Optimization and Simulation
Simulation Exercise 4: Variance Reduction

Bilal Farooq

April 12, 2013

TRANSP-OR

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

The Exercise: Statistical Analysis and Bootstrapping

- 4.1: Antithetic Draws
- 4.2: Control Variates
- 4.3: Analyze the Sioux Falls with 4.2
  - 4.3.1: Extension of the simulation
  - 4.3.2: Compute expectations of control variates
  - 4.3.3: Apply control variates

3.1 Antithetic Draw

- Implement the function `AntitheticDraws.m`
  - You will need to use function handles
- Test your implementation with `AntitheticDrawsTest.m`
  - Play around with different functions

3.2 Control Variates

- Implement the function `ControlledMean.m`
  - Use MATLAB functions: `mean`, `var`, and `cov`
- Test your implementation with `ControlledMeanTest.m`
  - Play around with the different parameters

4.3 Overview

- Find expected max. queue size on any road in Sioux Falls
- Control with total number of car passages across that road
- Solve this step by step
  - 4.3.1: Extract control variates from simulation
  - 4.3.2: Compute expectations of control variates
  - 4.3.3: Analyze the Sioux Falls Network

4.3.1 Extract Control Variates

- So far, your queueing simulation was `function [times, queues] = QueueingSimulation2(scenario)`
- Extend this into `function [times, queues, totals] = QueueingSimulation2(scenario)`
  - Where `totals` is a row vector that contains for each link the total number of vehicles that have passed that link during entire simulation
- `totals(i)` equals the number of `DEPARTURE` events on `i`

4.3.2 Expectations of Control Variates

- μ_i is total number of vehicles passing road i
- μ_i is the product of two terms:
  - Expected number of passages by a single random car
  - Expected total number of vehicles in the system

4.3.2 Expectations of Control Variates

- N is number of links in the network
- T is duration of demand entry period
- λ_i is external arrival rate on link i
- p_{ij} is probability to turn from link i into link j
- p_{i*} is probability to leave the network from link i
- π_{jk} is probability that the kth link of a random car trip is j
- π_{*k} is probability of leaving the network after having visited $(k–1)$ links

4.3.2 Expectations of Control Variates

- π_{jk} can be recursively expressed as

$$\pi_{jk} = \begin{cases} \lambda_j / \sum_{i=1}^{N} \lambda_i & k = 1 \\ \sum_{i=1}^{N} \pi_{i(k-1)} \cdot p_{ij} & k > 1 \end{cases}$$

- Writing $\pi_k = [\pi_{1k} \pi_{2k} \ldots \pi_{Nk} \pi_{*k}]$, we obtain

$$\boldsymbol{\pi}_k = \begin{cases} [\lambda_1 \ \lambda_2 \ \cdots \ \lambda_N \ 0] / \sum_{i=1}^{N} \lambda_i & k = 1 \\ \boldsymbol{\pi}_{k-1} \begin{bmatrix} p_{11} & \cdots & p_{1N} & p_{1*} \\ \vdots & \ddots & \vdots & \vdots \\ p_{N1} & \cdots & p_{NN} & p_{N*} \\ 0 & \cdots & 0 & 0 \end{bmatrix} & k > 1 \end{cases}$$

4.3.2 Expectations of Control Variates

- Let $\boldsymbol{\varrho} = [\varrho_1 \;\; \varrho_2 \;\; \cdots \;\; \varrho_N \;\; \varrho_{N+1}] = \sum_{k=1}^{\infty} \boldsymbol{\pi}_k$
  - ϱ_1 is expected number of times a car visits link i, $i = 1 \ldots N$
  - ϱ_{N+1} is expected number of times a car leaves the network
- Expected total number of vehicles in the system

$$T \sum_{i=1}^{N} \lambda_i$$

  - Expected number of passages across link i

$$\mu_i = \varrho_i T \sum_{i=1}^{N} \lambda_i$$

4.3.2 Expectations of Control Variates

1. construct $P = \begin{bmatrix} p_{11} & \cdots & p_{1N} & p_{1*} \\ \vdots & & \vdots & \vdots \\ p_{N1} & \cdots & p_{NN} & p_{N*} \\ 0 & \cdots & 0 & 0 \end{bmatrix}$
2. initialize $\boldsymbol{\pi} = [\lambda_1\ \lambda_2\ \cdots\ \lambda_N\ 0] / \sum_{i=1}^{N} \lambda_i$
3. initialize $\boldsymbol{\varrho} = \boldsymbol{\pi}$
4. iterate until $\|\boldsymbol{\pi}\| < 10^{-9}$:

   4.1 $\boldsymbol{\pi} \leftarrow \boldsymbol{\pi} P$

   4.2 $\boldsymbol{\varrho} \leftarrow \boldsymbol{\varrho} + \boldsymbol{\pi}$

- let $\mu_i = \varrho_i T \sum_{i=1}^{N} \lambda_i$ for $i = 1 \ldots N$

4.3.2 Expectations of Control Variates

- Implement `ExpectedPassages.m`
- Test it with `ExpectedPassagesTest.m`
  - You can either run many simulation
  - Or load existing simulation results from file
    - Provided files: linkTotals.dat, maxQueues.dat

4.3.3 Apply Control Variates

- Now apply your Control Variates
- The necessary implementation is already available
  - Just run `SiouxFallsControlVariatesTest.m`
    - Understand the working
    - Try different types of analysis