

Fertigungssystem

Dickbauer Y., Moser P., Perner M.

PS Computergestützte Modellierung, WS 2016/17

January 6, 2017

Outline

- 1 Aufgabenstellung
- 2 Interpretation und Vorgehensweise
- 3 Flow Chart
- 4 Programmcode
 - Main Funktion
 - Verwendete Funktionen
- 5 Beispiel

Aufgabenstellung

In einem Fertigungssystem werden Aufträge auf einer Maschine bearbeitet. Es gibt zwei Typen von Produkten: Typ 1 (Typ 2) benötigt auf der Maschine eine Bearbeitungszeit, die stetig gleichverteilt zwischen 2 und 6 min (1.5 und 4.5 min) liegt. Die Wahrscheinlichkeit, dass ein Produkt vom Typ 1 ist, ist 0.4. Die Produkte kommen exponentialverteilt mit Erwartungswert von 4 in das System. Die Kapazität der Warteschlange ist mit 5 Stück begrenzt; Produkte, die in das System kommen, während die Warteschlangenkapazität ausgelastet ist, werden aus dem System eliminiert.

Aufgabenstellung

Anschließend an die Bearbeitungsphase kommt eine Inspektion. Die Zeit, die man braucht, um diese durchzuführen, ist für Produkttyp 1 (Produkttyp 2) gleichverteilt zwischen 3 und 5 min (1 und 3 min). Es wird überprüft, ob ein Produkt defekt ist oder nicht. Die Wahrscheinlichkeit, dass ein Produkt defekt ist und somit aussortiert wird, ist 0.1.

- (a) Zählen Sie die Stücke, die entfernt werden, weil die Kapazität der Warteschlange vor der Maschine zu gering war, und die Anzahl der defekten Stücke.
- (b) Bestimmen Sie die durchschnittliche Länge der Warteschlange vor der Inspektionsstation und die Auslastung der Maschine und der Inspektionsstation.
- (c) Wie lange brauchen die Produkte durchschnittlich, um durch das System geschleust zu werden?

Aufgabenstellung

Beginnen Sie die Simulation mit einer Aufwärmphase von 8h. Danach sollen alle statistischen Werte gelöscht werden. Die Zeit der tatsächlichen Simulation ist 800h. Stellen Sie im Rahmen der Präsentation den Ablauf des Programmes anhand von selbstgewählten Zufallszahlen vor.

- Eingabe: -
- Output: Verlauf von Produktion (Startzeit, Bearbeitungszeit, Endzeit je Produkt), Warteschlangenlänge bei Bearbeitung und Inspektion, sowie die oben angeführten Kennzahlen.

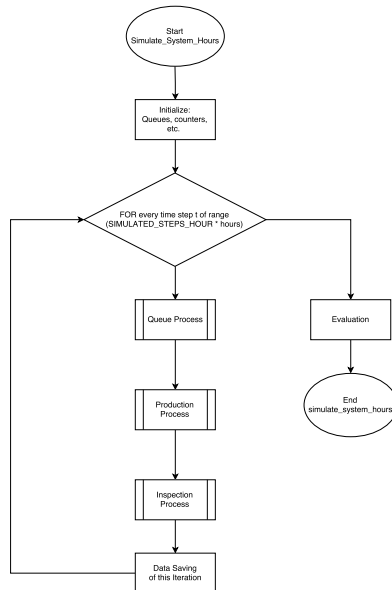
Problem Fertigungssystem

- 2 verschieden Produkttypen mit jeweiligen Processing Time Intervallen
- 1 Maschine
- 1 Inspektionsstation
- 2 Queues
 - vor Maschine mit Kapazität 5
 - vor Inspektionsstation mit Kapazität INF

Simulation

- Simulation in Sekunden Schritten
- Generation von exponential-verteilten ZZ für den Produktrelease in das Fertigungssystem
- Produkttypen treten mit bestimmter WK auf
- Generation von ZZ im gegebenen Produktionsprozessintervall
- Fehler treten mit gewisser WK auf

Flow Chart



Main Funktion - Programmeinstieg

```
1 def main():
2     print('Starting the warming phase (8 hours):\n')
3     simulate_system(8)
4
5     for i in range(5): print()
6     print('Starting the real simulation (800h):')
7     simulate_system(800)
```



- Funktion `simulate_system()`: siehe .py file bzw. Flow Chart

Funktion `random_number_from_interval(..)`

- Diese Funktion verlangt zwei Eingabeparameter *lower* und *upper*
- Gibt eine (pseudo)Zufallszahl (*float*) im Intervall [*lower*, *upper*) zurück
- `random.random()` ist eine Funktion der Python Standardbibliothek, welche eine Zufallszahl (*float*) im Intervall [*lower*, *upper*) zurück gibt
- Mersenne Twister Methode wird als Generator der ZZ verwendet^{1 2}

```
1 def random_number_from_interval(lower, upper):  
2     val = random.random()  
3     return lower + (upper - lower) * val
```



¹<https://docs.python.org/3.5/library/random.html>

²https://en.wikipedia.org/wiki/Mersenne_Twister

Funktion loaded_random_choice(..)

- Diese Funktion verlangt eine WSKL Liste als Eingabeparameter
- Gibt einen Index zurück, welcher 0 bis $|probability_list| - 1$ sein kann.
- Diese Indizes haben eine gewichtete WSKL, welche jeweils an der Position in der Eingabeliste steht
- Beispiel $probability_list := [0.9, 0.1]$ \Rightarrow mit $p=90\%$ wird 0 zurückgegeben, $p=10\%$ für 1

```
1 def loaded_random_choice(probability_list):
2     n = len(probability_list)
3     random_number = random.random()
4     cum_p = 0
5     for i in range(n):
6         cum_p += probability_list[i]
7         if cum_p > random_number:
8             return i
9     return None
```



Funktion random_exp(n, p)

- Erzeugt exponential verteilte Zufallszahl
- Gemäß Inversionsmethode:
- $ZZ = -\frac{1}{\lambda} \ln(u)$ mit $u=R(0,1)$

```
1 def random_exp(lambd):  
2     rand = random.random()  
3     return -(1/lambd) * log(rand)
```

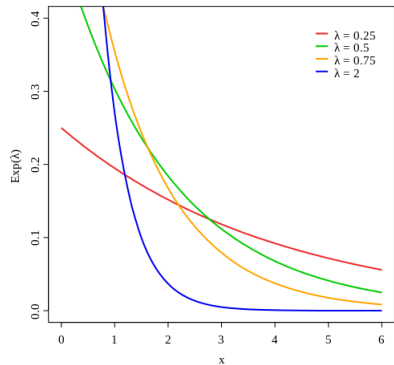


Figure 1: Wahrscheinlichkeitsverteilung der Exponentialverteilung (Wikipedia)

Produktrelease: Funktion

`generate_time_until_next_product(..)`

- Exponentialverteilung
- $\lambda = 4$ (Minuten)
- Beispiel:
 - $ZZ1 = 0.83$
 - $ZZ = -\frac{1}{4} \ln(0.83) \Rightarrow 0.0202 * 60 = 1.212 \text{sek}$

Funktionen zur Produkterzeugung und Processing:

- Funktion: `generate_product()`:
 - erzeugt mittels einer `loaded_random_choice` Funktion entweder Produkttyp 1 oder 2
- Funktion: `generate_processing_time()`:
 - erzeugt gegeben des Produkttyps aus einer `random_number_from_interval` Funktion eine ZZ (processing time in Minuten)

Funktionen zur Inspektion:

- Funktion: `is_product_defect()`:
 - sagt aus, ob Produkt einen Defekt hat (mit Hilfe von `loaded_random_choice`)
- Funktion: `generate_inspection_time()`:
 - mit Hilfe von `random_number_from_interval`

Berechnung eines Durchlaufs

1) QUEUE PART
PRODUCT A
 time_until_next_product == 0
 1.1) generate_product Produkttypen {0,4; 1}
 ZZ = 0,14
 0,14 fällt in Index 0 -> Produkttyp 1
 Kapazität queue ok!
 1.2) generate_time_until_next_product
 ZZ = 0,15
 ZZ = -1/4 * ln(0,15) * 60 -> 28 Sekunden

PRODUCT B
 time_until_next_product == 0

1.1) generate_product
 ZZ = 0,52
 0,52 fällt in Index 1 -> Produkttyp 2
 Kapazität queue ok!
 1.2) generate_time_until_next_product
 ZZ = 0,65
 ZZ = -0,25 * ln(0,65) * 60 -> 6 Sekunden

2) PRODUCTION PART

PRODUCT A
 Kein Produkt in der Maschine
 Produkt geht von queue in Maschine
 2.1) generate_processing_time
 ZZ = 2,7 * 60 = 160 Sekunden
 Zeit verstreicht 160 steps lang
 Finished Product -> geht in inspection_queue

PRODUCT B
 zum Zeitpunkt t = 28 -> Maschine besetzt
 wartet in queue noch = 160 - 28 steps lange
 t = 161 -> Maschine frei
 2.1) generate_processing_time
 ZZ = 3,2 * 60 = 192 Sekunden
 Zeit verstreicht 192 steps lang
 Finished Product -> geht in inspection_queue

3) INSPECTION PART

PRODUCT A
 Keine Inspektion im Gange
 Produkt geht von inspection_queue in inspection
 3.1) generate_inspection_time
 ZZ = 4,2 * 60 = 252 Sekunden
 Zeit verstreicht 252 Steps
 3.2) generate_defects {0,1; 1}
 ZZ = 0,25
 0,25 gehört zu Index 1 -> kein Defect

PRODUCT B
 Inspektion noch 413-353 = 60 Steps im Gange
 Es verstreichen 60 Steps
 3.1) generate_inspection_time
 ZZ = 1,7 * 60 = 102 Sekunden
 Zeit verstreicht 102 Steps
 3.2) generate_defects
 ZZ = 0,89
 0,89 gehört zu Index 1 -> kein Defect

für die Generation von ZZ [0,1] wurden die Nachkommastellen von pi benutzt

| Markierungen | Time Counter A | Time Counter B |
|--|----------------|----------------|
| nr_generated_products == 1 queue == 1 | | 1 |
| time_until_next_product == 28 | | |
| 28* (time_until_next_product - 1) nr_generated_products == 2 time_until_next_product == 0 | | 1 |
| queue == 1 | | 28 |
| 6* (time_until_next_product - 1) time_until_next_product == 0 | | |
| queue == 1 | | 1 |
| remaining_processing_time == 160 remaining_processing_time == 0 nr_finished_products == 1 | | 161 |
| queue == 1 | | 161 |
| remaining_processing_time == 192 remaining_processing_time == 0 nr_finished_products == 2 queue_inspection == 1 | | 353 |
| queue_inspection == 1 | | 161 |
| remaining_inspection_time == 252 remaining_inspection_time == 0 | | 413 |
| nr_finished_system == 1 | | |
| queue_inspection == 1 remaining_inspection_time == 60 remaining_inspection_time == 0 | | 413 |
| queue_inspection == 1 remaining_inspection_time == 102 remaining_inspection_time == 0 | | 515 |
| nr_finished_system == 2 | | |

Durchlaufzeit Produkt A von Zeitpunkt des Eintritts in die Queue

6,9 Minuten

Durchlaufzeit Produkt B von Zeitpunkt des Eintritts in die Queue

8,1 Minuten

Durchlaufzeit Produkte

1 Typ 1

1 Typ 2