Statistical Analysis of Output from Terminating Simulations

Chapter 6
What We’ll Do ...

- Time frame of simulations
- Strategy for data collection and analysis
- Confidence intervals
- Comparing two scenarios
- Comparing many scenarios via Arena Process Analyzer (PAN)
- Searching for an optimal scenario with OptQuest
Motivation

- Random input leads to random output (RIRO)
- Run a simulation (once) — what does it mean?
  - Was this run “typical” or not?
  - Variability from run to run (of the same model)?
- Need statistical analysis of output data
  - From a single model configuration
  - Compare two or more different configurations
  - Search for an optimal configuration
- Statistical analysis of output is often ignored
  - This is a big mistake – no idea of precision of results
  - Not hard or time-consuming to do this – it just takes a little planning and thought, then some (cheap) computer time
Time Frame of Simulations

- **Terminating**: Specific starting, stopping conditions
  - Run length will be well-defined (and finite)
- **Steady-state**: Long-run (technically forever)
  - Theoretically, initial conditions don’t matter
    - But practically, they usually do
  - Not clear how to terminate a run
- This is really a question of intent of study
- Has major impact on how output analysis is done
- Sometimes it’s not clear which is appropriate
- Here: Terminating (steady-state in Section 7.2)
Strategy for Data Collection and Analysis

- For terminating case, make IID replications
  - *Run > Setup > Replication Parameters:* Number of Replications field
  - Check both boxes for Initialize Between Replications

- Separate results for each replication – Category by Replication report
  - Model 5-3, but for 10 replications (= Model 6-1)

<table>
<thead>
<tr>
<th>Replication</th>
<th>Total Cost ($)</th>
<th>Percent Rejected</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>22,385.64</td>
<td>12.2759</td>
</tr>
<tr>
<td>2</td>
<td>20,612.12</td>
<td>11.6059</td>
</tr>
<tr>
<td>3</td>
<td>23,837.38</td>
<td>10.4558</td>
</tr>
<tr>
<td>4</td>
<td>21,915.24</td>
<td>11.9110</td>
</tr>
<tr>
<td>5</td>
<td>22,462.34</td>
<td>13.5546</td>
</tr>
<tr>
<td>6</td>
<td>20,573.78</td>
<td>10.9804</td>
</tr>
<tr>
<td>7</td>
<td>20,935.88</td>
<td>10.1093</td>
</tr>
<tr>
<td>8</td>
<td>22,078.91</td>
<td>9.4256</td>
</tr>
<tr>
<td>9</td>
<td>20,056.75</td>
<td>9.4972</td>
</tr>
<tr>
<td>10</td>
<td>21,325.23</td>
<td>11.3388</td>
</tr>
</tbody>
</table>

Note cross-replication variability
Strategy for Data Collection and Analysis (cont’d.)

- Category Overview report has some statistical-analysis results of output across replications
- How many replications?
  - Trial and error (now)
  - Approximate number for acceptable precision (below)
  - Sequential sampling (Chapter 12)
- Turn off animation altogether for max speed
  - Run > Run Control > Batch Run (No Animation)
Confidence Intervals for Terminating Systems

- Using formulas in Chapter 2, viewing cross-replication summary outputs as basic data:

<table>
<thead>
<tr>
<th></th>
<th>Total Cost ($)</th>
<th>Percent Rejected</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sample Mean</td>
<td>21,618.33</td>
<td>11.12</td>
</tr>
<tr>
<td>Sample Standard Deviation</td>
<td>1,136.24</td>
<td>1.30</td>
</tr>
<tr>
<td>95% Confidence Interval Half Width</td>
<td>812.82</td>
<td>0.93</td>
</tr>
<tr>
<td>Minimum Summary Output Value</td>
<td>20,056.75</td>
<td>9.43</td>
</tr>
<tr>
<td>Maximum Summary Output Value</td>
<td>23,837.38</td>
<td>13.55</td>
</tr>
</tbody>
</table>

- Possibly most useful part: 95% confidence interval on expected values
- This information (except standard deviation) is in Category Overview report
  - If > 1 replication, Arena uses cross-repl. data as above
  - Other confidence levels, graphics – Output Analyzer

Half Width, Number of Replications

- Prefer smaller confidence intervals — precision

- Notation:
  
  \[ n = \text{no. replications} \]
  \[ \bar{X} = \text{sample mean} \]
  \[ s = \text{sample standard deviation} \]
  \[ t_{n-1,\alpha/2} = \text{critical value from } t \text{ tables} \]

- Confidence interval:
  \[ \bar{X} \pm t_{n-1,\alpha/2} \frac{s}{\sqrt{n}} \]

- Half-width =
  \[ t_{n-1,\alpha/2} \frac{s}{\sqrt{n}} \]

- Can’t control \( t_{n-1,\alpha/2} \) or \( s \)

- Must increase \( n \) — how much?

Want this to be “small,” say \( \leq h \) where \( h \) is prespecified
Half Width, Number of Replications (cont’d.)

- Set half-width = $h$, solve for $n = t_{n-1,1-\alpha/2} \frac{s^2}{h^2}$
- Not really solved for $n$ ($t$, $s$ depend on $n$)
- Approximation:
  - Replace $t$ by $z$, corresponding normal critical value
  - Pretend that current $s$ will hold for larger samples
  - Get

Easier, different approximation:

$s = \text{sample standard deviation from “initial” number } n_0 \text{ of replications}$

$s = \frac{z_{1-\alpha/2}^2}{h^2}$

$\frac{n}{n_0} = \frac{h_0}{h^2}$

$n_0 = \text{half width from “initial” number } n_0 \text{ of replications}$

$n \text{ grows quadratically as } h \text{ decreases}$
Application to Model 6-1

- From initial 10 replications, 95% half-width on Total Cost was ± 812.82 (3.8% of $\bar{X} = 21,618.33$)
  - Let’s get this down to ± 250 or less
- First formula: $n \approx 1.96^2(1136.24^2/250^2) = 79.4$, so 80
- Second formula: $n \approx 10(812.76^2/250^2) = 105.7$, so 106
- Modified Model 6-1 into Model 6-2
  - Checked Run > Run Control > Batch Run (No Animation) for speed
  - In Run > Setup > Replication Parameters, changed Number of Replications to 110 (conservative based on above)
- Got 22175.19 ± 369.54, close to criterion (undershot a bit?)
  - BTW, from 110 replications got 11.745 ± 0.51 on Percent Rejected
  - Use max of sample sizes for precisions on multiple outputs
Interpretation of Confidence Intervals

- Interval with random (data-dependent) endpoints that’s supposed to have stated probability of containing, or covering, expected valued
  - “Target” expected value is a fixed, but unknown, number
  - Expected value = average of infinite number of replications
- Not an interval that contains, say, 95% of data
  - That’s a prediction interval … useful too, but different
- Usual formulas assume normally-distributed data
  - Never true in simulation
  - Might be approximately true if output is an average, rather than an extreme
  - Central limit theorem
  - Robustness, coverage, precision – see text (Model 6-3)
Comparing Two Scenarios

- Usually compare alternative system scenarios, configurations, layouts, sensitivity analysis
  - For now, just two scenarios ... more later

- Model 6-4
  - Model 6-3, except reduce to 110 replications, add file Total Cost.dat to Statistic module, Output column, Total Cost row
  - Similarly for percent rejected
  - Saves output statistics to these files for each replication
  - Two scenarios
    - Base case – all inputs as original Model 5-3, no extra resources
      Effect on total cost, percent rejected?
Comparing Two Scenarios (cont’d.)

• Reasonable but not-quite-right idea
  ▪ Make confidence intervals on expected outputs from each scenario, see if they overlap; look at Total Cost
  ▪ Base case:
    \[ 22175.19 \pm 369.54, \text{ or } [21805.65, 22544.73] \]
  ▪ More-resources case:
    \[ 24542.82 \pm 329.11, \text{ or } [24213.71, 24871.93] \]
  ▪ But this doesn’t allow for a precise, efficient statistical conclusion
Compare Means via Output Analyzer

• Output Analyzer is a separate application that operates on .dat files produced by Arena
  ▪ Launch separately from Windows, not from Arena
• To save output values (Expressions) of entries in Statistic data module (Type = Output) – enter filename.dat in Output File column
  ▪ Did for both Total Cost and Percent Rejected
  ▪ Will overwrite these file names next time
    – Either change names in Arena model, or out in operating system before next run
  ▪ .dat files are binary … can only be read by Output Analyzer
Compare Means via Output Analyzer (cont’d.)

- Start Output Analyzer, open a new data group
  - Basically, a list of .dat files of current interest
  - Can save data group for later use – .dgr file extension
  - Add button to select (Open) .dat files for data group

- Analyze > Compare Means menu option
  - Add data files … “A” and “B” for two scenarios
  - Select “Lumped” for Replications field
  - Title, confidence level, accept Paired-t Test, do not Scale Display since two output performance measures have different units
Compare Means via Output Analyzer (cont’d.)

- Results:
  - Confidence intervals on differences both miss 0
  - Conclude that there is a (statistically) significant difference on both output performance measures
Evaluating Many Scenarios with Process Analyzer (PAN)

- With (many) more than two scenarios to compare, two problems are:
  - Simple mechanics of making many parameter changes, making many runs, keeping track of many output files
  - Statistical methods for drawing reliable, useful conclusions
- Process Analyzer (PAN) addresses these
- PAN operates on program (.p) files – produced when .doe file is run (or just checked)
- Start PAN from Arena (Tools > Process Analyzer) or via Windows
- PAN runs on its own, separate from Arena
PAN Scenarios

• A **scenario** in PAN is a combination of:
  - A program (.p) file
  - Set of input *controls* that you choose
    - Chosen from Variables and Resource capacities – think ahead
    - You fill in specific numerical values
  - Set of output *responses* that you choose
    - Chosen from automatic Arena outputs or your own Variables
    - Values initially empty … to be filled in after run(s)
  - To create a new scenario in PAN, double-click where indicated, get Scenario Properties dialog
    - Specify Name, Tool Tip Text, .p file, controls, responses
    - Values of controls initially as in model, but *you can change them in PAN* – this is the real utility of PAN
    - Duplicate (right-click, Duplicate) scenarios, then edit for a new one
  - Think of a scenario as a row
PAN Projects and Runs

• **A project** in PAN is a collection of scenarios
  - Program files can be the same `.p` file, or `.p` files from different model `.doe` files
  - Controls, responses can be same, or differ across scenarios in a project – usually will be mostly the same
  - Think of a project as a collection of scenario rows – a table
  - Can save as a PAN (.pan extension) file

• Select scenarios in project to run (maybe all)
• PAN runs selected models with specified controls
• PAN fills in output-response values in table
  - Equivalent to setting up, running them all “by hand” but much easier, faster, less error-prone
Model 6-5 for PAN Experiments

- Same as Model 6-4 but remove Output File entries in Statistic module
  - PAN will keep track of outputs itself, so this is faster
  - Stick with 110 replications

- Start PAN, New project, double-click for scenario
  - Name = Base Case
  - Program File = Model 06-05.p (maybe with path)

- Six controls – all data type Integer
  - Resources > capacity of Trunk Line

- Responses – both from User Specified
  - Total Cost, Percent Rejected

Could also do a designed experiment with PAN, for more efficient study of controls’ effects, interactions.
Model 6-5 for PAN Experiments (cont’d.)

- **Experimental (non-base-case) scenarios**
  - Suppose you get $1360 more per week for more resources
    - Must spend all $1360 on a single type of resource; could get
      13 more trunk lines @ $98 each
      4 more of any one of single-product tech-support people @ $320 each
      3 more of all-product tech-support people @ $360 each
      4 more sales people @ $340 each
  - Create six more PAN scenarios
    - Right-click, Duplicate Scenario(s), edit fields
    - See saved PAN file `Experiment 06-05.pan`
  - Execute scenarios
    - Select which to run (click on left, Ctrl-Click, Shift-Click)
    - or Run > Go or F5
Model 6-5 for PAN Experiments (cont’d.)

What to make of all this? Statistical meaningfulness?
Statistical Comparisons with PAN

- Model 6-5 scenarios were made with 110 replications each
  - Better than one replication, but what about statistical validity of comparisons, selection of “the best”?
- Select Total Cost column, Insert > Chart (or right-click on column, then Insert Chart)
  - Chart Type: Box and Whisker
  - Next, Total Cost; Next defaults
  - Next, Identify Best Scenarios
    - Smaller is Better, Error Tolerance = 0 (not the default)
    - Show Best Scenarios; Finish

Repeat for Percent Rejected
Statistical Comparisons with PAN (cont’d.)

- **Vertical boxes:** 95% confidence intervals
- **Red scenarios statistically significantly better than blues**
  - More precisely, red scenarios are 95% sure to contain best one
  - Narrow down red set – more replications, or Error Tolerance > 0
  - More details in text

**Numerical values (including c.i. half widths) in chart** – right click on chart, Chart Options, Data

So which scenario is “best”? Criteria disagree. Combine them somehow?

Scenarios considered via PAN are just a few of many

Seek input controls minimizing Total Cost while keeping Percent Rejected $\leq 5$

- Explore all possibilities – add resources in any combination
- New rules:
  
  $26 \leq \text{number of trunk lines} \leq 50$
  
  Total number of new employees of all five types $\leq 15$
Searching for an Optimal Scenario with OptQuest – Formulation

- Formulate as an **optimization** problem:

  Minimize **Total Cost** \( \leq \text{Objective function is a simulation-model output} \)
  Subject to
  
  \[
  26 \leq MR(\text{Trunk Line}) \leq 50
  \]
  
  \[
  0 \leq \text{New Sales} + \text{New Tech 1} + \text{New Tech 2} + \text{New Tech 3} + \text{New Tech All} \leq 15
  \]
  
  \[
  \text{Percent Rejected} \leq 5 \leq \text{Constraint on another output}
  \]

- Reasonable start – best acceptable scenario so far
  - No PAN scenarios satisfied **Percent Rejected** \( \leq 5 \), so start with more-resources case earlier (29 trunk lines, 3 new employees of each of five types)

- Where to go from here? Explore all of feasible six-dimensional space exhaustively? **No.**
  - For this problem, choice (decision) variables are discrete, so can enumerate that there are 1,356,600 feasible scenarios – with 110 replications per scenario, would take two months on 2.1GHz PC.
Searching for an Optimal Scenario with OptQuest – Operation

- **OptQuest searches intelligently for an optimum**
  - Like PAN, OptQuest ...
    - runs as a separate application … can be launched from Arena
    - “takes over” running of your model
    - asks you to identify input controls, the output (just one) objective
  - Unlike PAN, OptQuest ...
    - allows you to specify constraints on input controls
    - allows you to specify “constraints” on outputs
    - decides itself what input-control-value combinations to try
    - uses internal heuristic algorithms to decide how to change input controls to move toward an optimum configuration

- **There are various stopping criteria for search**
  - Default is no significant improvement for 100 scenarios
Searching for an Optimal Scenario with OptQuest – Example

- **Model 6-6 for OptQuest**
  - Model 6-5, but OptQuest requires finite Replication Length
  - Make sure Model 6-6 model window is active

- **Make sure desired model window is active**

- **Tools > OptQuest for Arena**
  - New Optimization or Browse for saved one (.opt)
  - Tree on left, expand for Controls and Responses
Searching for an Optimal Scenario with OptQuest – Controls, Responses

- **Controls → Resources → Trunk Line**
  - Integer, Lower Bound = 26, Suggested Value = 29, Upper Bound = 50

- **Controls → User Specified → New Sales**
  - Integer, Lower Bound = 0, Suggested Value = 3, Upper Bound = 15
  - Similarly for others ... open Optimum Seeking 06-06.opt
  - Click on “Included” to collect selections at top or bottom

- **Responses → User Specified → Output**
  - Check Percent Rejected, Total Cost
Searching for an Optimal Scenario with OptQuest – Constraints, Objective

- **Constraints**
  - Add button, then each of first five controls, “+”, then “<= 15”
  - Add button, then Percent Rejected, then “<= 5”

- **Objectives**
  - Add button, Total Cost, Minimize radio button

- **Options**
  - Stopping rules
  - Tolerance for regarding results as “equal”
  - Replications per simulation
  - Solutions log file location
    - Stores all scenarios tried, results – valuable for second best, etc.
Searching for an Optimal Scenario with OptQuest – Running

- **or Run > Start or F5**
  - Optimization branch on tree to watch progress, scenarios so far, best scenario so far

- Can’t absolutely guarantee a true optimum
  - But usually finds far better configuration than possible by hand