

Fertigungsstraße

Dickbauer Y., Moser P., Perner M.

PS Computergestützte Modellierung, WS 2016/17

January 6, 2017

Outline

- 1 Aufgabenstellung
- 2 Flow Chart
 - Verwendete Funktionen
- 3 Grafische Darstellung

Aufgabenstellung

Eine Fertigungsstraße besteht aus 3 Maschinen. Die Straße erhält die zu bearbeitenden Teile aus einem Rohlager (mit unendlichem Vorrat); die bearbeitenden Teile werden in ein nachgeschaltetes Fertigteilager (mit ebenfalls unendlicher Kapazität) geliefert.

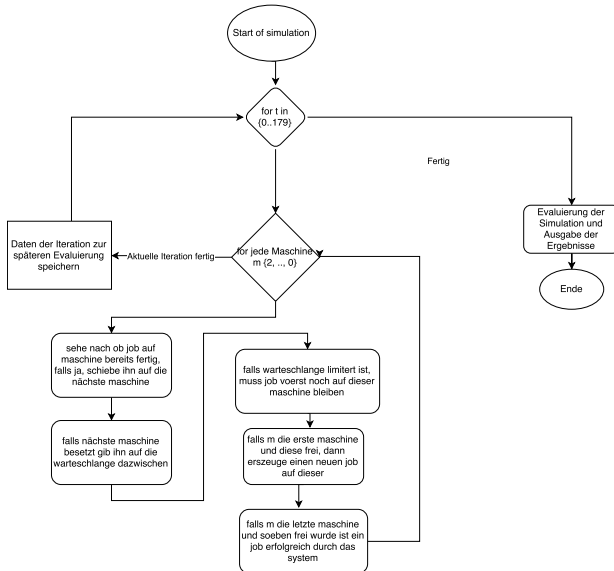
Die normale Bearbeitungszeit eines Teiles beträgt pro Maschine eine Minute; bei 15 Teilen tritt jedoch eine Störung von vier Minuten auf, so dass die gesamte Bearbeitungszeit eines Teiles dann 5 Minuten pro Maschine beträgt.

Aufgabenstellung

Untersuchen Sie durch Simulation über 180 Minuten, ob es zweckmässig ist, die drei Maschinen zu entkoppeln und zwischen den Maschinen ein Pufferlager einzurichten. Vergleichen Sie dazu beispielsweise den Ausstoß (gefertigte Stückzahl), die Stillstandszeiten und den Auslastungsgrad der Maschine sowie die durchschnittliche Bearbeitungszeit eines Teiles auf der Fertigungsstraße. Stellen Sie im Rahmen der Präsentation den Ablauf des Programmes anhand von selbstgewählten Zufallszahlen beziehungsweise mit einem Gantt-Diagramm die Abhängigkeiten der Maschinen vor.

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

Flow Chart - grobe Darstellung der Implementierung



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
```

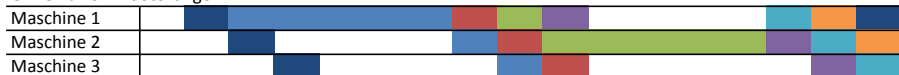


Gantt Diagramm - Zur Veranschaulichung des Unterschieds mit und ohne Zwischenpuffer

Ohne Puffer, Ohne Störung



Ohne Puffer mit Störungen



Mit Puffer und Störungen

