CHAPTER 11

Variance-Reduction Techniques

11.1 Introduction

Main drawback of using simulation to study stochastic models:

Results are uncertain — have *variance* associated with them

Would like to have as little variance as possible — more precise results

One sure way to decrease the variance:

Run it some more (longer runs, additional replications)

Not free

Sometimes can manipulate simulation to reduce the variance of the output at little or no additional cost — *not* just by running it some more

Another way of looking at it — try to achieve a desired level of precision (e.g., confidence-interval smallness) with less simulating — *Variance-reduction technique* (VRT)

Often, exploit controllability of random-number generator to recycle previously used random numbers and induce some kind of *correlation*

Basic relation used: For any RVs X and Y, and any constants a and b,

$$\text{Var}(aX + bY) = a^2\text{Var}(X) + b^2\text{Var}(Y) + 2ab\text{Cov}(X, Y)$$

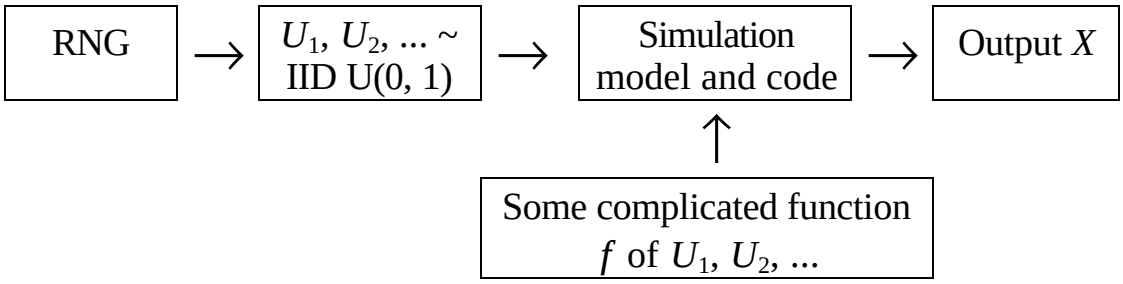
Several different VRTs in common (sometimes unconscious) use

Effectiveness of VRTs varies widely, unpredictably sometimes

Perhaps make preliminary "pilot" runs with and without a proposed VRT to measure how well (and *if*) it is working

Implementing VRTs requires care and understanding of the model and code

Useful "model" of a simulation's action to discuss VRTs:



i.e., view $X = f(U_1, U_2, \ldots)$; want to estimate $m = E(X) = E[f(U_1, U_2, \ldots)]$

11.2 Common Random Numbers

Applies when goal is to compare two (or more) alternative systems

Probably most widely used, successful VRT

Often used unconsciously

Other names: *correlated sampling*, *matched streams*, or *matched pairs*

Possible drawback: may invalidate (or at least complicate) statistical methods (e.g., ranking/selection, ANOVA)

Intuition

Compare the two alternatives under similar (external) conditions

"Compare like with like" — *blocking* in experimental-design terminology

Then the observed differences are more likely attributable to the actual system differences, rather than to the luck of the draw

11.2.1 Rationale

Estimate of difference is $X_1 - X_2$ (from one run of each)

$\text{Var}(X_1 - X_2) = \text{Var}(X_1) + \text{Var}(X_2) - 2\,\text{Cov}(X_1, X_2)$

If independent runs were made, $\text{Cov}(X_1, X_2)$ would be 0

Under CRN, we would expect that $\text{Cov}(X_1, X_2) > 0$, reducing $\text{Var}(X_1 - X_2)$

This effect clearly carries through on multiple runs of each

Implementation

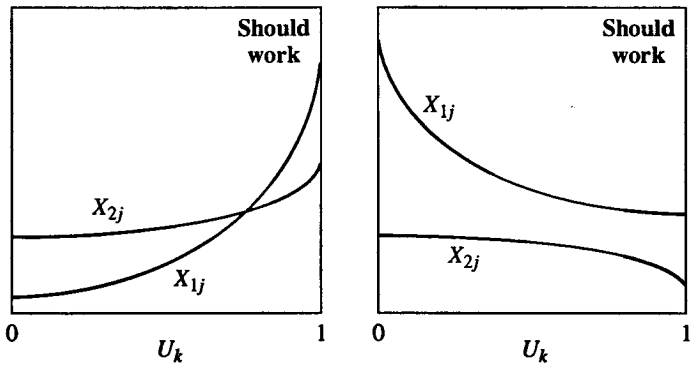
- Simulate system 1, get observation X_1
- Re-use the *same* random numbers used for system 1 to simulate system 2, and get X_2

Critical point: *synchronization*

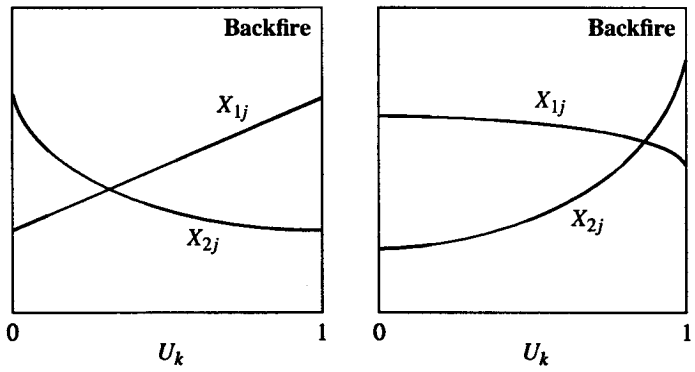
- Must re-use the same random numbers *for the same purposes* in the simulations of the two systems
- Failure to maintain synchronization of random-number usage can get the uses mixed up and dilute the effect of CRN, or even make it *backfire* (*increase* the variance); example later
- Best way to maintain synchronization: *Dedicate* a separate random-number *stream* to corresponding sources of randomness in the two systems (interarrival times, cycle times on specific machines, pass/fail decisions on inspections, etc.) — principal reason for having many separate and long streams in a random-number generator
- Also: Use inverse-transform method of variate generation wherever possible (one $U \rightarrow$ one X for simplicity, and induce strongest possible correlation between generated variates)

11.2.2 Applicability

Implicit Assumption About the Models: they will both react similarly to a large/small random number U_k used for a particular purpose:



If not, could get Cov(X_1, X_2) < 0 and backfiring (possible but probably rare):



Success of CRN in Terms of f Functions Representing the Simulation

Let f_1 and f_2 be the f functions for the two alternatives being compared

First suppose the "simulation" involves only one input random number:

If f_1 and f_2 are both monotone in the same direction, then Cov[$j_1(U)$, $j_2(U)$] ≥ 0, and so CRN will work (although how well we don't know)

Stated another way, $\frac{df_1}{du}\frac{df_2}{du} \geq 0$ is sufficient for CRN to work

Now, allow more than one input random number:

Def.: f_1 and f_2 are *concordant* if $\frac{\partial f_1}{\partial u_k}\frac{\partial f_2}{\partial u_k} \geq 0$ for each fixed k

Thm.: If f_1 and f_2 are concordant then

$$\text{Cov}[f_1(U_1, U_2, ...), f_2(U_1, U_2, ...)] \geq 0$$

and so CRN will work

In practice, how can we tell if our two alternatives are concordant?

We can't (reason for initial pilot experiments with and without CRN)

One important exception (has been shown to be concordant):

GI/*G*/*s* queues, as long as interarrival and service times are generated via inverse-transform method

11.2.3 Synchronization

Cannot just "reset the seed" for second alternative and let the simulation run using a single stream

Random-number usage in *non-synchronized* CRN for the *M*/*M*/1 vs. *M*/*M*/2 queue:

k	U_k	Usage in *M*/*M*/1	Usage in *M*/*M*/2	Agree?
1	0.401	A	A	Yes
2	0.614	A	A	Yes
3	0.434	S	S	Yes
4	0.383	A	A	Yes
5	0.506	S	S	Yes
6	0.709	A	A	Yes
7	0.185	S	S	Yes
8	0.834	A	A	Yes
9	0.646	A	S	No
10	0.376	A	A	Yes
11	0.348	S	S	Yes
12	0.764	A	A	Yes
13	0.446	A	A	Yes
14	0.910	S	A	No
15	0.474	S	A	No
16	0.475	A	S	No
17	0.852	S	S	Yes
18	0.804	S	S	Yes
19	0.723	S	A	No
20	0.700	A	A	Yes
.	.	.	.	.
.	.	.	.	.
197	0.574	A	S	No
198	0.859	A	A	Yes
199	0.096	A	S	No
200	0.154	S	A	No
201	0.326	S	S	Yes
202	0.577	A		
.	.	.		
.	.	.		
219	0.656	S		

Overall, only about *half* of the *U*'s were synchronized, which effectively neutralizes the benefit of CRN (example below)

Synchronization Methods (and "Tricks")

- Usually depends on structure of model and how its coded

- Dedicate a random-number stream to a particular source of randomness in the model; use separate streams for different sources of randomness
  - Make sure streams are long enough to avoid overlap — usually a rough calculation can estimate the number of random numbers needed for a particular purpose — if they'd overlap, then skip some intermediate streams

- To the extent possible, use the inverse-transform method of variate generation since it maintains one-to-one correspondence between random numbers and generated variates
  - Also maximizes strength of correlation induced, among all variate-generation approaches

- Feel free to "waste" some random numbers to maintain synchronization

- In queueing models, anticipate all of the random variates that an arriving entity could require during its time in the system, generate them at the time of arrival, and store them as attributes to ride along with the entity
  - Potentially major downside — increased memory use
  - Minor downside — generating variates that might not be needed for a particular entity

- When using CRN in multiple replications, ensure that synchronization is maintained on the 2nd and subsequent replications
  - Could jump to new streams for future replications — think of this as a two-dimensional stream-indexing scheme, or perhaps the replication streams being "substreams" of the major streams dedicated to different sources of randomness in the model

Clearly, it is highly desirable to have a random-number generator with a very large number of very long streams, especially as computers continue to get faster

11.2.4 Some Examples

Earlier $M/M/1$ (fast server) vs. $M/M/2$ (slow servers) — more examples in text

X_{ij} = average delay in queue from replication j of system i

$Z_j = X_{1j} - X_{2j}$ (pairing approach)

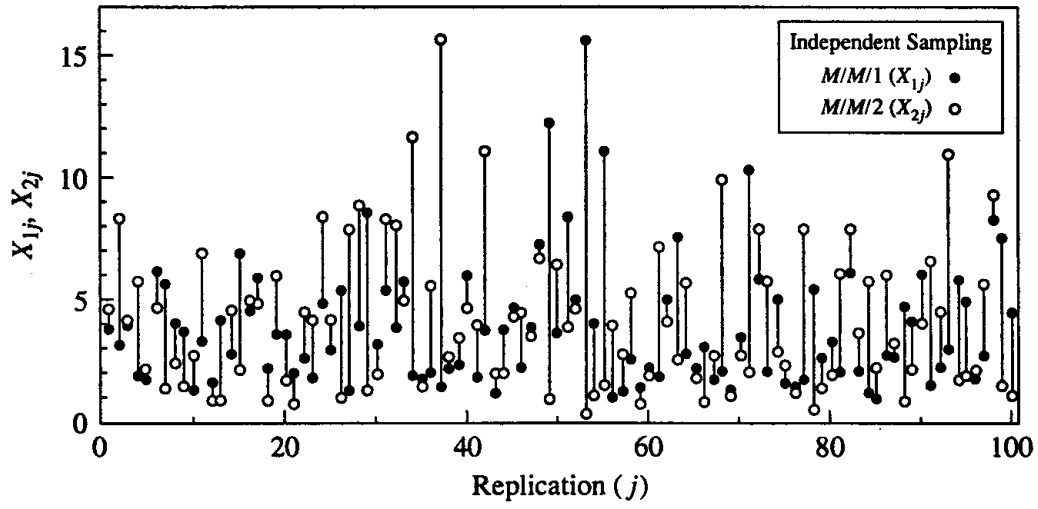
Made $n = 100$ pairs of runs, observed $Z_1, Z_2, ..., Z_{100}$

Options on degree to which CRN was used:

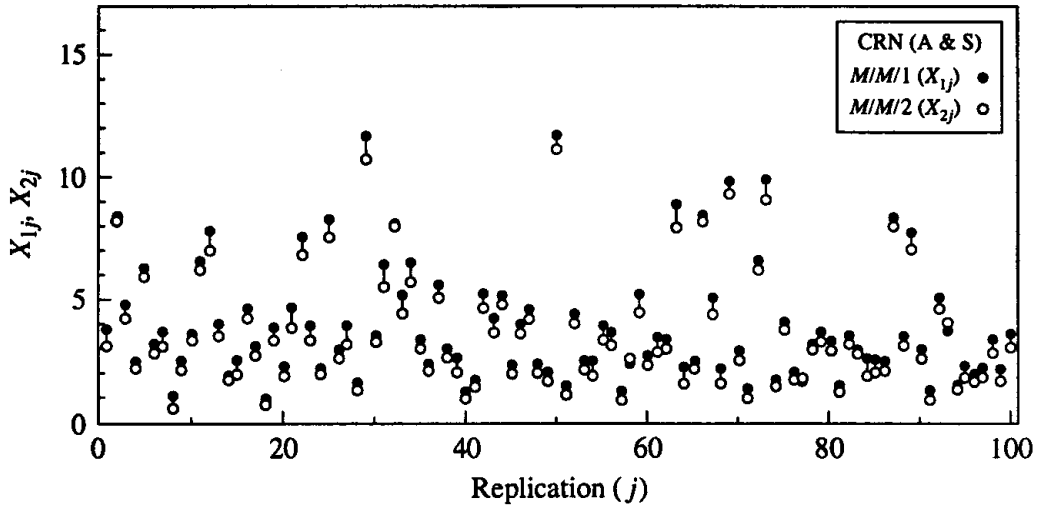
- I: Independent sampling (i.e., no CRN at all)
- A: Use CRN on interarrival times, but generate service times independently
- S: Use CRN on service times, but generate interarrival times independently
- A & S: Use CRN on both interarrival and service times

	I	A	S	A & S
Estimated variance of Z_j	18.00	9.02	8.80	0.07
90% c.i. half-length	0.70	0.49	0.49	0.04
P(wrong answer)	0.52	0.37	0.40	0.03
Estimated Cor(X_1, X_2)	–0.17	0.33	0.44	0.995

Plot of individual X_{ij}'s vs. replication number, to show stabilizing effect of CRN:

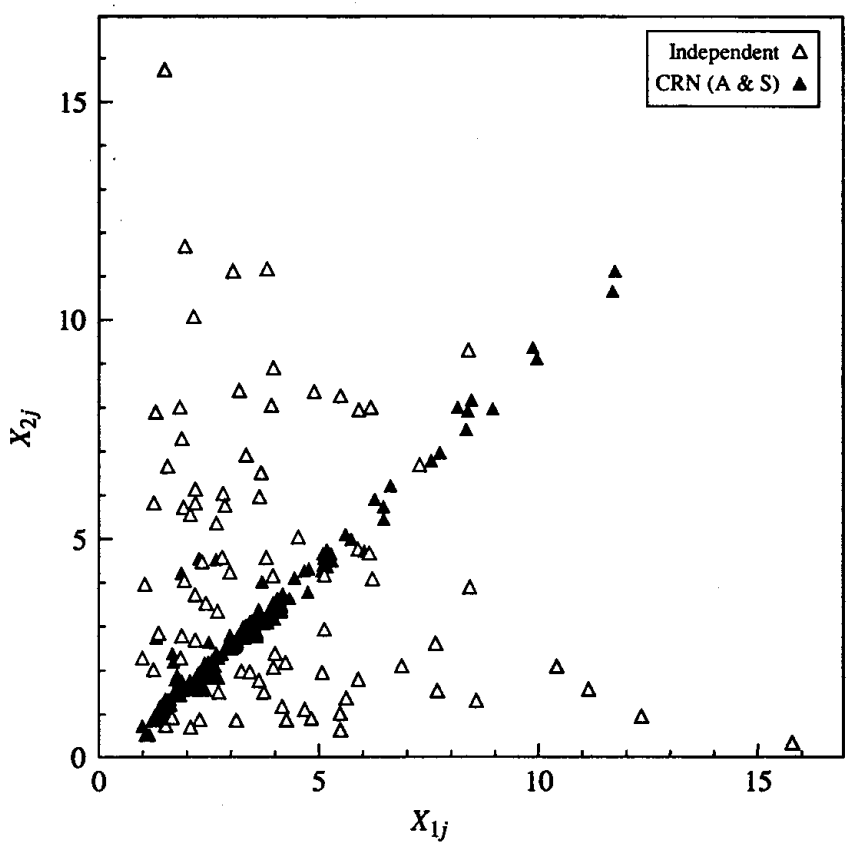


(*a*)

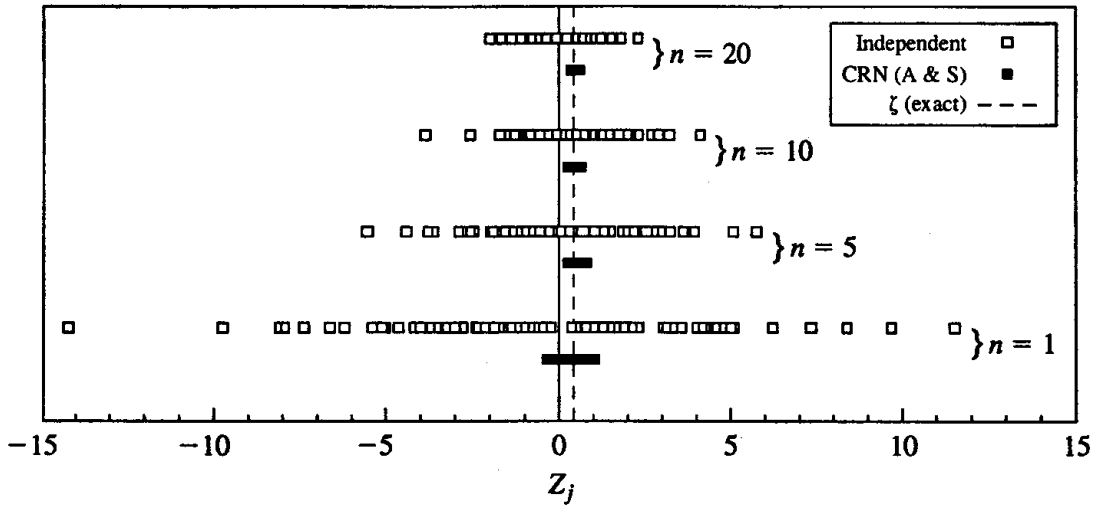


(*b*)

Plot of pairs (X_1, X_2) to show positive correlation:



Plot of Z_j's to show how CRN reduces the "spread" of the differences:



Ignoring Synchronization in *M*/*M*/1 vs. *M*/*M*/2

Still use the "same" random numbers, but program the model so that synchronization is ignored (two different programs that ignore it) — all codes are still correct in terms of simulating the models properly

Statistical consequences:

	Proper full synch.	Ignore synch.	Ignore synch. another way
$s^2(100)$	0.07	16.80	12.00
90% c.i. half-length	0.04	0.67	0.57
Proportion wrong order	0.07	0.43	0.42
Sample correlation	0.997	0.018	–0.028

Ignoring synchronization effectively neutralized the effect of CRN, underscoring the importance of synchronizing to as great an extent that is possible, in order to reap the full benefits of CRN

11.3 Antithetic Variates

Use for a single system (not comparisons)

Like CRN, recycle random numbers to induce correlation, but this time we want *negative* correlation

Intuition

- Counterbalance a "large" observation with a "small" one
- The average of the two observations should thus tend to be closer to the true expectation than if they were independent

Implementation

- Simulate the system, get observation $X^{(1)}$
- Re-use the same random numbers used to get $X^{(1)}$, but in their *complementary* form $1 - U$, to get $X^{(2)}$ — valid since $1 - U$ is also ~ U(0, 1)
- Use $[X^{(1)} + X^{(2)}]/2$ as "an observation" and replicate, etc. for analysis
- Critical point:
  - Must re-use the same random numbers *for the same purposes* in the two simulations of the system (just as in CRN)
  - Failure to maintain this *synchronization* of random-number usage can get the uses mixed up and dilute the effect of AV, or even make it *backfire* (*increase* the variance)
  - Best way to maintain synchronization: Random-number-stream dedication, as for CRN (but same techniques, tricks apply as in CRN)
- Also: Use inverse-transform for variate generation, as for CRN

Probabilistic Rationale for AV

Estimate of performance measure is $[X^{(1)} + X^{(2)}]/2$, which has variance
$\text{Var}\{[X^{(1)} + X^{(2)}]/2\} = \{\text{Var}[X^{(1)}] + \text{Var}[X^{(2)}] + 2\ \text{Cov}[X^{(1)}, X^{(2)}]\}/4$

If independent runs were made, $\text{Cov}[X^{(1)}, X^{(2)}]$ would be 0

Under AV, expect that $\text{Cov}[X^{(1)}, X^{(2)}] < 0$, reducing $\text{Var}\{[X^{(1)} + X^{(2)}]/2\}$

Implicit Assumption About the Model

It will react monotonically (up or down) to a large/small random number used for a particular purpose

In terms of f-function representation of simulation, it is sufficient that for each k, for $\dfrac{\partial f}{\partial u_k}$ to have the same sign for all $u_k \in [0, 1]$

In this case, $\text{Cov}[f(U_1, U_2, ...), f(1 - U_1, 1 - U_2, ...)] \leq 0$ and so AV is guaranteed to work (but again, we don't know how well)

If not, could get $\text{Cov}(X^{(1)}, X^{(2)}) > 0$ and backfiring

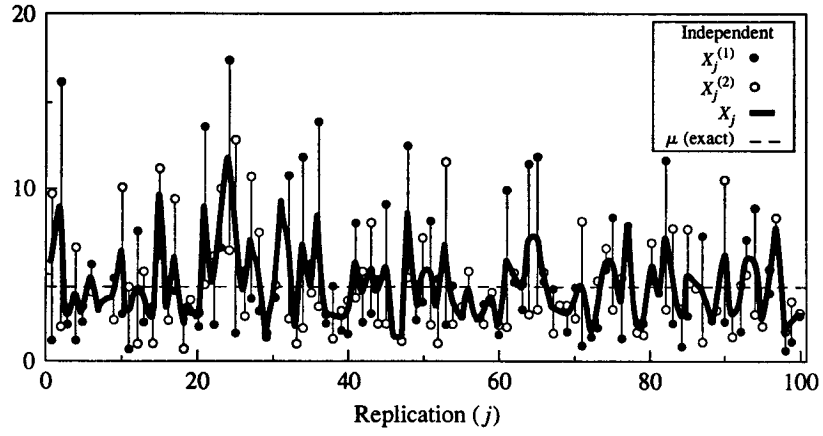
Example of Effectiveness of AV

M/*M*/1 queue, average-delay measure

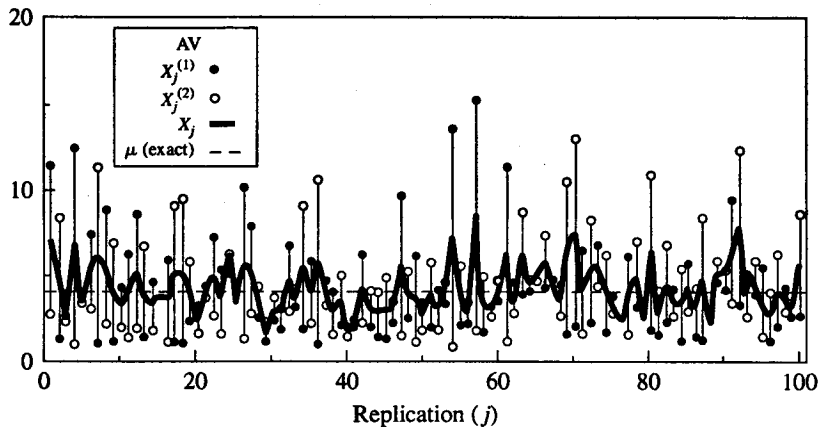
Made 100 pairs of runs using both independent sampling within a pair, and AV, observed 100 values X_j for $[X^{(1)} + X^{(2)}]/2$

	Independent	AV
Estimated variance of X_j	4.84	1.94
90% c.i. half-length	0.36	0.23
Estimated Cor($X^{(1)}$, $X^{(2)}$)	–0.07	–0.52

Plot of $X^{(1)}$, $X^{(2)}$, and X_j by replication, to show stabilizing effect of AV:

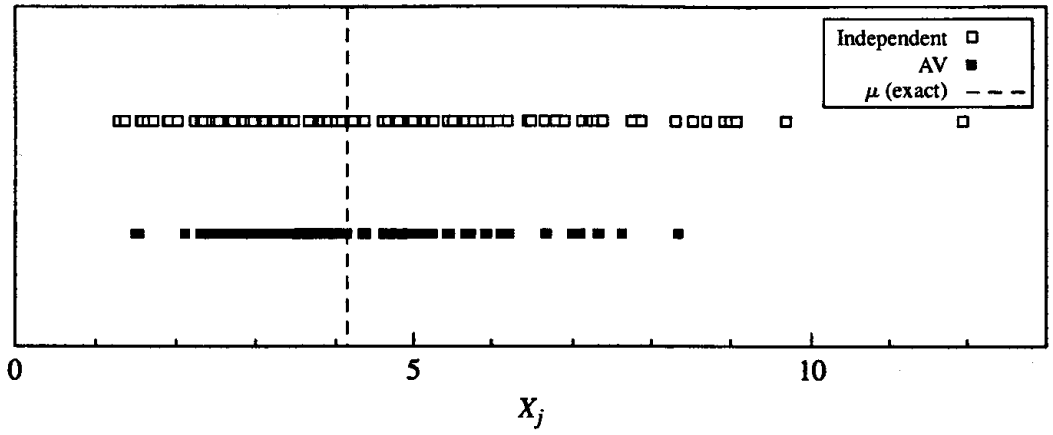


(*a*)

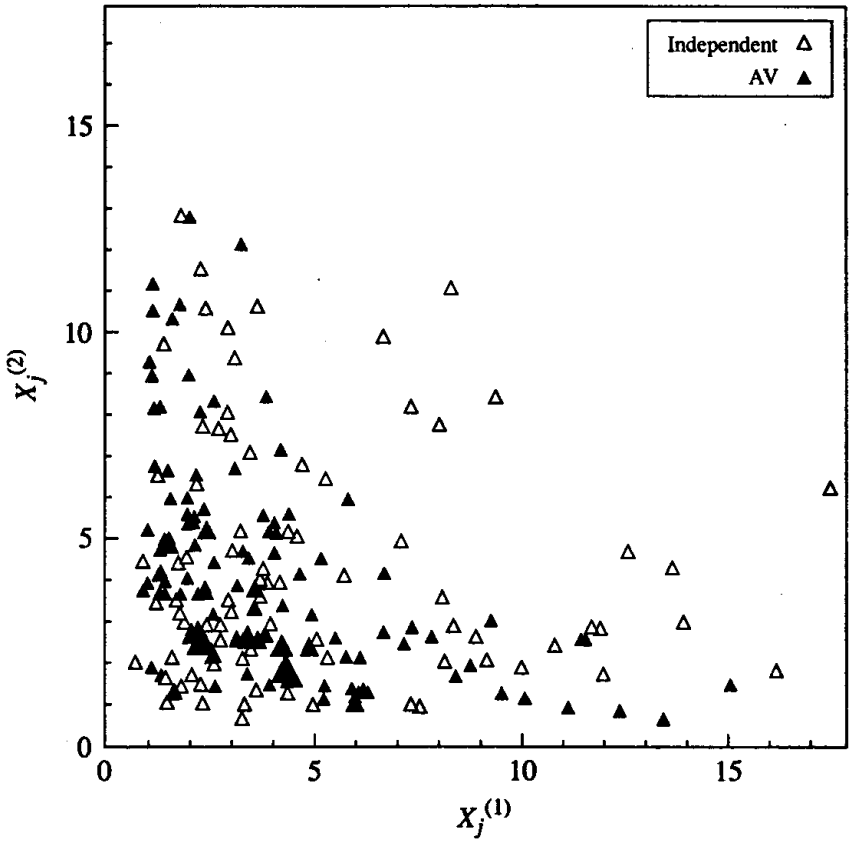


(*b*)

Dot plot of X_j's to show how AV reduces their spread:



Plot of $X^{(2)}$ vs. $X^{(1)}$ to show negative correlation:



Some Issues with AV

Partial AV if difficult or inconvenient to maintain full synchronization:

- Synchronize what you can
- Generate the rest independently
- (Same idea applies to CRN as well)

Same idea, but generate antithetic behavior other than replacing U by $1 - U$:

- Queueing simulation
- Run 1: as usual
- Run 2: reverse use of each U (interarrival vs. service)
- In $M/M/1$, got 65% variance reduction with this idea
- Clearly specialized to models where the basic inputs are themselves antithetic:
  - Big interarrival $\Rightarrow$ low congestion
  - Big service $\Rightarrow$ high congestion

Generalized AV: *rotation sampling*

When comparing two systems, use CRN and AV together?

	Primary run		Antithetic run
Alternative 1	$X_1^{(1)}$	$\leftarrow$ AV $\rightarrow$	$X_1^{(2)}$
	$\uparrow$		$\uparrow$
	CRN		CRN
	$\downarrow$		$\downarrow$
Alternative 2	$X_2^{(1)}$	$\leftarrow$ AV $\rightarrow$	$X_2^{(2)}$

- Assume that both AV and CRN are working properly "on their own"
- Does this guarantee variance reduction in overall estimate of difference, in comparison with what we'd get with independent sampling throughout?

11.4 Control Variates

Suppose X = output random variable of interest, want to estimate $m = E(X)$

Intuition and Implementation

Let Y be another random variable involved in some way in the simulation that is thought to be correlated (+ or –) with X

Also: We must know (exactly) the value of $n = E(Y)$

If X is a congestion measure for a queue, then Y could be:

Average of a fixed number of service times (positive correlation?)

Average of a fixed number of interarrival times (negative?)

Result of simulating a similar but simpler system (with CRN), for which exact queueing-theoretic results are available (positive?)

Run the simulation(s), observe X and Y —suppose $\text{Cor}(X, Y) > 0$

If $Y > n$ (which we can tell since we know n), then suspect that $X > m$ as well, so adjust X downward

If $Y < n$, then suspect that $X < m$ as well, so adjust X upward

(Adjustments are in the opposite direction if $\text{Cor}(X, Y) < 0$)

This adjustment pulls X in toward m from run to run, reducing variability

Thus, Y is used to adjust X, or partially *control* it, so Y is called a *control variate* for X

Probabilistic Rationale

Use *controlled estimator* $X_C = X - a(Y - n)$ for some value a

$E(X_C) = m$, so it's unbiased for m, as is X; hope that $\text{Var}(X_C) < \text{Var}(X)$

The parameter a will have same sign as $\text{Cor}(X, Y)$ — adjustment is in proper direction

The controlled estimator has another random component involving the random variable Y, so the adjustment must compensate for this additional variation (and then some):

$$\text{Var}(X_C) = \text{Var}(X) + a^2\text{Var}(Y) - 2a\ \text{Cov}(X, Y),$$

so get a variance reduction if and only if $2a\ \text{Cov}(X, Y) > a^2\text{Var}(Y)$

Key questions:

What is a? How do we choose it?

Where do we get the control variate(s)?

Extend to multiple control variates $Y_1, Y_2, ..., Y_m$?

Pick $a = \pm 1$?

Suppose we know the sign of $\text{Cor}(X, Y)$, and pick $a = \begin{cases} +1 & \text{if } Cor(X,Y) > 0 \\ -1 & \text{if } Cor(X,Y) < 0 \end{cases}$

Then get variance reduction if and only if $|\text{Cov}(X, Y)| > \text{Var}(Y)/2$, placing entire burden for success on finding a "powerful" enough control variate Y

Pick a Optimally?

View Var(X_C) as a function of a; pick a to minimize it:

let $g(a) = \text{Var}(X_C) = \text{Var}(X) + a^2\text{Var}(Y) - 2a\,\text{Cov}(X, Y)$

Set $g'(a) = 2a\,\text{Var}(Y) - 2\text{Cov}(X, Y) = 0$

Solve for variance-minimizing value $a^* = \text{Cov}(X, Y) / \text{Var}(Y)$

With optimal a^*,

$$\text{Var}(X_C^*) = \text{Var}(X) - [\text{Cov}(X, Y)]^2/\text{Var}(Y) = (1 - r_{XY}^2)\text{Var}(X) \le \text{Var}(X),$$

where $r_{XY}^2 = [\text{Cor}(X, Y)]^2 \ge 0$

So a variance reduction is *guaranteed* regardless of how weak a controller Y might be (i.e., how close Cor(X, Y) is to 0)

In fact, with optimal a^*, for stronger controllers Y, Cor(X, Y) $\to \pm 1$ and so Var(X_C^*) $\to 0$ (a nice situation indeed!!)

But in reality, we won't be able to find the optimal a^*:

May or may not know Var(Y)

Certainly will not know Cov(X, Y) since X is the simulation output r.v.

Must estimate a^* from data somehow:

Let $\hat{C}_{XY}(n)$ and $S_Y^2(n)$ be the usual sample estimators of Cov(X, Y) and Var(Y), respectively, from n replications of the simulation

Estimate a^* by $\hat{a}^*(n) = \hat{C}_{XY}(n)/S_Y^2(n)$ (use Var(Y) on bottom, if known)

Note that $\hat{C}_{XY}(n)$ and $\hat{C}_{XY}(n)$ are strongly consistent for Cov(X, Y) and Var(Y), respectively, so $\hat{a}^*(n)$ is strongly consistent for a^* as well

Final (feasible) controlled point estimator from the n replications:

$$\overline{X}_C^*(n) = \overline{X}(n) - \underbrace{\hat{a}^*(n)}_{\text{now a r.v.}}[\overline{Y}(n) - n]$$

The bad news: Since $\hat{a}^*(n)$ is a r.v. not independent of $\overline{Y}(n)$, $\overline{X}_C^*(n)$ is no longer unbiased for m; some remedies (each has drawbacks):

Splitting the sample

Jackknifing

Hoping the bias is more than offset by variance reduction (MSE??)

Multiple Controllers

May have several choices for Y

$Y^{(1)}$ = mean of interarrival times

$Y^{(2)}$ = mean of service times

$Y^{(3)}$ = proportion of parts failing at an inspection station

Define the multiple-control-variate estimator $X_C = X - \sum_{l=1}^{m} a_l (Y^{(l)} - \nu^{(l)})$, where

$E(Y^{(l)}) = \nu^{(l)}$ (which must be known)

Allowing all possible dependencies (between control variates and X, as well as one control variate and another),

$$\mathrm{Var}(X_C) = \mathrm{Var}(X) + \sum_{l=1}^{m} a_l^2 \, \mathrm{Var}(Y^{(l)}) - 2\sum_{l=1}^{m} a_l \, \mathrm{Cov}(X, Y^{(l)}) + 2\sum_{l_1=2}^{m}\sum_{l=1}^{l_1-1} a_{l_1} a_{l_2} \, \mathrm{Cov}(Y_{l_1}, Y_{l_2})$$

Viewing this as a function of the a_l's and minimizing over them, get optimal a_l^*'s similar to single-controller case

Still must estimate the optimal a_l^*'s, still have potential bias problems, same remedies

Finding optimal a_l^*'s turns out to be equivalent to finding the b coefficients in a particular multiple-regression problem, and so CV is sometimes called *regression sampling*

Sources of Control Variates

Typically, control variates (the $Y^{(l)}$'s) come from three sources:

1. *Internal*

   Simple functions of variates generated for input to the simulation
   - Average of interarrival, processing times
   - Proportions of "hard" vs. "easy" jobs generated

   Easiest, lowest-cost — don't have to be very powerful to "pay off"

   Sometimes called *concomitant* since they are there anyway

2. *External*

   Construct a system that is similar to, but simpler than, the one being simulated

   Must be simple enough to admit an analytic solution (for the n)

   Simulate this other system alongside the one of interest, using CRN

   Expect that $\text{Cor}(X, Y) > 0$
   - e.g., system of interest is Uniform/Gamma/1 queue
   - Make the other system $M/M/1$ (known mean performance measures)

   Clearly not costless, so Y must be pretty powerful if this is to pay off

3. *Multiple Point Estimators*

Suppose that for the system of interest we have k different unbiased point estimators $X^{(1)}, ..., X^{(k)}$ for m

(How? Indirect estimation, discussed below, is one way)

Let $b_1, ..., b_k$ be real numbers with $b_1 + \cdots + b_k = 1$ (but b_i's can be < 0)

Then $X_C = \sum_{i=1}^{k} b_i X^{(i)}$ is also unbiased for m

To put this into CV notation, note that $b_1 = 1 - b_2 - \cdots - b_k$, so

$$\begin{aligned} X_C &= (1 - b_2 - \cdots - b_k) X^{(1)} + \sum_{i=2}^{k} b_i X^{(i)} \\ &= X^{(1)} - \left(\sum_{i=2}^{k} b_i \right) X^{(1)} + \sum_{i=2}^{k} b_i X^{(i)} \\ &= X^{(1)} - \sum_{i=2}^{k} b_i (X^{(1)} - X^{(i)}) \end{aligned}$$

so we get $k - 1$ control variates $X^{(1)} - X^{(i)}$ for $i = 2, 3, ..., k$

Somewhat specialized

Must pay the price for computing the extra estimators

Example of Effectiveness of CV

$M/M/1$ queue, X = average of 100 delays in queue, traffic intensity = 0.9

To estimate optimal a^*, made $n = 10$ replications

Variances estimated "externally" by repeating the above 100 times:

Control variate	Variance without CV	Variance with CV	Variance reduction
Avg. service time, $\overline{S}$	0.99	0.66	33%
Avg. interarrival time, $\overline{A}$	0.99	0.89	10%
$\overline{S} - \overline{A}$ (*not* multiple CV)	0.99	0.56	43%

11.5 Indirect Estimation

Mentioned before: Usually want several performance measures

Expected time in queue(s) of parts

Time-average queue length(s)

Machine utilizations(s)

Intuition and Implementation

For some classes of models, there are relationships among these measures that might be used to get better (lower-variance) estimators

Best-known examples: Queueing models

Notation: l = arrival rate

L = Expected time-average number in system

Q = Expected time-average number in queue

w = Expected time a customer spends in system

d = Expected time a customer spends in queue

S = Random variable for service time

Relationships: $L = l w$

$Q = l d$

$w = d + E(S)$

Conservation Equations

These relationships are valid for a *very* wide class of queueing systems

Options for estimating w:

Directly: Collect times in system, average them

Indirectly: Estimate d, then add $E(S)$ (which would be known)

Seems clear: Indirect method is better, since direct method essentially estimates the known value of $E(S)$

Moral: Don't estimate things that you know

Options for estimating Q:

Directly: Time-average via integrating under $Q(t)$ curve: $\overline{Q}$

Indirectly: Estimate d directly ($\overline{d}$), then multiply by l

Fact (but not so clear): $\text{Var}(l\,\overline{d}) < \text{Var}(\overline{Q})$

Also: Better to estimate L indirectly via w, which in turn is best estimated by $d + E(S)$ — estimate L by $l(\overline{d} + E(S))$

Summing up: Get "everything" indirectly via $\overline{d}$

Example of Effectiveness of IA

Exact asymptotic (length of simulation $\to \infty$) variance reductions in using indirect rather than direct estimator for Q (r is the traffic intensity):

	% Variance Reduction		
Service-time distribution	$r = 0.5$	$r = 0.7$	$r = 0.9$
Exponential	15	11	4
4-Erlang	22	17	7
Hyperexponential	4	3	2

Unfortunately, method becomes weaker as r increases (just when you need variance reduction the most)

This can be remedied by looking at optimally-weighted combinations of all the estimators, in a way reminiscent of CV (details, references in text)

11.6 Conditioning

Use knowledge of "parts" of output, rather than estimating them (similar in spirit to indirect estimation)

Output random variable X, want to estimate $m = E(X)$

Suppose there is another random variable Z such that if we know the value of Z, we'd know the expected value of X for sure

i.e., the *conditional expectation* $E(X \mid Z = z)$ is a known, deterministic function of z

Example: Single-server queueing system

X is a delay in queue of an arriving customer

Service times are exponential (so memoryless) with mean $E(S)$

Let Z = number of customers already in queue when customer arrives

Then $E(X \mid Z = z) = (z + 1)E(S)$

For simplicity, suppose Z is discrete with mass function $p(z) = P(Z = z)$

Partitioning over the space of Z,

$$m = E(X) = E_Z[E(X \mid Z)] = \sum_z E(X \mid Z = z)\, p(z)$$

Know: $E(X \mid Z = z)$ components

Don't know: $p(z)$ components, so simulate just Z to estimate them

Procedure: Simulate to observe a value z for Z, then compute $E(X \mid Z = z)$ to serve as a "basic observation"

Single-server queue example: Use simulation to generate values z for number in queue, then tally values of $(z + 1)E(S)$

Variance advantage:

$$\begin{aligned}\mathrm{Var}_Z\left[E(X \mid Z)\right] &= \mathrm{Var}(X) - E_Z\left[\mathrm{Var}(X \mid Z)\right] \\ &\le \mathrm{Var}(X)\end{aligned}$$

Trick: Specify the "right" Z — want:

Ease of generating Z (still have to simulate it)

Ease of computing the known deterministic function $E(X \mid Z = z)$, any z

$E_Z\left[\mathrm{Var}(X \mid Z)\right]$ to be large (see equation) — we never have to simulate $X|Z$ so don't care how big its variance is

Issues:

Clearly need to understand model, exploit special structure

Often get as a by-product the phenomenon of artificially increasing the occurrence of rare but important events

Examples: (details in text)

Branch points in computer-performance model, condition on branching decision and queue length at next station decided upon

Estimate response time of fire trucks to serious fires, condition on location of fire trucks