  to be tuned?
  - x1   -2   2   0   0.5  n  y
  - x2   -2   2   -1  0.5  n  y
- [/parameters]
Example

\[ y = (q_0 + q_{123}x_1x_2x_3) + (q_1x_1 + q_{23}x_2x_3) + (q_2x_2 + q_{13}x_1x_3) + (q_3x_3 + q_{12}x_1x_2) \]

Let \[ x_1x_2x_3 = 1 \Rightarrow x_3 = x_1x_2 \Rightarrow x_1 = x_2x_3 \Rightarrow x_2 = x_1x_3 \]

Then \[ y = q'_0 + q'_1x_1 + q'_2x_2 + q'_3x_3 \]

\[ q'_0 = q_{123} \]

is called confounding of effects
Design with Sign Table

- First create a $2^3$ sign table, replace $x_{12}$ with $x_3$,

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>-1</th>
<th>+1</th>
<th>-1</th>
<th>+1</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_2$</td>
<td>-1</td>
<td>-1</td>
<td>+1</td>
<td>+1</td>
</tr>
<tr>
<td>$x_3(x_1x_2)$</td>
<td>+1</td>
<td>-1</td>
<td>-1</td>
<td>+1</td>
</tr>
</tbody>
</table>