Proseminar Technische Informatik
Rare Event Simulation

Betreuerin: PD Dr. Katinka Wolter
Autor: Jan Sydow

14. Januar 2010

1 Abstract

Bei Simulationen sind häufig die seltenen Ereignisse die, die von besonderem Interesse sind. Mit naiver Simulation würde es häufig zu lange dauern um ausreichende Erkenntnisse über solche Ereignisse zu gewinnen. Man bedient sich deshalb spezieller Methoden um mit weniger Simulationszeit eine ausreichende Anzahl seltener Ereignisse zu erzeugen. Die zwei wichtigsten Methoden sind Importance Sampling und Importance Splitting. Mit Importance Sampling versucht man die Wahrscheinlichkeit des Auftretens seltener Ereignisse zu erhöhen. Beim Importance Splitting wird der Simulationsprozess an Zwischenzuständen zu seltenen Ereignissen aufgesplittet um die Menge auftretender, Ausnahmefreier Zustände zu minimieren. Dabei können Ergebnisverfälschungen vorkommen, die gegengerechnet werden müssen. Hier sollen die Methoden vorgestellt und verglichen werden.

2 Einleitung

Seltene Ereignisse in Simulationen, sind Ereignisse mit Auftrittswahrscheinlichkeiten von 10^{-9} bis 10^{-10}. Meist sind es Ausnahmesituationen, die ohnehin, so selten wie möglich auftreten sollen. Dennoch oder gerade deshalb sind meist gerade diese seltenen Ereignisse bei Simulationen von Interesse. Um verlässliche Aussagen über das Ereignis treffen zu können, werden viele unabhängige Auftritte des Ereignisses benötigt. Mit naiver Simulation, dauert es häufig viel zu lange auch nur ein, geschweige denn hundert, solcher Ereignisse zu erhalten. Deswegen wird hier Rare Event Simulation benötigt um schneller an verlässliche Aussagen zu kommen.

Beispiele für seltene Ereignisse in der Informatik sind der Ausfall ausfallsicherer Systeme oder Pufferüberläufe in Netzwerkqueues, wie Zellverluste in ATM-Netzwerkqueues.

Zellverluste, die durch Überlauf eines endlich großen Netzwerkbuffers entstehen, sollen eine Auftrittswahrscheinlichkeit von 10^{-9} oder weniger haben. Da also hier die Simulation ziemlich ineffizient wäre, wird Rare Event Simulation verwendet [10].

Rare Event Simulation wird aber nicht nur in der Informatik verwendet. Finanzmathematiker benutzen Rare Event Simulation um Versicherungsrisiken, wie Ruins zu bewerten. Für Biologen und Chemiker kann die Wahrscheinlichkeit der Reaktion zweier Moleküle ein seltenes Ereignis sein und Erdbeben zum Beispiel sind für Geologen seltene Ereignisse von Interesse.

Die zwei wichtigsten Rare Event Simulationsverfahren sind Importance Splitting und Importance Sampling. Mit Importance Splitting wird das System in Richtung seltenes Ereignis getrieben. Vielversprechende Simulationsläufe werden weiterverfolgt, weniger vielversprechende werden abgebrochen. Bei Importance Sampling wird das zu simulierende System so verändert, dass man weniger Ereignisse benötigt um eine gewisse Genauigkeit zu erzielen, und diese auch schneller erhält.

Zur mathematischen Beschreibung von Abläufen werden stochastische Prozesse verwendet. Stochastische Prozesse sind eine Menge auch $\{X(t) : t \in T\}$ von Zufallsvariablen über einem Wahrscheinlichkeitsraum $(\Omega, \mathcal{F}, P)$, wobei t ein Parameter über dem Parameterraum T ist, meistens die Zeit. X_t ist eine Abbildung $X_t : \Omega \mapsto Z$, wobei der Wertebereich Z als Zustandsraum bezeichnet wird und Ω die Ereignismenge ist. Eine Folge von Zuständen $X(t \geq 0)$ heißt Pfad und ist eine Teilmenge von Z^T [20].

Im Folgenden werden in Abschnitt 3 die beiden häufigsten Simulationsmethoden vorgestellt. Danach sollen die Methoden an einem Beispiel veranschaulicht werden in Abschnitt 4. In Abschnitt 5 dann werden diese kurz verglichen und ein Ausblick gegeben.

3 Methoden für Rare Event Simulation

3.1 Importance Sampling

Betrachten wir das Problem mathematisch:
Sei Z eine Zufallsgröße mit einer Wahrscheinlichkeit $p(\cdot)$ über dem Ereignisraum Ω. Seien darin $R \in \Omega$ die seltenen Ereignisse. Dann ist eine interessante Größe

$$\gamma = p(R) = E_p(1_{z \in R})$$

die Wahrscheinlichkeit, das eines der seltenen Ereignisse auftritt. Hier ist $p(\cdot)$ eine Abbildung $p : \Omega \mapsto [1, 0]$, die die Wahrscheinlichkeit eines Ereignisses angibt. $E_p(\cdot)$ ist der Erwartungswert unter $p(\cdot)$.

Oft wird auch noch ein Seltenheitsparameter ϵ benutzt, für den gilt:

$$\epsilon \to 0 \Rightarrow \gamma \to 0$$

ϵ könnte zum Beispiel die maximale Ausfallrate in einem ausfallsicheren System sein. Bei Warteschlangen könnte $\epsilon = \text{Queuelänge}^{-1}$ sein, wenn das seltene Ereignis ein Überlauf der Queue ist.

Mit Standardsimulation sieht das Ganze dann so aus: Man simuliert das System n mal und erhält damit $Z_1 \dots Z_n$ Ergebnisse. Damit erhält man dann folgende Schätzung für γ:

$$\hat{\gamma} \equiv \frac{\sum_{i=1}^{n} 1_{\{Z_i \in R\}}}{n} \tag{1}$$

Möchte man damit eine bestimmte, konstante Anzahl seltener Ereignisse erhalten, so muss man öfter simulieren, wenn $\hat{\gamma}$ kleiner wird: $\hat{\gamma} \to 0 \Longrightarrow n \to \infty$. Wie in [17] gezeigt, gilt für den dazugehörigen relativen Fehler RE_p:

$$\epsilon \to 0 \Rightarrow RE_p \to \infty$$

Je niedriger ϵ also wird, desto mehr Ergebnisse n benötigt man, um einen bestimmten relativen Fehler zu erhalten. Da bei seltenen Ereignissen dieses ϵ in der Regel sehr klein ist, wird mit Standardsimulierung der Simulationsaufwand enorm.

Importance Sampling ist nun ein Verfahren, das die Varianz der Ergebnisse verringern soll. Dadurch wird der relative Fehler geringer und es wird insgesamt weniger Simulationsaufwand benötigt um Ergebnisse mit gleicher Sicherheit zu bekommen. Es gilt also ein neues Wahrscheinlichkeitsmaß $p'(\cdot)$ zu finden, sodass gilt:

$$\forall z \in R: \; p(z) > 0 \Rightarrow p'(z) > 0$$

Dann gilt für γ:

$$\gamma = \sum_{z \in \Omega} 1_{\{z \in R\}} \cdot p(z) \tag{2}$$

$$= \sum_{z \in \Omega} 1_{\{z \in R\}} \cdot p'(z) \cdot \frac{p(z)}{p'(z)} \tag{3}$$

$$= \sum_{z \in \Omega} 1_{\{z \in R\}} \cdot p'(z) \cdot L_{p'}(z) \tag{4}$$

$L_{p'}(\cdot)$ ist hier der so genannte Likelihood ratio, der definiert ist als:

$$L_{p'} = \begin{cases} \frac{p(z)}{p'(z)} & \text{, wenn } p'(z) > 0 \\ 0 & \text{, sonst} \end{cases}$$

Dieses Verhältnis gibt an, um welchen Faktor das Ergebnis verfälscht wurde. Zum Gegenrechnen der Ergebnisverfälschung muss das Ergebnis nur mit dem Likelihoodratio multipliziert werden wie in (4).

Damit ergibt sich für n Versuche folgender Schätzwert für n Simulationen:

$$\hat{\hat{\gamma}} \equiv \frac{\sum_{i=1}^{n} 1_{\{Z_i \in R\}} \cdot L_{p'}(z)}{n}$$

Dieser ist jetzt durch den Einsatz von $L_{p'}(\cdot)$ und $p'(\cdot)$ unverfälscht.

3.1.1 Varianzreduktion

Beim Importance Sampling sucht man allgemein eine neue Verteilung $p'(\cdot)$, die weniger Varianz und einen geringeren relativen Fehler aufweist. Damit werden weniger Simulationswerte n benötigt um einem bestimmten Konfidenzintervall zu genügen. Oder anders herum: Bei gleicher Simulationsanzahl erhöht sich die Genauigkeit des Schätzwertes. Das Konfidenzintervall ist ein Intervall, in dem zu einer gegebenen Wahrscheinlichkeit $(1-\alpha)100\%$ (z.B. 95%) der wahre Wert liegt. Das Konfidenzintervall hängt vom Standardfehler ab, welcher von der Stichprobengröße abhängt [12].

Häufig ist mit der neuen Verteilung ein erhöhter Rechenaufwand verbunden. Um einen Simulationswert zu erhalten muss dann unter Umständen länger gerechnet werde. Jedoch nimmt man diesen zusätzlichen Rechenaufwand in Kauf, wenn die Varianzreduktion sich in viel höheren Dimensionen bewegt, als der zusätzliche Rechenaufwand.

Wie man durch Importance Sampling die Varianz reduzieren kann, so kann man bei Wahl einer ungünstigen neuen Verteilung $p'(\cdot)$ eine Erhöhung der Varianz erreichen. Damit dies nicht passiert, ist im Allgemeinen eine genaue Kenntnis des zu simulierenden Systems erforderlich.

Es ist theoretisch möglich eine Varianz von 0 zu erreichen. Das heißt, dass jede Simulation immer einen konstanten Wert liefert. Damit ist dann immer $1_{\{Z_i \in R\}} = 1$ und $L_{p'} = \gamma$ [1]. Es gilt für alle $A \in R$:

$$p'(A) = \frac{p(A)}{\gamma}$$

und für alle $B \notin R$:

$$p'(B) = 0$$

Wie man sieht wird für eine Bestimmung von $p'(\cdot)$ die Kenntnis von γ benötigt. Dies ist jedoch meist die gesuchte Größe. Deswegen ist diese Zero-Variance Eigenschaft im Allgemeinen nicht implementierbar.

Praxisrelevanter ist die Bounded Relative Error Eigenschaft. Sei hierzu $(\mathcal{E}_b : b \geq 1)$ eine Sequenz von Ereignissen, mit den dazugehörigen Wahrscheinlichkeiten $\gamma_b = P(\mathcal{E})$, wobei b der Seltenheitsparameter ist, sodass gilt:

$$b \to \infty \Rightarrow \gamma_b \to 0$$

Die Ereignisse werden also mit ansteigendem b immer seltener. Sei dann $(Z_b : b \geq 1)$ ein stochastischer Prozess unter dem Wahrscheinlichkeitsmaß p'. Z_b soll ein unverfälschter

Schätzwert für γ_b sein. Dann hat $(Z_b : b \geq 1)$ genau dann die Bounded Relative Error Eigenschaft, wenn gilt:

$$\limsup_{b\to\infty} \frac{\sigma_{p'}(Z_b)}{\gamma_b} \leq \infty$$

wobei $\sigma_{p'}$ die Standardabweichung und $\frac{\sigma_{p'}(Z_b)}{\gamma_b}$ der relative Fehler ist [1]. Dies bedeutet, dass es eine obere Schranke für die Anzahl n der Simulationen gibt um einen bestimmten relativen Fehler zu erreichen, egal wie klein γ_b auch ist. Damit ist der Rechenaufwand begrenzt.

Auch wenn man versucht möglichst die Bounded Relative Error Eigenschaft zu erreichen, ist dies oft nicht der Fall. Deshalb gibt es noch eine andere schwächere Eigenschaft, nämlich asymptotisch optimal oder auch asymptotisch effizient. Aus der Ungleichung $E_{p'}(Z_b^2) \geq \gamma_b^2$ folgt

$$\frac{\log(E_{p'}(Z_b^2))}{\log \gamma_b} \leq 2$$

Gilt diese Ungleichung für $b \to \infty$ so spricht man davon, dass $p'(\cdot)$ optimal auf logarithmischer Ebene ist, also asymptotisch optimal [1].

3.1.2 Anwendung

Betrachten wir den stochastischen Prozess $\{X(s) : s \geq 0\}$, wobei $s \in S$ ein Parameter über dem Zustandsraum S ist. s kann meist als Zeiteinheit angesehen werden. Weiterhin unterteilen wir S in $S = G \cup B$. Seien B hier die seltenen Zustände, die von Interesse sind und $G = S \backslash B$ alle anderen Zustände. Angenommen der Prozess erreicht einen stabilen Zustand, d.h. $s \to \infty \Longrightarrow X(s) \to X_\infty$ [17]. Dann ist

$$\alpha = E(f(X_\infty))$$

von Interesse, wobei $f(x) = 1_{\{x \in B\}}$. Dies ist auf lange Sicht der Anteil der Zeit, die der Prozess in B verbringt. Manchmal will man auch nur bestimmen wie lange der Prozess im Zeitintervall $[0, t]$ in B verbringt, wobei der Zeitpunkt 0 hier ein Zeitpunkt ist, an dem sich das System in einem stabilen Zustand befindet. Hierfür gilt dann:

$$\alpha(t) = E\left(\frac{\int_{s=0}^{t} f(X(s))ds}{t}\right)$$

Man ist auch an der durchschnittlichen Zeit zwischen zwei B-Besuchen interessiert (β). Wenn τ_B die erste Zeit ist, in der der Prozess in B ist, dann ist auch $E(\tau_B)$ und $P(\tau_B < t)$ von Interesse, wobei $P(\tau_B < t)$ die Wahrscheinlichkeit ist, dass das seltene Ereignis vor t eintrifft.

Betrachten wir den Fall, dass der Prozess regenerativ ist, d.h. es gibt einen Regenerationszeitpunkt X_t, sodass der Prozess davor und danach stochastisch unabhängig voneinander ist und die gleiche Verteilung aufweisen. Bei Queues könnte dies der Fall

sein, wenn die Schlange leer ist. Sei τ die Zeit, die der Prozess zwischen 2 Besuchen des gleichen regenerativen Zustands verbringt. Weiterhin sei W die Zeit, die der Prozess in τ in B verbringt. Dann lässt sich α bestimmen durch:

$$\alpha = \frac{E(W)}{E(\tau)}$$

$E(\tau)$ ist meist kein kleiner Wert und damit auch mit naiver Simulation leicht bestimmbar. Für $E(W)$ jedoch muss Importance Sampling angewendet werden, weil W in den meisten Versuchen 0 ist. Sei dafür Z ein Pfad im stochastischen Prozess mit der dazugehörigen Wahrscheinlichkeit $P(\cdot)$, dann sei R die Menge aller Pfade, die B besuchen und $W(Z) \equiv W$. Damit ist

$$E(W) \equiv E_p(W(Z)) = E_p(W(Z)1_{\{Z \in R\}})$$

Mit Importance Sampling wird nun eine neue Wahrscheinlichkeit $p'(\cdot)$ benutzt, sodass mehr R-Pfade erzeugt werden. Wird während eines Prozesses B erreicht, so muss allerdings wieder $p(\cdot)$ eingesetzt werden [17]. Angenommen zum Zeitpunkt $t = 0$ befindet sich das System in einem regenerativen Zustand, dann ist

$$E(\tau_B) = \frac{E(\tau_{min})}{P(R)}$$

τ_{min} ist hier die Zeit, nach der entweder B oder der regenerative Zustand erreicht wird. Dies ist leicht bestimmbar, da die regenerativen Zustände meist nicht selten sind. Für $P(R)$ muss allerdings Importance Sampling angewendet werden.

Angenommen nun, dass der regenerative Zyklus sehr lange dauert, dann können wir dieses System als nicht-regenerativ betrachten. Nicht-regenerative Systeme kann man jedoch ähnlich behandeln, wie regenerative Systeme. Sei dazu A ein Zustand der häufig besucht wird. Dann ist ein A-Zyklus die Zeit zwischen zwei S-Besuchen. Damit gilt weiterhin das Verhältnis

$$\alpha = \frac{E(W)}{E(\tau)},$$

wobei dann W die Zeit ist, die der Prozess in einem A-Zyklus in B verbringt. τ ist die Zeit, eines A-Zyklus. Nun existiert jedoch das Problem, dass zwei aufeinanderfolgende A-Zyklen oft nicht stochastisch unabhängig sind. Deswegen läuft hier die Simulation ein klein wenig anders ab:

Man lässt den Prozess erst ein paar A-Zyklen lang einlaufen und splittet dann bei jedem A-Zyklenstart den Prozess auf. Der abgesplittete Prozess wird mit der gemäß Importance Sampling veränderten Wahrscheinlichkeiten weiter simuliert um $W(Z)$ und $L_{p'}(Z)$ zu erhalten, während der Originalprozess weitersimuliert wird um einen Wert für $\tau(Z)$ zu bekommen.

β erhält man durch:

$$\beta = \frac{E(\tau)}{E(N)}$$

wobei wieder $E(\tau)$ leicht mit Standardsimulation zu bestimmen ist. $E(N)$ erhält man wieder nur durch Importance Sampling, wobei N die Anzahl der B-Besuche in einem A-Zyklus ist [6].

3.2 Importance Splitting

Bei naiver Simulation werden meist die Zustandsgebiete simuliert, die weit weg vom seltenen Ereignis B sind. Zum Beispiel will man in einer Netzwerkqueue Pufferüberläufe simulieren. Wenn jedoch in einer Warteschlange die Servicerate der Pakete viel höher ist als die Ankunftsrate, dann wird der Puffer meist leer oder wenig befüllt sein. Man befindet sich weit weg von einer vollen Queue und verschwendet damit Rechenzeit (Abb. 1, blaue Pfade). Viel lieber möchte man das System n-mal simulieren und dabei erfolgversprechende Simulationen weiterverfolgen (roter Pfad) und die anderen beenden.

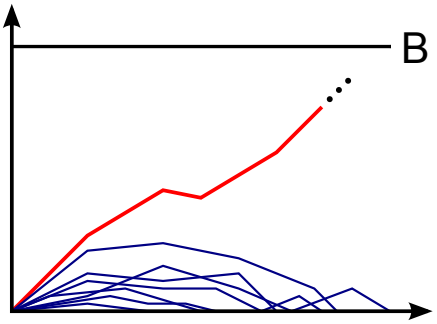

Abbildung 1: Simulation

Genau dies tut Importance Splitting. Wenn 0 der Startzustand ist und B das seltene Ereignis, dann führt man im Intervall $[0, B]$ zusätzliche Stufen $L_0, \ldots, L_n$ ein, wobei $L_0 = 0$ und $L_n = B$. Zusätzlich gilt, dass $L_n \subset L_{n-1} \subset \cdots \subset L_1$. Eine Stufe L_k bei einer Queue könnte z.B. sein, dass die Queue mindesten halb voll ist. Demnach ist eine höhere Stufe, z.B. Queue zu 90 % voll, eine Teilmenge der Zustände aus L_k.

Zu Bestimmen ist

$$\gamma = P(\tau_B < \tau_0),$$

wobei τ_0 die Zeit ist, bis ein Simulationspfad wieder den Startzustand L_0 erreicht, und τ_B die Zeit, bis der Simulationspfad den Zustand B erreicht. Man bestimmt nun die Wahrscheinlichkeiten $p_1 \ldots p_n$, dass ein Simulationspfad von einer Stufe p_{i-1} aus p_i erreicht, bevor der Pfad wieder den Startzustand L_0 erreicht. Dazu simuliert man das System von L_0 aus n_1 mal und speichert die Zustände aller Pfade , die L_1 erreichen, bevor sie L_0 wieder erreichen. Von den gespeicherten Zuständen (Abb. 2, rote Kreise), simuliert man nun eine bestimmte Anzahl an Pfaden weiter um Werte für die Wahrscheinlichkeiten $p_1 \ldots p_n$ zu erhalten.

Sei dazu $h : X \mapsto \mathbb{R}$ eine Funktion, die jedem Zustand ein *importance value* zuordnet. Diese Funktion ist die *importance function* [11] . Sie hat die Eigenschaft, dass für den

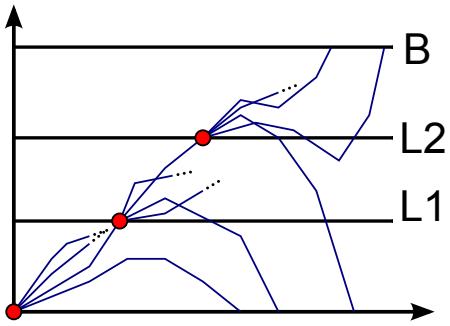

Abbildung 2: Splitting

Startzustand 0 gilt: $h(0) = 0$. Für das seltene Ereignis B gilt $h(B) = l$ und für die Stufen dazwischen: $h(L_i) = l_i$ für $i \in \{0, \ldots, n\}$ mit $0 = l_0 < l_1 < \cdots < l_n = l$. Sei weiterhin $D_k = \{T_k < \tau_A\}$ das Ereignis, dass ein Simulationspfad Level l_k erreicht, bevor er wieder den Startzustand erreicht. T_k ist hier $T_k = \inf\{j > 0 : h(X_j \geq l_k\}$, die Zeit bis das Level k das erste Mal erreicht wird. Seien nun $p_1 = P(D_1)$ die Wahrscheinlichkeit, dass ein Simulationspfad D_1 erreicht, und $p_k = P(D_k|D_{k-1})$ die Wahrscheinlichkeit, dass wenn D_{k-1} erreicht wird auch D_k erreicht wird. Damit lässt sich γ wie folgt bestimmen:

$$\gamma = P(D_n) = p_1 p_2 \ldots p_n = \prod_{k=1}^{n} p_k$$

Man bestimmt jetzt alle p_k separat. Vom Startzustand L_0 aus werden N_0 unabhängige Simulationspfade gestartet, bis jeder der Pfade entweder den Startzustand oder L_1 erreicht. Dann lässt sich p_1 annähern durch

$$\hat{p}_1 := \frac{R_1}{N_0},$$

wobei R_1 die Anzahl der Pfade ist, die L_1 erreichen. Wie man leicht sieht, ist dieser Schätzwert für p_1 unverfälscht. Ähnlich verfährt man für die Bestimmung der restlichen $\hat{p}_k$. Von den bei L_{k-1} gespeicherten Simulationspfaden werden N_{k-1} neue Simulationspfade erzeugt und diese bis zu L_0 oder L_k weiter simuliert. Sei wiederum R_k die Anzahl der Pfade, die L_k erreicht haben, dann gilt für $\hat{p}_k$:

$$\hat{p}_k := \frac{R_k}{N_{k-1}}$$

Nun gibt es im Großen und Ganzen 2 Möglichkeiten, wie die Anzahl und auf welche Weise die Simulationspfade für das nächste Level ausgewählt werden.

3.2.1 Fixed Splitting

Bei der *Fixed Splitting* (FS) Methode wird jeder Pfad der Level L_k erreicht r_k mal aufgesplittet, wobei r_k eine Konstante ist. Das heißt, dass $N_k = r_k \cdot R_k$. Das Problem ist

nun, r_k so zu wählen, dass die Simulationspfade weder aussterben noch in ihrer Anzahl explodieren. Ist nämlich r_k zu groß gewählt nimmt die Anzahl der Simulationspfade immer weiter zu und der Rechenaufwand wird enorm. Andererseits, wenn r_k zu klein gewählt ist, sterben die Simulationspfade aus und die Varianz erhöht sich stark. Für bestmögliche Effizienz bei einem bestimmten Gesamtaufwand (*effort*) r sollten die Anzahl der Level

$$n \approx \frac{-\log(\gamma)}{2},$$

die Wahrscheinlichkeit des Erreichens des k-ten Levels

$$p_k \approx e^{-2}$$

und der Erwartungswert für die Anzahl der Pfade auf der Stufe k

$$N_k \approx \frac{r}{n}$$

gewählt werden [5].

3.2.2 Fixed Effort

Bei *Fixed Effort* (FE) wird ein Gesamtaufwand (*effort*) r festgelegt. Dann soll $r = r_0 + \cdots + r_{n-1}$ gelten. Man setzt weiterhin $N_k = r_k$. Die Anzahl der Simulationspfade pro Level ist also konstant. Ein Aussterben oder Explodieren der Simulationspfade wird damit verhindert. Für beste Varianzreduktion sollte

$$r_0 = r_1 = \cdots = r_{n-1}$$

und

$$p_k = \gamma^{1/m} = e^{-2}$$

gewählt werden [5]. Für die Art, auf welche der gespeicherten Zustände für die neuen Simulationen benutzt wird, gibt es 2 Möglichkeiten:

Fixed Assignment: Die gespeicherten Zustände werden so gut wie möglich auf die neuen Simulationspfade verteilt. Sei dazu $d_k = N_k \mod R_k$ und D_k die Menge von d_k zufällig ausgewählten Zuständen aus R_k. Dann wird jeder gespeicherte Zustand X_k für $\lfloor N_k/R_k \rfloor + 1_{\{X_k \in D_k\}}$ neue Simulationspfade benutzt.

Random Assignment: Für jeden neuen Simulationspfad wird zufällig einer der gespeicherten Zustände gewählt. Diese Variante ergibt jedoch immer eine höhere Varianz als Fixed Assignment (siehe [11]).

3.2.3 Vergleich von FE und FS

Beide Methoden Fixed Splitting und Fixed Effort können in etwa die gleiche Varianzreduktion erzielen [5]. Jedoch ist Fixed Effort die robustere Variante. Dafür kann die Fixed Splitting Variante ressourcenschonender implementiert werden. Während Fixed Effort

nur so implementiert werden kann, dass eine komplette Stufe simuliert wird (*breadth-first* Implementierung), müssen sehr viele Zustände gespeichert werden. Fixed Splitting hingegen kann *depth-first* implementiert werden; es wird jeder Simulationspfad erst zu ende simuliert bis der nächste simuliert wird. Dadurch ist weniger Speicher notwendig.

Ein abschließendes Problem taucht bei der Simulation höherer Level auf. Pfade müssen bis zur nächsthöheren Stufe L_k oder bis zum Startzustand L_0 simuliert werden. Damit wird hier der Simulationsaufwand mit zunehmendem Importance Level immer höher, da der Simulationsweg und damit die Simulationszeit für Pfade, die zum Startzustand zurückkehren immer länger wird. Man kann Simulationspfade abbrechen, sobald diese ein Level $L_{k-\beta}$ wieder erreichen, wobei $\beta > 1$. Damit verfälscht man allerdings das Ergebnis und damit p_k. Für kleine β ist die Verfälschung relativ groß, aber der Rechenaufwand kann stark reduziert werden. Ist β zu groß, wird zwar die Verfälschung geringer, allerdings spart man nicht viel Rechenzeit mehr ein. Alternativ gibt es auch Möglichkeiten des unverfälschten Abbruches (*truncation*) (siehe [11]).

4 Beispiel

Es soll als Beispiel eine einfache M/M/1 Warteschlange verwendet werden mit Ankunftsrate λ und Bedienungsrate μ. Man definiert:

$$\rho = \frac{\lambda}{\mu}$$

Die Wahrscheinlichkeit, dass der Zustand N, nämlich die Füllmenge der Warteschlange, einen Zustand $N \geq r$ annimmt ist:

$$p = Pr\{N \geq r\} = \rho^r$$

Sei dies unser seltenes Ereignis, das simuliert werden soll und $\gamma = p$. Es gibt hier also bereits eine geschlossene Formel. Daher würde man in der Praxis dieses System nicht extra simulieren. Meist existiert jedoch einfach keine geschlossene Formel und man ist auf das Simulieren angewiesen. Der Einfachheit halber nehmen wir jedoch dieses System um die vorgestellten Rare-Event Methoden zu veranschaulichen.

Die Varianz für den Schätzer zur naiven Simulation aus Gleichung (1) ist $\sigma_p^2(1_{\{Z_i \in R\}}) = \gamma(1-\gamma)$. Da γ unbekannt ist, nimmt man als Näherung $\hat{\gamma}$:

$$\sigma_p^2(1_{\{Z_i \in R\}}) = \hat{\gamma}(1-\hat{\gamma})$$

Das Konfidenzintervall $(1-\alpha)100\%$ ist hier:

$$\hat{\gamma} \pm z_{\alpha/2}\frac{\sigma_p(1_{\{Z_i \in R\}})}{\sqrt{n}} = \hat{\gamma} \pm z_{\alpha/2}\sqrt{\frac{\hat{\gamma}(1-\hat{\gamma})}{n}}$$

$z_{\sigma/2}$ ist eine Konstante die man der Tabelle der Standardnormalverteilung entnehmen kann. So ergibt sich z aus der Tabelle der Standardnormalverteilung durch $1-\Phi(z) = \alpha/2$.

n ist die Anzahl der Simulationsergebnisse [1]. Die Weite des Konfidenzintervalls beträgt damit:

$$2z_{\alpha/2}\sqrt{\frac{\hat{\gamma}(1-\hat{\gamma})}{n}}$$

Nun möchte man, dass diese Konfidenzintervallweite nur z.B. 5% von γ ist, um sichere Aussagen treffen zu können:

$$2z_{\alpha/2}\sqrt{\frac{\hat{\gamma}(1-\hat{\gamma})}{n}} \geq 0,05 \cdot \gamma$$

Da γ meist unbekannt ist, nimmt man den Schätzer $\hat{\gamma}$. Umgestellt nach n ergibt sich:

$$n \geq \left(\frac{2z_{\alpha/2}\sqrt{\hat{\gamma}(1-\hat{\gamma})}}{0,05 \cdot \hat{\gamma}}\right)^2$$

Man sieht, dass n sehr schnell sehr groß wird, wenn γ gegen 0 geht. Das heißt man benötigt sehr viele Ergebnisse um Aussagen mit einer bestimmten Genauigkeit treffen zu können.

Dies soll nun mit Zahlenbeispielen verdeutlicht werden. Sei $\lambda = 0,5$, $\mu = 1$ und damit $\rho = 0,5$. Weiterhin sei das seltene Ereignis, dass sich mindestens k Kunden in der Warteschlange befinden. Da wir keine Messergebnisse haben, nehmen wir zur Bestimmung von $\hat{\gamma}$ die geschlossene Formel zu γ.

k	$(1-\alpha)100\%$	γ	n
10	95%	$9,7656 \cdot 10^{-4}$	2.508
10	98%		2.982
20	95%	$9,5367 \cdot 10^{-7}$	81.006
20	98%		96.298
50	95%	$8,8818 \cdot 10^{-16}$	263.066.513
50	98%		312.727.028
100	95%	$7,8886 \cdot 10^{-31}$	8.827.060.334.190.963
100	98%		10.493.393.152.380.075

Wie man sieht wird der Simulationsaufwand schon für relativ kleine Belegungen von 10, 20, 50 oder 100 Kunden enorm. Allein der Fall mit 100 Kunden ist reell nich simulierbar, da die benötigten Ergebnisse jeden Zeitrahmen sprengen. Wir wenden jetzt also Importance Sampling an. Die Wahrscheinlichkeitsmaßänderung ist hier eine Änderung der Parameter λ und μ und damit eine Änderung von ρ. Zum Vergleich nehmen wir das seltene Ereignis, dass 50 oder mehr Kunden in der Warteschlange warten und das Ergebnis in einem Konfidenzintervall von 95% liegen soll.

ρ'	γ'	n	$L_{\rho'}$
0,5	$8,8818 \cdot 10^{-16}$	263.066.513	1
0,7	$1,7985 \cdot 10^{-8}$	584.602	$4,9384 \cdot 10^{-8}$
0,9	$5,1538 \cdot 10^{-3}$	1.089	$1,7233 \cdot 10^{-13}$

Man sieht, dass durch eine Änderung der Simulationsparameter eine starke Varianzreduktion erreicht wird. Es werden viel weniger Ergebnisse benötigt, um die gleiche Genauigkeit zu erzielen.

Importance Splitting verfolgt einen anderen Ansatz. Hier kommt es primär nicht auf die Varianzreduktion an, sondern man versucht schneller seltene Ereignisse zu forcieren. Deshalb kann es hier nicht direkt mit naiver Simulation und Importance Sampling verglichen werden.

5 Conclusion

Mit Importance Sampling und Importance Splitting stehen zwei sehr unterschiedliche Ansätze zur Verfügung, um an Informationen über seltene Ereignisse zu kommen. Bei Importance Sampling muss das System genau bekannt sein, damit eine neue, effektivere Wahrscheinlichkeitsverteilung gefunden werden kann, die eine geringere Varianz und eine höhere Wahrscheinlichkeit für das seltene Ereignis aufweist. Bei Importance Splitting muss nur eine Importance function gefunden werden, um den Algorithmus implementieren zu können. Diese Funktion kann auch durchaus mehrdimensional sein bei komplexeren Systemen [11].

Für die Simulation seltener Ereignisse gibt es aber auch schon Programme und Bibliotheken, wie z.B. SAVE, MonteQueue und UltraSAN [3] für Importance Sampling und ASTRO [19] für Importance Splitting [6]. Diese sind zum Teil für die Simulation spezieller Systeme ausgelegt, wie z.B. MonteQueue für Product-Form Multiclass Queueing Netzwerke [16].

Auch wenn die beiden vorgestellten Verfahren, die wichtigsten sind, so gibt es trotzdem noch andere Verfahren für die Rare Event Simulation. Als Beispiel wären da zu nennen: Repeated Acceptance/Rejection und Filtered Conditional Monte Carlo [21].

Aber auch bei den vorgestellten Methoden Importance Sampling und Importance Splitting ist noch längst nicht alles gesagt. Bei Importance Sampling gibt es zum Beispiel noch Methoden, die die Bounded Relative Error Eigenschaft garantieren, wie Balanced Failure Biasing [15] [2]. Weitere Vertiefung geht vor allem in Richtung der zu untersuchenden Systeme. Für zuverlässige Systeme siehe [18] [8] [14] und [7] und für Netzwerkqueues siehe [10] [7] und [4].

Eine Erweiterung zu Importance Splitting ist noch RESTART und RESTART/LRE. Weitere Informationen zu der Importance Function: [13]

Interessant ist noch folgender Artikel, der die Schwierigkeit der Anwendung der Methoden verdeutlicht: [9].

Quellen

[1] Rare event simulation techniques: An introduction and recent advances. In *Handbook in Operations Research and Mangement Sciences*, volume 13, pages 291–350. Elsevier, 2006.

[2] Christos Alexopoulos and Bruce Shultes. The balanced likelihood ratio method for estimating performance measures of highly reliable systems. In *WSC '98: Proceedings of the 30th conference on Winter simulation*, pages 1479–1486, Los Alamitos, CA, USA, 1998. IEEE Computer Society Press.

[3] Joseph A. Couvillion, Roberto Freire, Ron Johnson, W. Douglas Obal, M. Akber Qureshi, Manish Rai, William H. Sanders, and Janet E. Tvedt. Performability modeling with ultraSAN. *IEEE Software*, 8(5):69–80, September 1991.

[4] R. D. Fresnedo. Quick simulation of rare events in networks. In *WSC '89: Proceedings of the 21st conference on Winter simulation*, pages 514–523, New York, NY, USA, 1989. ACM.

[5] Marnix J. J. Garvels and Dirk P. Kroese. A comparison of restart implementations. In *WSC '98: Proceedings of the 30th conference on Winter simulation*, pages 601–608, Los Alamitos, CA, USA, 1998. IEEE Computer Society Press.

[6] Boudewijn R. Haverkort, Raymond Marie, Gerardo Rubino, and Kishor Shridharbhai Trivedi. *Performability Modelling: Techniques an Tools*, chapter 8, pages 163–178. Wiley, 2001.

[7] Philip Heidelberger. Fast simulation of rare events in queueing and reliability models. *ACM Trans. Model. Comput. Simul.*, 5(1):43–85, 1995.

[8] Philip Heidelberger, Victor F. Nicola, and Perwez Shahabuddin. Simultaneous and efficient simulation of highly dependable systems with different underlying distributions. In *WSC '92: Proceedings of the 24th conference on Winter simulation*, pages 458–465, New York, NY, USA, 1992. ACM.

[9] A.C.M. Hopmans and J.P.C. Kleijnen. importance sampling in system simulation: a practical failure? In *Mathematics and Computing in Simulation XXI*, pages 09–220, 1979.

[10] Pierre L'Ecuyer and Yanick Champoux. Importance sampling for large atm-type queueing networks. In *WSC '96: Proceedings of the 28th conference on Winter simulation*, pages 309–316, Washington, DC, USA, 1996. IEEE Computer Society.

[11] Pierre L'Ecuyer, Valérie Demers, and Bruno Tuffin. Splitting for rare-event simulation. In *WSC '06: Proceedings of the 38th conference on Winter simulation*, pages 137–148. Winter Simulation Conference, 2006.

[12] Wolfgang Ludwig-Mayerhofer. Konfidenzintervalle so einfach wie möglich erklärt.

[13] Dirk P. Kroese Marnix J. J. Garvels, Jan-Kees C. W. Van Ommeren. On the importance function in splitting simulation. In *European Transactions on Telecommunications*, volume 13, pages 363 – 371, 2002.

[14] Marvin Nakayama. General conditions for bounded relative error in simulations of highly reliable markovian systems. In *Advances in Applied Probability*, pages 687–727, 1996.

[15] Victor F. Nicola, Perwez Shahabuddin, and Marvin Nakayama. Techniques for the fast simulation of models of highly dependable systems. *IEEE Transactions on Reliability*, 50:246–264, 2001.

[16] K. W. Ross and WANG. Montequeue: A software package for analyzing productform multiclass queueing networks. Technical report, University of Pennsylvania, 1994.

[17] Perwez Shahabuddin. Rare event simulation in stochastic models. In *WSC '95: Proceedings of the 27th conference on Winter simulation*, pages 178–185, Washington, DC, USA, 1995. IEEE Computer Society.

[18] Perwez Shahabuddin, Victor F. Nicola, Philip Heidelberger, Ambuj Goyal, and Peter W. Glynn. Variance reduction in mean time to failure simulations. In *WSC '88: Proceedings of the 20th conference on Winter simulation*, pages 491–499, New York, NY, USA, 1988. ACM.

[19] Manuel Villén-Altamirano and José Villén-Altamirano. Restart: a straightforward method for fast simulation of rare events. In *Winter Simulation Conference*, pages 282–289, 1994.

[20] Gerhard Winkler. Stochastische Prozesse in der statistischen Modellierung. 2000.

[21] Xiaowei Zhang, Jose Blanchet, and Peter W. Glynn. Efficient suboptimal rare-event simulation. In *WSC '07: Proceedings of the 39th conference on Winter simulation*, pages 389–394, Piscataway, NJ, USA, 2007. IEEE Press.