Computer Systems Performance Analysis and Benchmarking (37-235)

Analytic Modeling
Simulation
Measurements / Benchmarking

Lecture by:
Prof. Thomas Stricker

Assignments/Projects:
Christian Kurmann

Textbook:
Raj Jain, "The Art of Computer Systems Performance Analysis", 1991 Wiley & Sons, New York

Topic of Today:

- **Proof of SST=SSA+SSB+SSAB**
- **Introduction to Simulation**
- **Verification / Validation**
- **Transient Removal**

Proof for Allocation of Variation

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

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

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

The model used in a 2^2 design is

$$y_i = q_0 + q_A x_{Ai} + q_B x_{Bi} + q_{AB} x_{Ai} x_{Bi}$$

Columns x_A, x_B, and $x_A x_B$ of the design matrix in Table 17.3 have the following properties:

1. The sum of entries in each column is zero:

$$\sum_{i=1}^{4} x_{Ai} = 0; \qquad \sum_{i=1}^{4} x_{Bi} = 0; \qquad \sum_{i=1}^{4} x_{Ai} x_{Bi} = 0$$

2. The sum of the squares of entries in each column is 4:

$$\sum_{i=1}^{4} x_{Ai}^2 = 4; \qquad \sum_{i=1}^{4} x_{Bi}^2 = 4; \qquad \sum_{i=1}^{4} (x_{Ai} x_{Bi})^2 = 4$$

3. The columns are orthogonal since the inner product of any two columns is zero:

$$\sum_{i=1}^{4} x_{Ai} x_{Bi} = 0; \qquad \sum_{i=1}^{4} x_{Ai} (x_{Ai} x_{Bi}) = 0 = \sum_{i=1}^{4} x_{Bi} (x_{Ai} x_{Bi})$$

Proof ...

These properties allow us to compute the total variation

$$\text{Sample mean } \bar{y} = \frac{1}{4}\sum_{i=1}^{4} y_i$$

$$= \frac{1}{4}\sum_{i=1}^{4} (q_0 + q_A x_{Ai} + q_B x_{Bi} + q_{AB} x_{Ai} x_{Bi})$$

$$= \frac{1}{4}\sum_{i=1}^{4} q_0 + \frac{1}{4} q_A \sum_{i=1}^{4} x_{Ai} + q_B \frac{1}{4}\sum_{i=1}^{4} x_{Bi} + q_{AB} \frac{1}{4}\sum_{i=1}^{4} x_{Ai} x_{Bi}$$

$$= q_0$$

and

$$\text{Total variation} = \sum_{i=1}^{4} (y_i - \bar{y})^2$$

$$= \sum_{i=1}^{4} (q_A x_{Ai} + q_B x_{Bi} + q_{AB} x_{Ai} x_{Bi})^2$$

$$= \sum_{i=1}^{4} (q_A x_{Ai})^2 + \sum_{i=1}^{4} (q_B x_{Bi})^2 + \sum_{i=1}^{4} (q_{AB} x_{Ai} x_{Bi})^2$$

$$+ \text{product terms}$$

$$= q_A^2 \sum_{i=1}^{4} (x_{Ai})^2 + q_B^2 \sum_{i=1}^{4} (x_{Bi})^2 + q_{AB}^2 \sum_{i=1}^{4} (x_{Ai} x_{Bi})^2 + 0$$

$$= 4q_A^2 + 4q_B^2 + 4q_{AB}^2$$

Common Mistakes

- Inappropriate level of detail
- Improper language
- Unverified models
- Invalid models
- Improperly handled initial conditions
- Too short simulations
- Poor random number generators
- Improper selection of seeds

...and all the common mistakes in software engineering

Checklist

Box 24.1 Checklist for Simulations

1. Checks before developing a simulation:
   (a) Is the goal of the simulation properly specified?
   (b) Is the level of detail in the model appropriate for the goal?
   (c) Does the simulation team include personnel with project leadership, modeling, programming, and computer systems backgrounds?
   (d) Has sufficient time been planned for the project?
2. Checks during development:
   (a) Has the random-number generator used in the simulation been tested for uniformity and independence?
   (b) Is the model reviewed regularly with the end user?
   (c) Is the model documented?
3. Checks after the simulation is running:
   (a) Is the simulation length appropriate?
   (b) Are the initial transients removed before computation?
   (c) Has the model been verified thoroughly?
   (d) Has the model been validated before using its results?
   (e) If there are any surprising results, have they been validated?
   (f) Are all seeds such that the random-number streams will not overlap?

See also Box 2.1 for a checklist applicable to all performance evaluation projects.

Terminology

- State variable (set of all state variables)
- Event
- Time models:
  - Continuous time
  - Discrete time
- State models
  - Continuous state/continuous event
  - Discrete state/discrete event
- Determinism Property:
  - Deterministic model
  - Probabilistic model
- Time Property:
  - Static model
  - Dynamic model

- Model Equation Property
  - Linear model
  - Nonlinear model
- System Property
  - Open model
  - Closed model
- Convergence
  - Stable model
  - Unstable model

Examples:

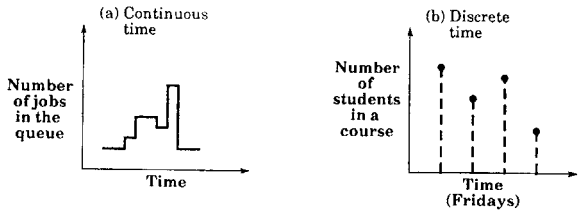


FIGURE 24.1 Continuous-time versus discrete-time models.

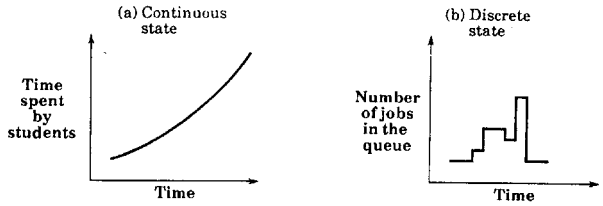


FIGURE 24.2 Continuous-state versus discrete-state models.

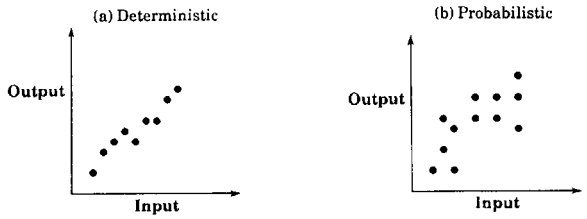


FIGURE 24.3 Deterministic and probabilistic models.

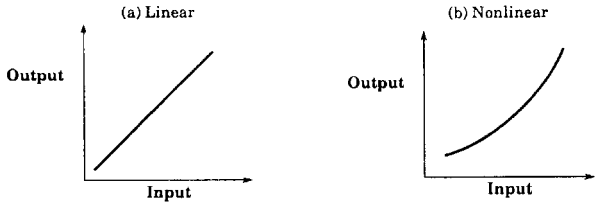


FIGURE 24.4 Linear and nonlinear models.

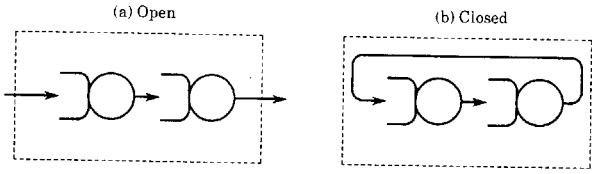


FIGURE 24.5 Open and closed models.

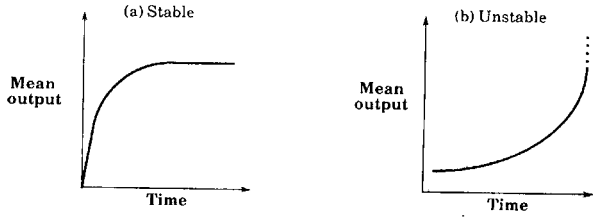


FIGURE 24.6 Stable and unstable models.

Languages for Simulation

Specialized languages:

- Simula
- SIMSCRIPT

General Purpose

- Oberon, Fortran, C

General Purpose with Toolkit

- Fortran + GASP
- QNET and RESQ

Differential Equation Systems

- CSMP and Dynamo

Discrete Event Systems

- Simula and GPSS

Monte Carlo Simulations

Example 24.1 The following integral is to be evaluated:

$$I = \int_0^2 e^{-x^2} dx$$

One way to evaluate this integral is to generate uniformly distributed random numbers x and for each number compute a function y as follows:

$$x \sim \text{Uniform}(0,2)$$

$$\text{Density function } f(x) = \tfrac{1}{2} \quad \text{iff} \quad 0 \le x \le 2$$

$$y = 2e^{-x^2}$$

The expected value of y is

$$E(y) = \int_0^2 2e^{-x^2} f(x)\,dx$$

$$= \int_0^2 2e^{-x^2} \tfrac{1}{2}\,dx$$

$$= \int_0^2 e^{-x^2} dx$$

$$= I$$

Thus, the integral can be evaluated by generating uniformly distributed random numbers x_i, computing y_i, and then averaging as follows:

$$x_i \sim \text{Uniform}(0,2)$$

$$y_i = 2e^{-x_i^2}$$

$$I = E(y) = \frac{1}{n}\sum_{i=1}^{n} y_i \qquad \square$$

Trace Driven Simulation

Advantages:

- Credibility
- Easy validation
- Accurate workload
- Detailed trade-offs
- Less randomness
- Fair comparison
- Similarity to the actual implementation

Disadvantages:

- Complexity
- Representativeness
- Finiteness
- Single point of validation
- Detail
- Trade-offs (no change to trace)

Discrete Event Simulation

- Event scheduler
- Simulation clock and a time advancing mechanism
  - unit time
  - event driven
- System state variables
- Event routines
- Input routines
- Report generators
- Initialization routines
- Trace routines
- Dynamic memory management
- Driver: main program

Data structures for the event queue

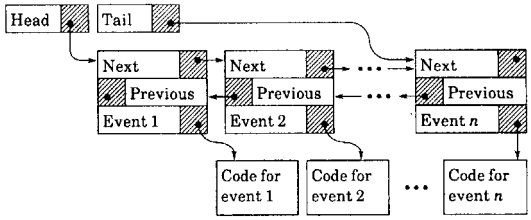


FIGURE 24.7 Doubly linked list.

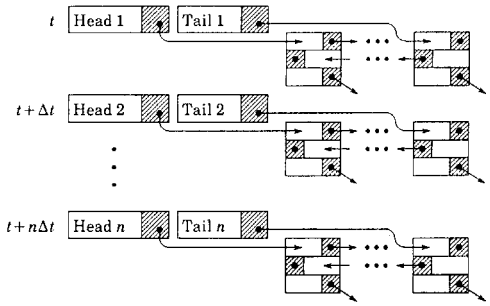
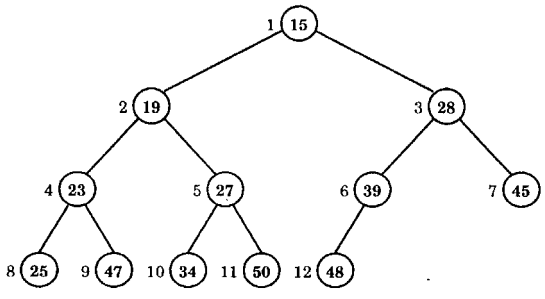
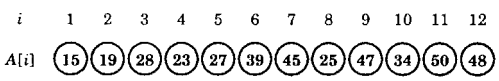


FIGURE 24.8 Indexed list.



(a) Tree representation of a heap.



(b) Array representation of a heap.

FIGURE 24.9 A heap.

Verification vs. Validation

Verification of a simulator

- Is the simulator correct?

Validation of a Simulator

- Is the simulator applicable?
- Are the assumptions reasonable?

Simulators are complex pieces of software:

- Modularity
  - some generic modules
  - some domain dependent modules
  - some model specific modules

Example

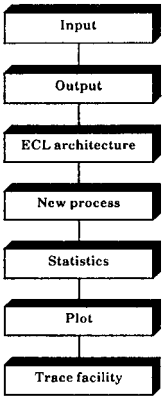


FIGURE 25.1 Layered structure of the congestion simulation model.

Example 25.1 Figure 25.1 shows the modules for a computer network simulation developed for congestion control studies. The model simulates a network with a number of source nodes, a number of intermediate nodes, and a number of destinations, as shown in Figure 25.2. Packets start from the source nodes, travel through a number of prespecified intermediate nodes (called paths), and reach the destination. The packet sizes and service times at various nodes are randomly distributed. In Figure 25.2, S_i's are sources, R_i's are intermediate nodes, and D_i's are destinations. The model simulates n sources sharing a common path throught m intermediate nodes for any given n and m. This is equivalent to two local-area networks connected through m intermediate nodes.

Example

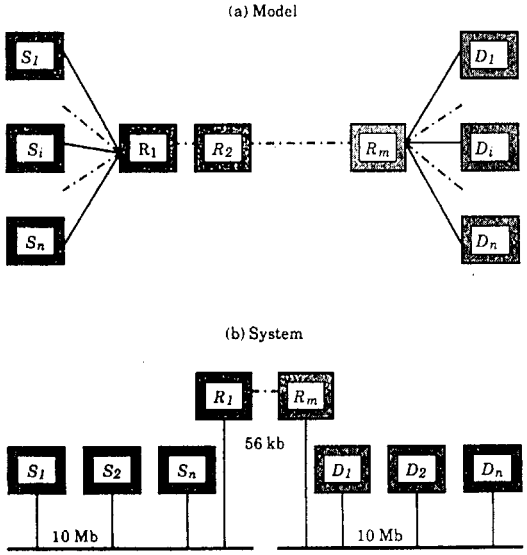


FIGURE 25.2 Model of two interconnected local-area networks.

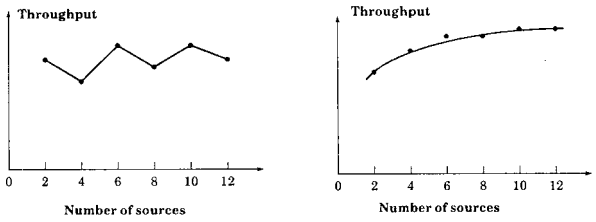
Techniques:

- The usual game in software engineering...
- Modular design
- Debugging
- Structured walk-through
- Deterministic models
- Simplified cases
- On-line graphic animation
- Traces
- Continuity tests
- Degeneracy tests
- Consistency tests
- Seed independence tests

Traces/Continuity tests

```
PKT=100 CR=4 #S=2 RS=2                                  ! This was a run with 100
                                                          packets, 4 credits per source,
                                                          two sources. Router service time
                                                          was twice that of the source.
Time    Node:    Pkt #/   Sample  Delay
        Event    Attempt  Delay   Estimate
-----   -------  -------  ------  --------
0.00:   S1: TIMR  1- 1                    ! Round trip delay-measuring
                                            stop watch started by source 1
        S1: SEND  1- 1/ 1                 ! Packet 1 sent by source 1
        S1: STRT                          ! Timeout alarm clock set by
                                            source 1
        S2: TIMR  2- 1                    ! Round trip delay-measuring
                                            stop watch started by source 2
        S2: SEND  2- 1/ 1                 ! Packet 1 sent by source 2
        S2: STRT                          ! Timeout alarm clock set by
                                            source 2
1.00:   R1: QUED  1- 1/ 1                 ! Packet 1 of source 1 was put
                                            into a queue at router 1
        R1: LOST! 2- 1/ 1                 ! Packet 1 of source 2 was lost due
                                            to lack of buffer at router 1
3.00:   D1: RECD  1- 1/ 1                 ! Packet 1 of source 1 was
                                            received at destination 1
        S1: ACKD  1- 1                    ! Acknowledgment for packet 1
                                            was received at source 1
        S1: UPDT  1- 1    3.00    3.00    ! Source 1 updated its estimate of
                                            round trip delay
```

FIGURE 25.3 Sample packet event-trace output from the simulation model.



Validation Techniques

Three Key Aspects of the system

- Assumptions
- Input parameter values
- output parameter values - conclusions

Methods

- Expert intuition
- Real systems measurements
- Theoretical results

Expert Intuition Example

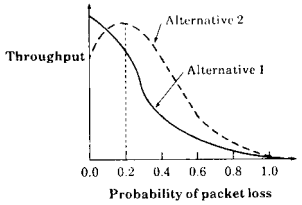


FIGURE 25.6 Example of problems caused by invalid assumptions that are easily detected by experts.

Simple Statistical Methods for...

- Transient removal
- Termination tail removal
- Stopping criteria
- Regenerative systems

Transient removal

- Long runs
- Proper initialization
- Truncation
- Initial data deletion
- Moving average of independent replications
- Batch means

Truncation

1,2,3,4,5,6,7,8,9,10,11,10,9,10,11,10,9,10

Example 25.2 Consider the following sequence of observations: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 10, 9, 10, 11, 10, 9, 10, 11, 10, 9,....

Ignoring the first observation ($l = 1$), the range of the remaining observations is (2, 11). Since the second observation is equal to the minimum, the transient phase is longer than 1.

Ignoring the first two observation ($l = 2$), the range of the remaining sequence is (3, 11). Again, the next (third) observation is equal to the minimum; the truncation continues with $l = 3$ and so on.

Finally, at $l = 9$ the range of the remaining sequence is (9, 11), and the tenth observation 10 is neither the minimum nor the maximum. The length of the transient interval is therefore 9, and the first nine observations are discarded.

A trajectory of this set of observations is shown in Figure 25.7. It is seen from the figure that the transient phase for this data does indeed end after nine observations. □

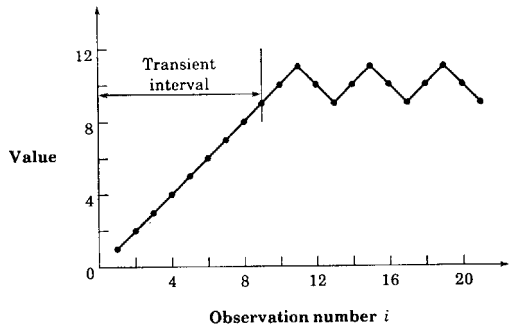


FIGURE 25.7 Plot of the data used in the truncation method example.

Systematic Method

Initial Data Deletion

1. Get a mean trajectory by averaging across replications:

$$\overline{x}_j = \frac{1}{m}\sum_{i=1}^{m} x_{ij}, \qquad j = 1,2,\ldots,n$$

Figure 25.8a shows the trajectories of several replications, and an average trajectory is shown in Figure 25.8b.

2. Get the overall mean:

$$\overline{\overline{x}} = \frac{1}{n}\sum_{j=1}^{n} \overline{x}_j$$

Set $l = 1$ and proceed to the next step.

3. Assuming that the transient state is only l long, delete the first l observations from the mean trajectory and get an overall mean from the remaining $n - l$ values:

$$\overline{\overline{x}}_l = \frac{1}{n-l}\sum_{j=l+1}^{n} \overline{x}_j$$

4. Compute the relative change in the overall mean:

$$\text{Relative change} = \frac{\overline{\overline{x}}_l - \overline{\overline{x}}}{\overline{\overline{x}}}$$

5. Repeat steps 3 and 4 by varying l from 1 to $n - 1$. Plots of the overall mean and the relative change as functions of l are shown in Figure 25.8c and 25.8d. After a certain value of l, the relative change graph stabilizes. This point is known as the knee, and the value of l at the knee is the length of the transient interval.

Moving Average of Independent Replic.

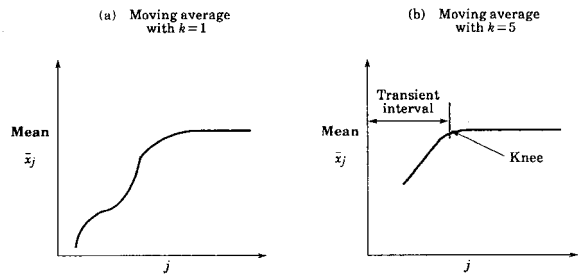


FIGURE 25.9 Moving average of independent replications.

1. Get a mean trajectory by averaging across replications:

$$\bar{x}_j = \frac{1}{m}\sum_{i=1}^{m} x_{ij}, \qquad j = 1,2,\ldots,n$$

   Set $k = 1$ and proceed to the next step.
2. Plot a trajectory of the moving average of successive $2k + 1$ values:

$$\bar{\bar{x}}_j = \frac{1}{2k+1}\sum_{l=-k}^{k} \bar{x}_{j+l}, \qquad j = k+1, k+2, \ldots, n-k$$

3. Repeat step 2, with $k = 2,3,\ldots$ until the plot is sufficiently smooth.
4. Find the knee of the plot. The value of j at the knee gives the length of the transient phase.

Batch Means

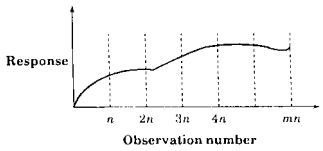


FIGURE 25.10 Transient removal by batch means requires dividing the data into m batches of size n each.

1. For each batch, compute a batch mean:

$$\bar{x}_i = \frac{1}{n}\sum_{j=1}^{n} x_{ij}, \qquad i = 1,2,\ldots m$$

2. Compute the overall mean:

$$\bar{\bar{x}} = \frac{1}{m}\sum_{i=1}^{m} \bar{x}_i$$

3. Compute the variance of the batch means:

$$\text{Var}(\bar{x}) = \frac{1}{m-1}\sum_{i=1}^{m} (\bar{x}_i - \bar{\bar{x}})^2$$

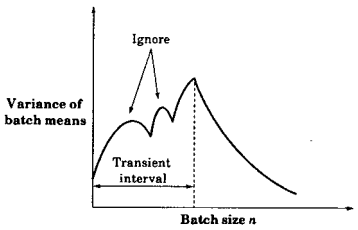


FIGURE 25.11 Transient removal by batch means.

Stopping Criteria / Variance Estimation

Independent Replication

1. Compute a mean for each replication:

$$\bar{x}_i = \frac{1}{n}\sum_{j=n_0+1}^{n_0+n} x_{ij}, \qquad i = 1,2,\ldots,m$$

2. Compute an overall mean for all replications:

$$\bar{\bar{x}} = \frac{1}{m}\sum_{i=1}^{m} \bar{x}_i$$

3. Calculate the variance of replicate means:

$$\text{Var}(\bar{x}) = \frac{1}{m-1}\sum_{i=1}^{m} (\bar{x}_i - \bar{\bar{x}})^2$$

The confidence interval for the mean response is*

$$[\bar{\bar{x}} \mp z_{1-\alpha/2}\text{Var}(\bar{x})]$$

Batch means

TABLE 25.1 Autocovariance and Variance for Various Batch Sizes

Batch Size	Autocovariance	Variance
1	−0.18792	1.79989
2	0.02643	0.81173
4	0.11024	0.42003
8	0.08979	0.26437
16	0.04001	0.17650
32	0.01108	0.10833
64	0.00010	0.06066
128	−0.00378	0.02992
256	0.00027	0.01133
512	0.00069	0.00503
1024	0.00078	0.00202

Batch Means

1. Compute means for each batch:

$$\bar{x}_i = \frac{1}{n}\sum_{j=1}^{n} x_{ij}, \qquad i = 1,2,\ldots,m$$

2. Compute an overall mean:

$$\bar{\bar{x}} = \frac{1}{m}\sum_{i=1}^{m} \bar{x}_i$$

3. Calculate the variance of batch means:

$$\text{Var}(\bar{x}) = \frac{1}{m-1}\sum_{i=1}^{m} (\bar{x}_i - \bar{\bar{x}})^2$$

The confidence interval for the mean response is

$$[\bar{\bar{x}} \mp z_{1-\alpha/2}\text{Var}(\bar{x})]$$

$$\text{Cov}(\bar{x}_i, \bar{x}_{i+1}) = \frac{1}{m-2}\sum_{i=1}^{m-1} (\bar{x}_i - \bar{\bar{x}})(\bar{x}_{i+1} - \bar{\bar{x}})$$

This quantity is also called the **autocovariance.**

Method of Regeneration

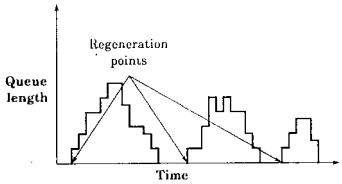


FIGURE 25.13 Regeneration points.

Suppose you have a regenerative simulation consisting of m cycles of sizes $n_1, n_2, \ldots, n_m$, respectively. Cycle means are given by

$$\bar{x}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} x_{ij}$$

However, the overall mean is not an arithmetic mean of cycle means:

$$\bar{\bar{x}} \neq \frac{1}{m} \sum_{i=1}^{m} \bar{x}_i$$

The correct procedure to compute the overall mean and its confidence interval is as follows:

1. Compute cycle sums:

$$y_i = \sum_{j=1}^{n_i} x_{ij}$$

2. Compute the overall mean:

$$\bar{\bar{x}} = \frac{\sum_{i=1}^{m} y_i}{\sum_{i=1}^{m} n_i}$$

3. Calculate the difference between expected and observed cycle sums:

$$w_i = y_i - n_i \bar{\bar{x}}, \qquad i = 1, 2, \ldots, m$$

4. Calculate the variance of the differences:

$$\text{Var}(w) = s_w^2 = \frac{1}{m-1} \sum_{i=1}^{m} w_i^2$$

5. Compute the mean cycle length:

$$\bar{n} = \frac{1}{m} \sum_{i=1}^{m} n_i$$

The confidence interval for the mean response is given by

$$\bar{\bar{x}} \mp z_{1-\alpha/2} \frac{s_w}{\bar{n}\sqrt{m}}$$