

## Aufgabenblatt 7 <sup>1</sup>

### Vorbereitung

#### Aufgabe 1: Phasen von nicht-terminierenden Simulationen

Was versteht man unter der transienten Phase und dem Steady State einer nicht-terminierenden Simulation?

#### Aufgabe 2: Transienten-Erkennung

Rekapitulieren Sie anhand der Folien, wie man eine Transienten-Erkennung durchführen kann.

Wir bedienen uns hierfür der Funktion

$$f(k) = \frac{\overline{\overline{X}}_k - \overline{\overline{X}}}{\overline{\overline{X}}}$$

wobei

- $m$  : Anzahl der Replikationen
- $n$  : Anzahl der Samples in einer Replikation
- $X_{i,j}$  Wert des  $j$ . Samples der  $i$ . Replikation
- $\overline{X}_j$  Mittelwert des Wertes  $j$  über alle Replikationen
- $\overline{\overline{X}}$  Mittelwert der Mittelwerte aller Replikationen
- $\overline{\overline{X}}_k$  Mittelwert der Mittelwerte aller Replikationen ohne die ersten  $k$  Werte

$$\overline{X}_j = \frac{1}{m} \sum_{i=1}^m X_{i,j}$$

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

$$\overline{\overline{X}}_k = \frac{1}{n-k} \sum_{j=k+1}^n \overline{X}_j$$

Bei direkter Anwendung dieser Formeln müsste man für jeden Wert  $k$  jedesmal die Summation neu ausführen, was relativ ineffizient ist. Einfacher ist es, nicht nur die Samples  $X_{i,j}$  selbst sondern auch die bis dahin aufgelaufene Summe  $S_{i,k}$  der ersten  $k$  Samples in der Simulation aufzuzeichnen. Drücken Sie  $S_{i,k}$  analytisch durch  $X_{i,j}$  aus. Wie hängen  $\overline{\overline{X}}$  und  $\overline{\overline{X}}_k$  von  $S_{i,k}$  ab? Finden Sie auch hierfür die Formeln.

#### Aufgabe 3: Konfidenzintervalle

Bestimmen Sie das 95%-Konfidenzintervall für den Mittelwert der Werte 12, 15, 9, 17, 13, 14, 10, 11, 16, 18 mit dem Verfahren, das in Selbststudienmaterialien beschrieben ist. Warum dürfte man es hier eigentlich nicht anwenden?

---

<sup>1</sup>Stand: 12. Oktober 2023

## Praktischer Teil

Denken Sie bitte –wie immer bei den praktischen Aufgaben– an aussagekräftige Debug-Ausgaben. Die Sourcen für diese Aufgabe befinden sich im Verzeichnis `protsim07`.

Der Queue-Implementierung aus der letzten Aufgabe fehlen noch einige Kleinigkeiten, um damit Simulationsstudien, die wissenschaftlichen Anforderungen genügen würden, durchzuführen.

### Aufgabe 4: Implementierung der Transienten-Entfernung

Um die Erkennung und spätere Entfernung der Transienten zu unterstützen, muss Ihr Programm wie folgt erweitert werden:

- Übertragen Sie zunächst Ihren Code für die Queue-Implementierung an die entsprechend gekennzeichneten Stellen der Vorgabe.
- Der `PSOutVector waitingTimeSumVector` soll die Summe der bis dahin aufgelaufenen Wartezeiten aufnehmen. Schreiben Sie den nötigen Code um die entsprechenden Daten zu erfassen. Bitte beachten Sie, dass der Vektor ein Tupel aus Nummer des Samples (beginnend bei 1) und Summe erwartet: `void record(double a, double b)`.
- Für die Transienten-Entfernung müssen die statistischen Daten aus der Anfangsphase der Simulation ignoriert werden. Ergänzen Sie die Funktion `resetStatistics` so, dass Statistiken, Histogramm und Zähler zurückgesetzt werden. Denken Sie auch daran, `lastUtilChange` und `lastQueueChange` auf passende Werte zu setzen.
- Der Parameter `transientRemoval`, der in die gleichnamige Variable übertragen wird, enthält die Anzahl der Wartezeiten, die am Anfang der Simulation verworfen werden sollen. Ergänzen Sie die Funktion `detectTransient()` so, dass anhand der Statistik `waitingTimeStat` erkannt wird, wann dies der Fall ist. In diesem Fall sollte
  - die Tatsache, dass die Transiente vorbei ist, festgehalten werden,
  - die Statistiken etc. zurückgesetzt werden,
  - die drei Out-Vektoren aktiviert werden (`enable()`).

Rufen Sie die Funktion `detect()`, welche auch `detectTransient()` aufruft, nach jeder Änderung von `waitingTimeStat` auf. Achten Sie darauf, dass alle damit zusammenhängenden Änderungen anderer Statistiken **vor** dem Aufruf von `detect()` stattfinden sollten.

### Aufgabe 5: Transienten-Erkennung

Um das Datenaufkommen zu reduzieren sind in diesem Termin alle Out-Vektoren außer die in der `InterfaceQueue 1` des Knotens 2 deaktiviert.

Für die Erkennung der Transienten ist der Parameter `transientRemoval` zunächst auf 0 gesetzt, so dass die komplette Simulation aufgezeichnet wird.

Indem Sie `./runCmd.sh -c RunX` ausführen, wird die Kommandozeilenversion gestartet. Das hat den Vorteil, dass es schneller läuft und weniger Speicher verbraucht.

Zur Erkennung der Transienten gehen Sie bitte wie folgt vor:

1. Führen Sie Run 1 aus. Bei der Kommandozeilenversion geschieht dies durch den Parameter `-c Run1`. Dies erzeugt eine Datei `results/statistics01.vec.psOV`.
2. Im Unterverzeichnis `eval` finden Sie das Python-Skript `ps_eval.py`, welches die Transienten-Erkennung wie in Aufgabe 2 skizziert durchführt. Das Skript erwartet die zu behandelnden Replikationen (Runs), den Basisnamen der Vektor-Dateien und den Namen der zu erzeugenden Datei. Wechseln Sie in das entsprechende Unterverzeichnis und rufen Sie das Skript mit `./ps_eval.py -h` auf um Informationen zu den Parametern zu erhalten.

Zum Auswerten des Run 1 reicht

```
./ps_eval.py -r 1 ../results/statistics transient.vec
```

3. Rufen Sie nun `gnuplot` auf.
4. Plotten Sie die erzeugten Daten mit dem Kommando  
`gnuplot> plot "transient.vec" with lines`
5. Sie können (und sollten) den geplotteten Bereich einschränken. Wenn Sie zum Beispiel nur die x-Werte zwischen 0 und 500 plotten wollen, können Sie folgendes Kommando verwenden:  
`gnuplot> plot [0:500] "transient.vec" with lines`
6. Lesen Sie das Ende der transienten Phase ab und setzen Sie den Parameter `transientRemoval` in `omnetpp.ini` auf den entsprechenden Wert. Runden Sie dabei großzügig auf.

#### **Aufgabe 6:** Simulation mit mehreren Replikationen

Führen Sie nun die Runs 1-31 (`./runAll.sh`) aus. Sie unterscheiden sich nur durch die Seeds für den Zufallszahlengenerator. Das Evaluation-Skript berechnet auch den Mittelwert und die Varianz aus den Mittelwerten der Durchläufe. Zu diesem Zweck können Sie es ohne Angabe einer Ausgabedatei aufrufen. Berechnen Sie das 99.9%-Konfidenzintervall für die mittlere Wartezeit. Der entsprechende Tabellenwert für die Normalverteilung beträgt in diesem Fall  $Z_{\alpha/2} = 3.29$ .