

# Modellgestützte Analyse und Optimierung

## Übungsblatt 2

**Ausgabe:** 11. April, **Abgabe:** 18. April, 12 Uhr

### Aufgabe 2.1 (4 Punkte)

Betrachten sie das Warteschlangensystem wie in Aufgabe 1.2 beschrieben. Nehmen Sie an, dass die Maschine zum Zeitpunkt  $t = 3$  für 3 Zeiteinheiten wegen Wartungsarbeiten ausfällt. Danach bleibt sie für den Rest der Zeit (bis  $t = 20$ ) funktionstüchtig. Während der Ausfallzeit werden die Bauteile nicht weiter bearbeitet. Ein Bauteil, welches sich zum Zeitpunkt des Ausfalls gerade in Bearbeitung befindet, wird nach der Ausfallzeit (für den Rest seiner Bearbeitungszeit) weiter bearbeitet. Der Ankunftsstrom neuer Bauteile wird durch den Ausfall der Maschine nicht beeinträchtigt.

- a) Führen Sie Aufgabenteile a) bis c) aus Aufgabe 1.2 für dieses Warteschlangensystem durch. Überlegen Sie, welche zusätzlichen Ereignisse betrachtet werden müssen. Betrachten sie die Maschine während der Ausfallzeit als ausgelastet, also  $B(t) = 1$ .
- b) Bestimmen Sie die Leistungsgröße „Auslastung der Maschine“ neu, indem Sie nicht nur zwei (ausgelastet/nicht ausgelastet) sondern die folgenden drei Zustände der Maschine unterscheiden:
  - i) Maschine ist nicht ausgefallen und bearbeitet gerade ein Bauteil
  - ii) Maschine ist nicht ausgefallen und bearbeitet gerade kein Bauteil
  - iii) Maschine ist ausgefallen (mit oder ohne aktueller Bearbeitung eines Bauteils)

### Aufgabe 2.2 (8 Punkte)

An einem Supermarkt treffen „immer wieder“ Kunden ein. Jeder Kunde nimmt sich zunächst einen Einkaufskorb, sucht anschließend eine „gewisse Anzahl“ von Waren aus, was „eine gewisse Zeit“ dauert, wendet sich dann den Kassen zu (wo er unter Umständen auf Bedienung warten muss) und wird schließlich bedient, was wieder „eine gewisse Zeit“ in Anspruch nimmt. Genauere Spezifikation:

- Die Ankunftszeiten der verschiedenen Kunden sind festgelegt durch Zwischenankunftszeiten. Die Zwischenankunftszeiten sind durch Beobachtungen am realen Supermarkt bestimmt worden. Für den  $i$ -ten Kunden steht dieser Wert in der Variablen  $z_i$ .
- Die Zahl der Einkaufskörbe ist begrenzt auf  $M$ . Findet ein eintreffender Kunde keinen freien Korb, dann verlässt er den Supermarkt direkt wieder.

- Die Zahl der Waren die ein Kunde aussucht sei „schwankend“. Diese ist ebenfalls durch Beobachtungen am realen Supermarkt bestimmt worden und in der Variablen  $k_i$  gespeichert.
- Abhängig von der Anzahl  $k_i$  auszusuchender Waren benötigt ein Kunde die Zeit  $k \cdot T_s$  für die Warenauswahl und die Zeit  $k \cdot T_p$  für die Bedienung an der Kasse.
- Die Zahl der Kassen ist begrenzt auf  $N$ . Findet ein zahlungswilliger Kunde keine freie Kasse, dann wartet er bis eine Kasse frei wird.
- Fehlen noch Angaben? Treffen sie in diesem Fall „Ihre“ Annahmen und notieren diese tabellarisch.

Entwerfen Sie für das oben beschriebene System ein ereignisorientiertes Simulationsmodell wie in der Abbildung im Skript auf den Seiten 33 und 34. Machen Sie Gebrauch von Pseudo-Code (in Anlehnung an C/C++ oder Java) und umgangssprachlichen Beschreibungen wie Sie es für sinnvoll halten. Beschreiben Sie tabellarisch, welche Zustandsvariablen benötigt werden, welche Attribute zu einem Kunden gehören, welche Ereignisse auftreten können und wie die Ereignisroutinen aussehen. Zur Vereinfachung nutzen Sie bitte bei der Verwendung von Warteschlangen folgende Funktionen: Die Funktion *ws\_einreihen(schlange, element)* fügt ein Element am Ende in eine Warteschlange ein, *ws\_erster(schlange)* liest das erste Element einer Warteschlange aus, *ws\_entfernen(schlange)* entfernt das erste Element einer Warteschlange, und *ws\_leer(queue)* testet, ob eine Warteschlange leer ist.