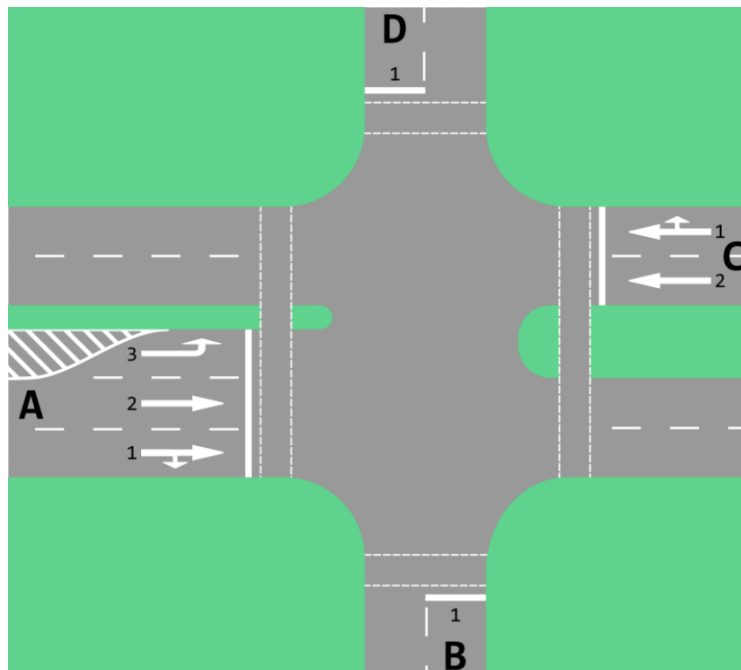


# Intelligente Systeme

## Simulation einer Straßenverkehrs-Kreuzung

### Praktikum 1



Paras Mani Adhikari  
Naresh Mehta  
Bibek Upreti

## Einleitung:

In dieser Projektreport geht es um die Erstellung eines Java-Softwareprogramms, das Simulationen von eine Verkehrskreuzung durchführt. Die Kreuzung verfügt insgesamt über 7 Fahrspuren. Für die Frage gelten mehrere komplizierte Regeln, die die Simulation einhalten muss, um Folgendes sicherzustellen:

1. Es kommt nicht zu Kollisionen.
2. Auf der Fahrspur dürfen nicht mehr Autos fahren, als sie groß ist.
3. Autos verlassen am Ende die Spur, die ihrer Richtung entspricht.

Die Simulation selbst besteht aus drei Phasen:

1. Neue Fahrzeuge hinzufügen.
2. Aktualisierung der Lichter in jeder Spur.
3. Entfernen von Fahrzeugen auf Fahrspuren mit grünem Licht.

Durch die Implementierung der Software werden Messungen der durchschnittlichen Verweildauer von Fahrzeugen, ihrer Anzahl auf den einzelnen Fahrspuren und mögliche Optimierungen anhand von Änderungen an der Kreuzung oder den Ampelschaltzeiten durchgeführt werden. In dieser Einführung werden die wichtigsten Aspekte der Aufgabe dargelegt, um einen Überblick über den Umfang und die Ziele der Softwareentwicklung zu geben.

## Architektur der Software

1. Fahrzeug
  - a. Für jedes hinzugefügte Fahrzeug wird die Eingabezeit durch den Konstruktor gespeichert.
  - b. Für jedes Fahrzeug, das die Simulation bei grüner Ampel verlässt, wird seine Austrittszeit gespeichert und dann wird das Fahrzeug in der exitedCars()-Liste der jeweiligen Fahrspur gespeichert.
2. Spur
  - a. Beim Erstellen einer neuen Spur wird die Wahrscheinlichkeit gespeichert, sodass nur dann ein neues Fahrzeug zur Spur hinzugefügt wird, wenn der Wahrscheinlichkeitswert erreicht ist.
  - b. Jede Spur verfügt über eine Linkedlist aktueller und verlassener Fahrzeuge: Wir implementieren Linkedlist, weil es einfach ist, Elemente aus dem Kopf und auch aus dem angegebenen Index zu entfernen. Der Nachteil ist es braucht mehr Speicher

- c. Aktuelle Spur werden aktualisiert, indem jede Sekunde neue Fahrzeuge hinzugefügt mit bestimmte Wahrscheinlichkeit und Fahrzeuge gemäß den vorgegebenen Wahrscheinlichkeit in Grünzeit entfernt werden
- d. In Spur A und Spur C sind folgende Regeln implementiert:
  - i. Spur 1 (Rechtsabbieger) Fahrzeuge werden immer auf Spur 1 platziert
  - ii. Zuerst wird Spur 2 (Geradeausfahrer) überprüft, um zu sehen, welche von Spur 1 und Spur 2 kürzer sind. Die kürzere Spur erhält das neue Fahrzeug.
  - iii. Wenn ein Fahrzeug auf Spur A3 generiert wird, wird geprüft, ob A2 mehr als 2 Fahrzeuge hat. Wenn ja, wird das Fahrzeug von A3 zu A2 hinzugefügt. Hier im Bild ist "carfromA3-car4" von der Spur A3.

```

A2
| car9 | car10 | car11 | car12 | carfromA3-car4 | car13
A3
| car2 | car3

```

- iv. Ein Auto aus Lane entfernen:  
 Wenn sich eine Spur in Grünzeit befindet und die Wahrscheinlichkeit von 0,5 erfüllt, wird sie entfernt. Aber im Fall der Spur A2. Wir prüfen an Position 4, ob ein Auto auf die Spur A3 wartet. Wenn dies der Fall ist, wird dieses Auto dort warten und ein Auto von dieser Fahrspur entfernt und ein Platz vor dem wartenden Auto frei gemacht. Und in der nächsten Sekunde wechselt dieses wartende Auto die Spur auf die A3.

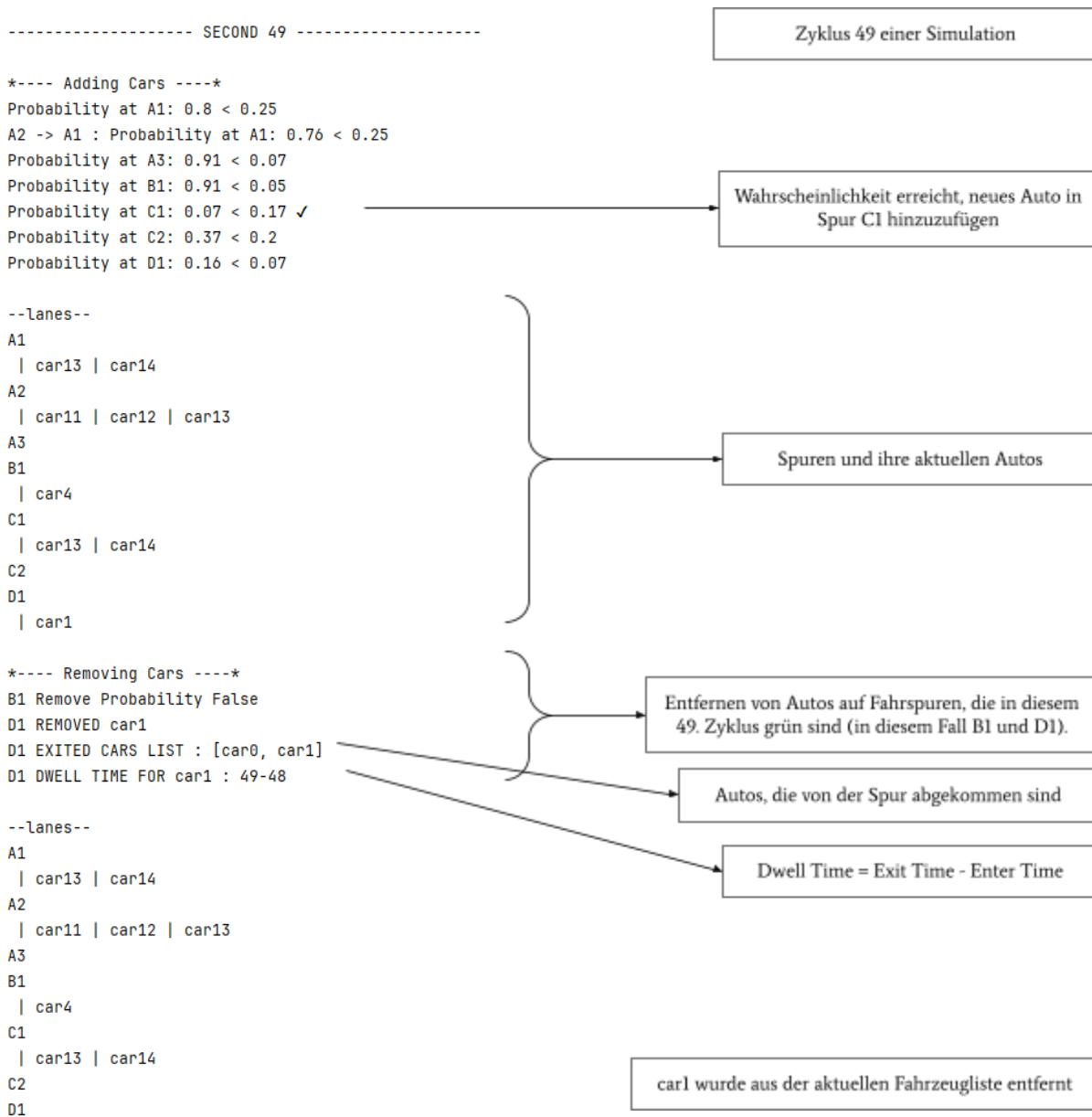
### 3. Simulation

- a. Jede Simulation hat 120 Zyklen, wir haben 30 Simulationen durchgeführt.
- b. Bei jeder Simulation werden die durchschnittlichen Verweilzeiten für jede Fahrspur ermittelt.
- c. Bei jeder Simulation wird die Gesamtzahl der am Ende auf jeder Spur verbliebenen Fahrzeuge ermittelt.

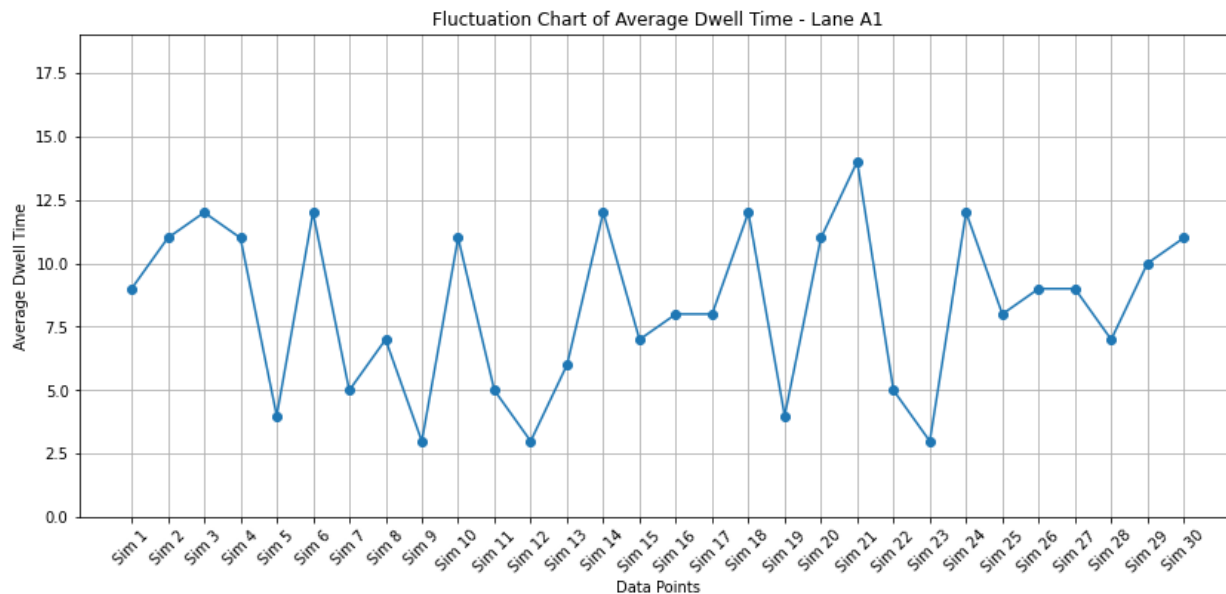
### 4. Cycle

- a. One cycle (1 second) will contain adding 1 car to all the Lane with given probability to corresponding Lane

# Beispiel für einen Zyklus

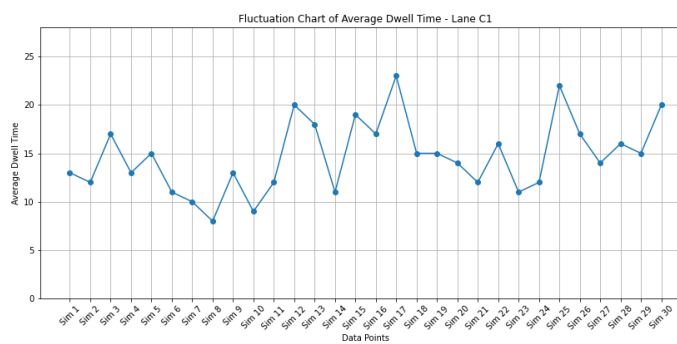
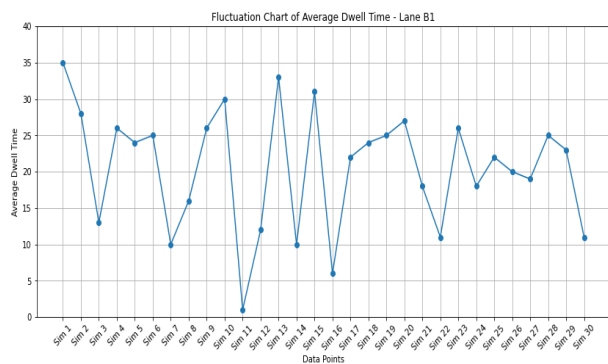
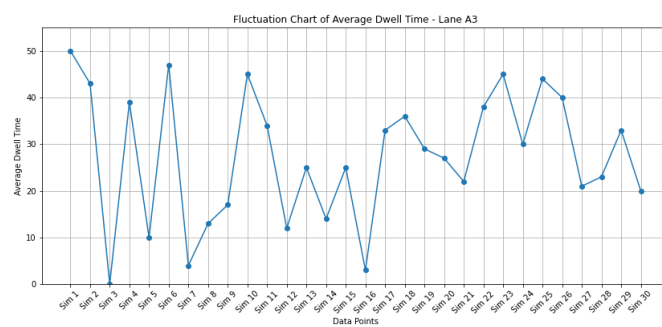
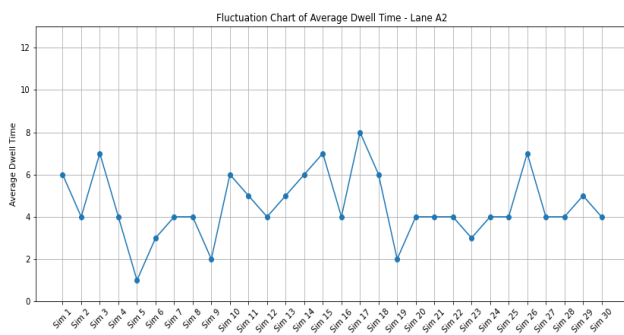


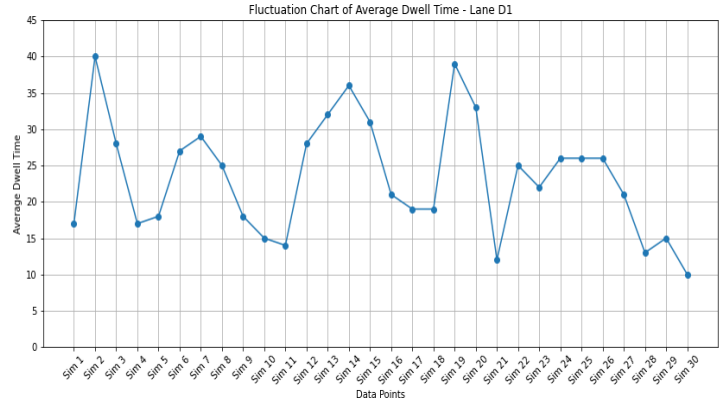
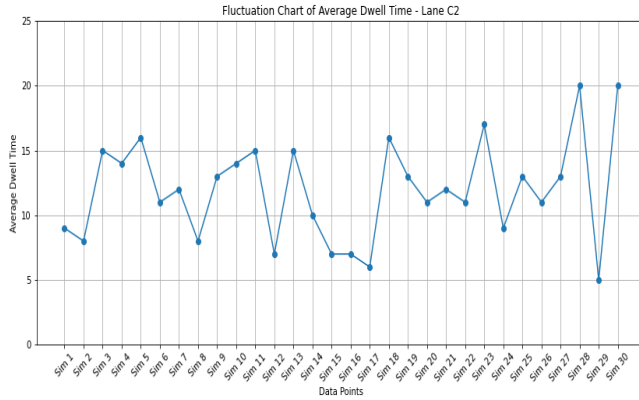
## Ergebnisse - Aufgabe 1a



Fahrtrichtungen	A1	A2	A3	B1	C1	C2	D1
Mittl.Verweildauer	8.3	4.5	27.4	20.57	14.67	11.93	23.4

Fig: Table 1



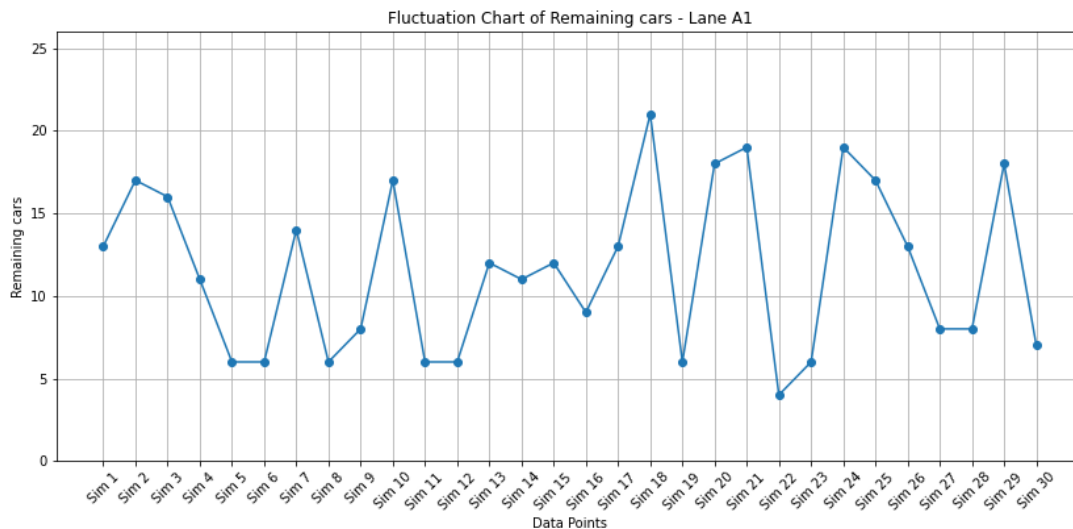


Wir führten 120 Zyklen 30 Mal in insgesamt 2 Stunden Simulation durch, speicherten die durchschnittliche Verweildauer für jede Fahrtrichtung und verwendeten dann Python, um die Schwankungen darzustellen.

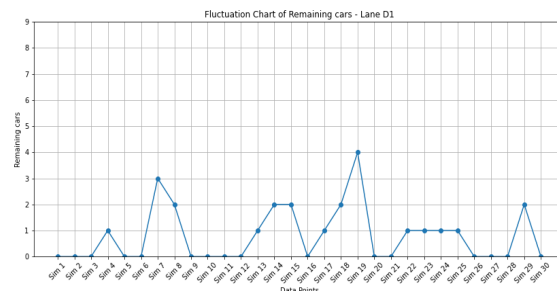
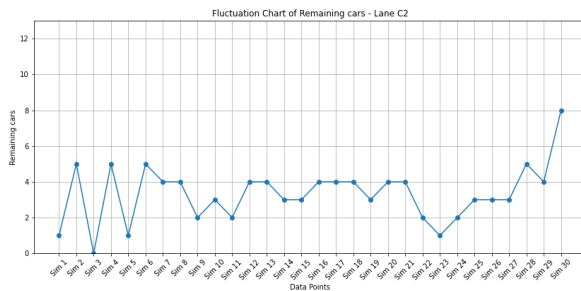
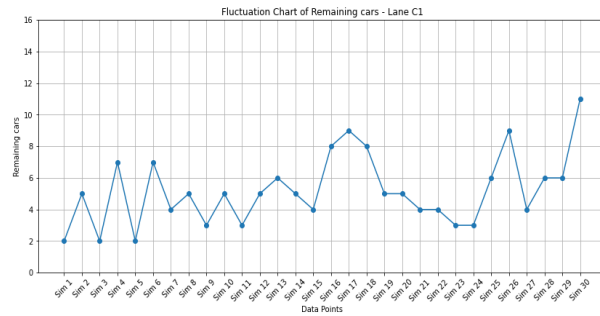
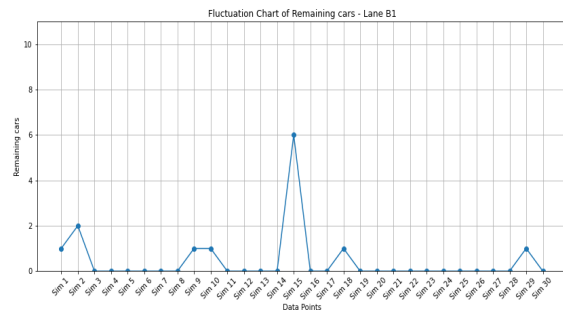
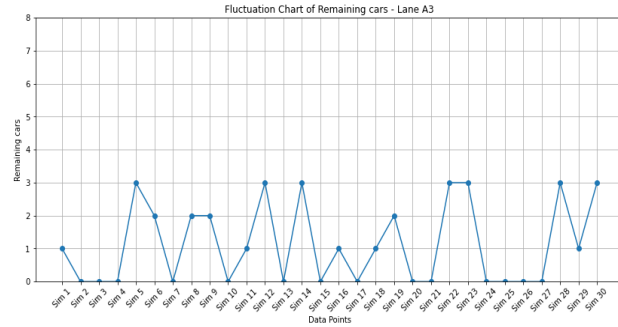
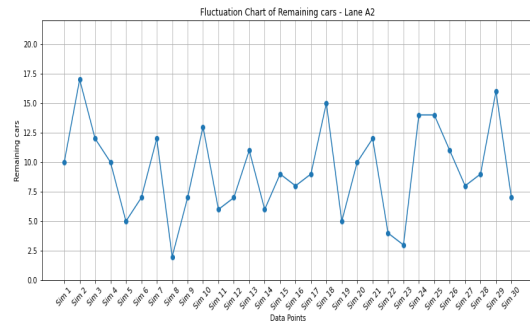
Die durchschnittlichen Verweilzeiten sind in Tabelle 1 angegeben.

## Ergebnisse - Aufgabe 1b

Ebenso haben wir die durchschnittlichen verbleibenden Fahrzeuge für jede Simulation in einer Liste gespeichert und diese Liste dann verwendet, um mit Python die folgenden Fluktuation-Diagramme zu erstellen.



Fahrtrichtungen	A1	A2	A3	B1	C1	C2	D1
Durchschnittliche Fahrzeuge	11.57	9.3	1.13	0.43	5.2	3.33	0.8

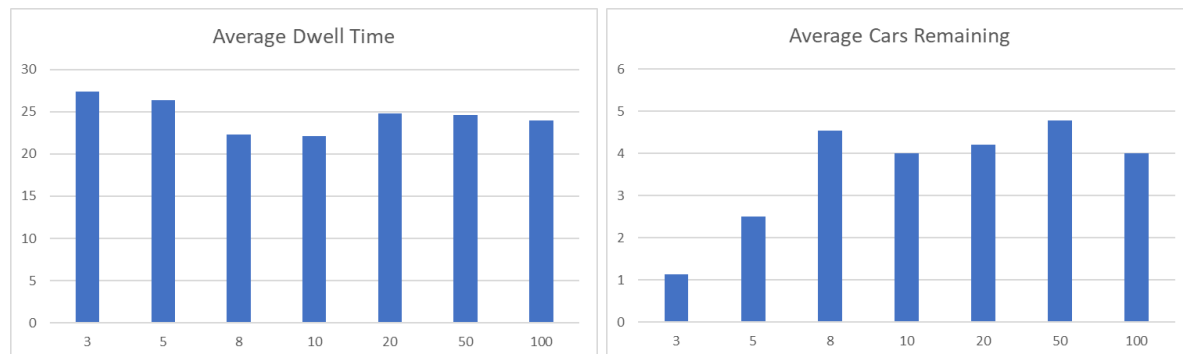


## Ergebnisse - Aufgabe 2

Um die Genauigkeit zu erhöhen, haben wir die Simulation 1000 Mal ausgeführt und folgende Ergebnisse gefunden:

Size of Lane A3	Average Dwell Time	Average Cars Remaining
3	27.4	1.13
5	26.37	2.5
8	22.31	4.54
10	22.1	4.0

20	24.84	4.2
50	24.64	4.79
100	23.95	4.0



Daraus können wir ableiten, dass die Verweildauer von 8 auf 10 abnimmt und danach aufgrund der großen Anzahl an Fahrzeuge im Bereich von 23 bis 24 bleibt. Die durchschnittlich verbleibende Fahrzeuge bleiben ebenfalls im Bereich von 4 bis 5. Daraus können wir schließen, dass 8 bis 10 Fahrzeuge die optimale Größe für Spur A3 wären.

### Ergebnisse - Aufgabe 3

Grünzeit der Linksabbieger	Average Dwell Time	Average Cars Remaining	C1 & C2 Avg. Dwell Time
0-10	27.4	1.13	15.27 & 14.07
0-15	19.37	0.67	21..27 & 18.57
0-20	15.13	0.37	29.63 & 24.17
0-25	8.6	0.47	42 & 36.83
0-30	6.4	0.5	52.52 & 51.93



Daraus haben wir gelernt:

1. Wenn die Grünlichtzeit für A3 verlängert wird, wirkt sich dies direkt auf die Fahrspuren C1 und C2 aus, da diese weiter nach hinten verschoben werden müssen. Wenn wir beispielsweise die Grünzeit für A3 auf 0–20 festlegen, müssen wir zur Vermeidung von Kollisionen die Grünzeit von C1 und C2 auf 24–39 ändern.
2. Wenn wir die Grünlichtzeit für A3 um 5 Sekunden erhöhen, werden relativ gute Ergebnisse vor allem für Spur A3 und nicht allzu schlechte Ergebnisse für Spur C1 und C2 erzielt.

## Einschränkung

Wenn an Position 3 ein Fahrzeug wartet und davor Leerraum ist. Wenn dieses Fahrzeug auf A3 fährt. Das hintere Fahrzeug, wenn es zu A2 gehört, sollte an Position 2 fahren und den Leerraum entfernen, aber es fährt an Position 3 und „Leer“ wird nicht entfernt.

A2

| car14 | car15 | Empty | carfromA316 | car17

A3

| car1

A2

| car15 | Empty | car17

A3

| carfromA316

**ENDE**