FDSA and SPSA in Simulation-Based Optimization

• Stochastic approximation provides ideal framework for carrying out simulation-based optimization
  – Rigorous means for handling noisy loss information inherent in Monte Carlo simulation: \( y(\theta) = Q(\theta, V) = L(\theta) + \text{noise} \)
  – Most other optimization methods (GAs, nonlinear programming, etc) apply only on \textit{ad hoc} basis

• “…FDSA, or some variant of it, remains the method of choice for the majority of practitioners” (Fu and Hu, 1997)
  – No need to know “inner workings” of simulation, as in gradient-based methods such as IPA, LR/SF, etc.

• FDSA and SPSA-type methods much easier to use than gradient-based method as they only require simulation inputs/outputs
• Common random numbers (CRNs) provide a way for improving simulation-based optimization by reusing the Monte-Carlo-generated random variables

• CRNs based on the famous formula for two random variables $X$, $Y$:

$$\text{var}(X - Y) = \text{var}(X) + \text{var}(Y) - 2\text{cov}(X, Y)$$

• Maximizing the covariance minimizes the variance of the difference

• The aim of CRNs is to reduce variability of the gradient estimate

  $\Rightarrow$ Improves convergence in algorithm
CRNs (cont’d)

• For SPSA, the gradient variability is largely driven by the numerator \( y(\hat{\theta}_k + c_k \Delta_k) - y(\hat{\theta}_k - c_k \Delta_k) \)

• Two effects contribute to variability:
  (i) difference due to perturbations \( \pm c_k \Delta_k \) (desirable)
  (ii) difference due to noise effects in measurements (undesirable)

• CRNs useful for reducing undesirable variability in (ii)

• Using CRNs maximizes covariance between two \( y(\cdot) \) values in numerator

  \[ \Rightarrow \text{Minimizes} \text{ variance of difference} \]
CRNs (cont’d)

- In simulation (vs. most real systems) some form of CRNs is often feasible
- The essence of CRN is to use same random numbers in both \( y(\hat{\theta}_k + c_k\Delta_k) \) and \( y(\hat{\theta}_k - c_k\Delta_k) \)
  - Achieved by using same random number seed for both simulations and synchronizing the random numbers
- Optimal rate of convergence of iterate to \( \theta^* \) (à la \( k^{-\beta/2} \)) is \( k^{-1/2} \) (Kleinman et al., 1999); this rate is same as stochastic gradient-based method
  - Rate is improvement on optimal non-CRN rate of \( k^{-1/3} \)
- Unfortunately, “pure CRN” may not be feasible in large-scale simulations due to violating synchronization requirement
  - e.g., if \( \theta \) represents service rates in a queuing system, difference between \( \hat{\theta}_k + c_k\Delta_k \) and \( \hat{\theta}_k - c_k\Delta_k \) may allow additional (stochastic) arrivals to be serviced in one case
Numerical Illustration (Example 14.8 in /SSO)

• Simulation using exponentially distributed random variables and loss function with $p = \dim(\theta) = 10$
• Goal is to compare CRN and non-CRN
• $\theta^*$ is minimizing value for $L(\theta)$
• Table below shows improved accuracy of solution under CRNs; plot on next slide compares rate of convergence

<table>
<thead>
<tr>
<th>Total Iterations $n$</th>
<th>$\frac{|\hat{\theta}_n - \theta^<em>|}{|\hat{\theta}_0 - \theta^</em>|}$</th>
<th>CRNs</th>
<th>Non-CRNs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>0.02195</td>
<td>0.04103</td>
<td></td>
</tr>
<tr>
<td>10,000</td>
<td>0.00658</td>
<td>0.01845</td>
<td></td>
</tr>
<tr>
<td>100,000</td>
<td>0.00207</td>
<td>0.00819</td>
<td></td>
</tr>
</tbody>
</table>
Rates of Convergence for CRN and Non-CRN (Example 14.9 in ISSO)

Mean Values of $n^{\beta/2}(\hat{\theta}_n - \theta^*)$

- CRN, $\beta = 1$
- Non-CRN, $\beta = 1$
- Non-CRN, $\beta = 2/3$
Partial CRNs

- By using the same random number seed for \( y(\hat{\theta}_k + c_k \Delta_k) \) and \( y(\hat{\theta}_k - c_k \Delta_k) \) it is possible to achieve a partial CRN.

- Some of the events in the simulations will be synchronized due to common seed.
  - Synchronization is likely to break down during course of simulation, especially for small \( k \) when \( c_k \) is relatively large.

- Asymptotic analysis produces convergence rate identical to pure CRN since synchronization occurs as \( c_k \to 0 \).
  - Also require new seed for simulations at each iteration (common for both \( y(\bullet) \) values) to ensure convergence to \( \min_{\theta} L(\theta) = \min_{\theta} E[Q(\theta, V)] \).

- In partial CRN, practical finite sample rate of convergence for SPSA tends to be lower than in pure CRN setting.
Numerical Example: Partial CRNs
(Kleinman et al., 1999; see p. 398 of ISSO)

• A simulation using exponentially distributed random variables was conducted in Kleinman, et al. (1999) for $p = 10$
  
  – Simulation designed so that it is possible to implement pure CRN (not available in most practical simulations)

• Purpose is to evaluate relative performance of non-CRN, partial CRN, and pure CRN
Numerical Example (cont’d)

- Numerical Results for 100 replications of SPSA and FDSA (no. of $y(\bullet)$ measurements in SPSA and FDSA are equal with total iterations of 10000 and 1000 respectively):

<table>
<thead>
<tr>
<th>CRN Type</th>
<th>$|\hat{\theta}_{10000}^{SPSA} - \theta^*|$</th>
<th>$|\hat{\theta}_{1000}^{FDSA} - \theta^*|$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-CRN</td>
<td>0.0190</td>
<td>0.0410</td>
</tr>
<tr>
<td>Partial CRN</td>
<td>0.0071</td>
<td>0.0110</td>
</tr>
<tr>
<td>Pure CRN</td>
<td>0.0065</td>
<td>0.0064</td>
</tr>
</tbody>
</table>

- Partial CRN offers significant improvement over non-CRN and SPSA outperforms FDSA (except in idealized pure CRN case)