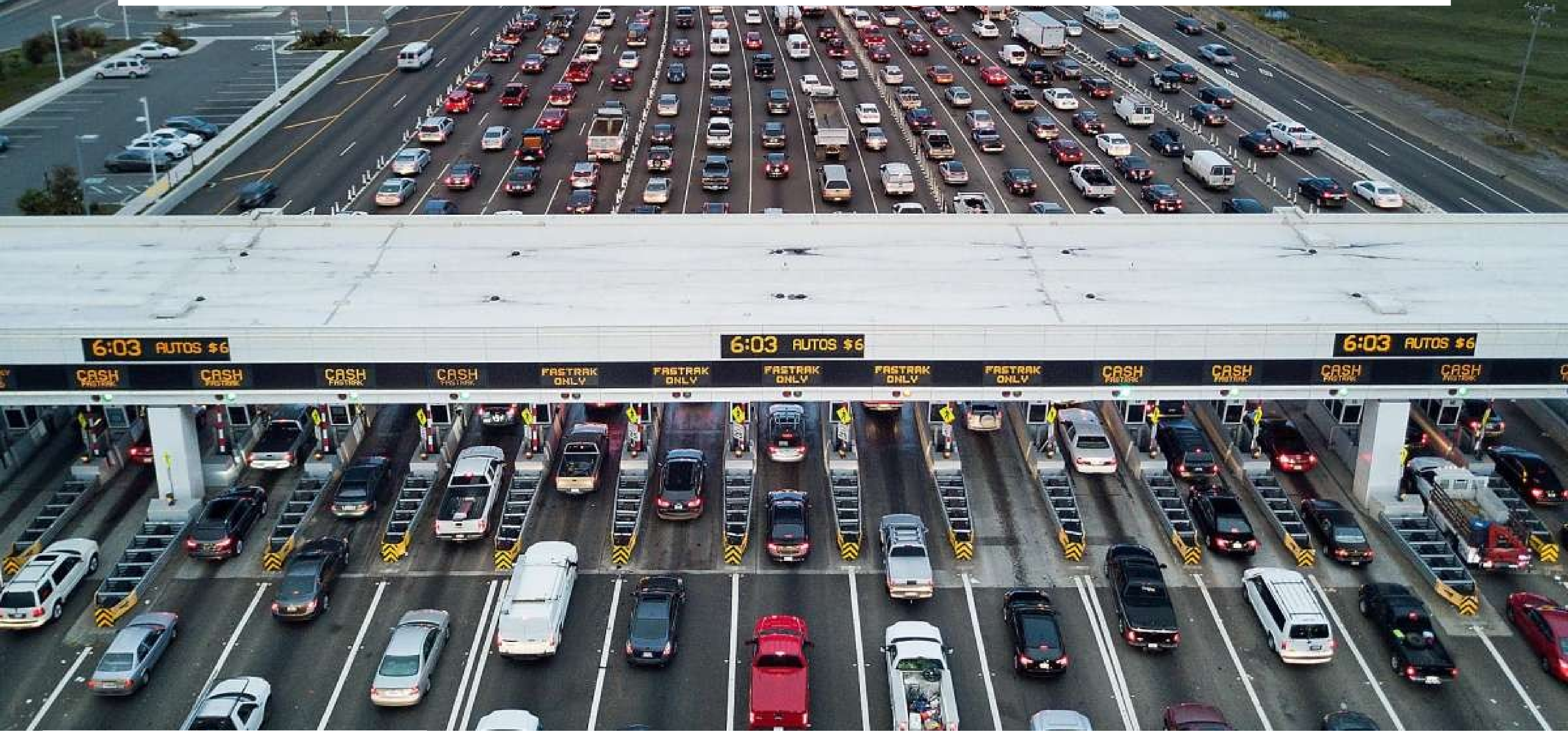
Modeling Toll Plaza Behavior Using Queuing Theory

by Abdellah Ait Elmouden



Introduction

- When a toll plaza is designed, choosing the right number of tollbooths is a critical issue.
- We try to determine the optimal number of tollbooths by creating a model for traffic in a toll plaza.
- We break the travel process in a toll plaza into two stages: toll collection and merging.
- We apply Queuing Theory to each stage, modeling each stage as a queuing system

Problem Statement

- State highway department is planning a new toll exit for an existing turnpike.
- The number of toll booths to put at the exit is in question.
- Keep costs low by having as few booths as possible

Drivers behave when they approach the toll plaza

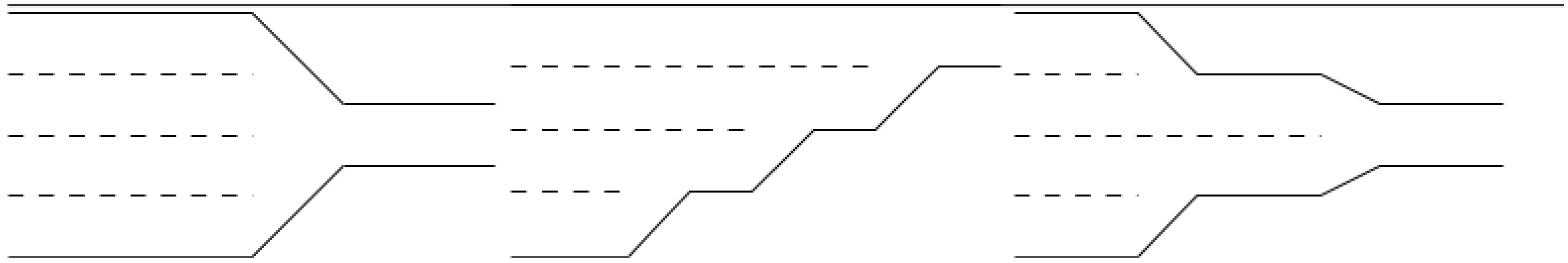
- 1. Cars arrive and merge to the toll Plaza,
- 2. Cars enter the queuing area and stop at the end of a tollbooth line (track).
- 3. Drivers may choose a tollbooth that already has a long line over a shorter one.
- 4. Drivers can perform dangerous maneuvers that lead to collisions.
- 5. Driver Pay toll at the toll booth.
- 6. Driver leave the tollbooths and merge to traffic lane.

Assumptions

- The traffic flow is constant in a short period.
- The time between two cars entering the toll plaza is of exponential distribution.
- The traffic streams fan out into tollbooths smoothly and evenly.
- The drivers are delayed by waiting in lines for toll collection.
- The drivers are delayed by toll collection, and the delay is distributed exponentially.
- The drivers are delayed by the merging process after leaving the tollbooths.
- The toll plaza adopts the side merging layout at the exit.

Wasted Time - Merging Point

- The total delay by the entire merging process is more complicated to analyze.
- We shall first consider the simple merging process when cars from two lanes merge into one



Estimates of Constants

- Number of incoming lanes (N): The typical range of number of lanes on a highway (in one direction) is about 1 to 6. I will consider one values of N =1 in my simulation.
- Total traffic flow (Φ): The maximum traffic flow per lane is 2000 (1/hr) . While N ranges from 1 to 6,
- we would consider various values for traffic flows, including heavy and light traffic conditions.
- Service rate at a tollbooth (μ_A): The service rate of a tollbooth is about 350 vehicles per hour [2, 3].
- Service rate at a merging point: when merging does not occur (μ_0) μ_0 is the service rate when there is only one car in the merging point system. This value is the time for a car to drive through the merging area at the average highway speed. $\mu_0 = 3600/1.1932 = 3017.1$ (1/hr)

Calculation

- Wasted Time - Tollbooth

$$w_A = \frac{1}{\mu_A - \Phi/T}$$

- The average wasted time - Merging Point

$$t_{diff}(\lambda) = t_{sys}(\lambda) - \frac{1}{\mu_0} = \frac{1}{\mu_B - \lambda} + \frac{\mu_B - \mu_0}{\lambda(\mu_B - \mu_0) + \mu_0\mu_B} - \frac{1}{\mu_0}$$

- Wasted Time - Total Merging Process

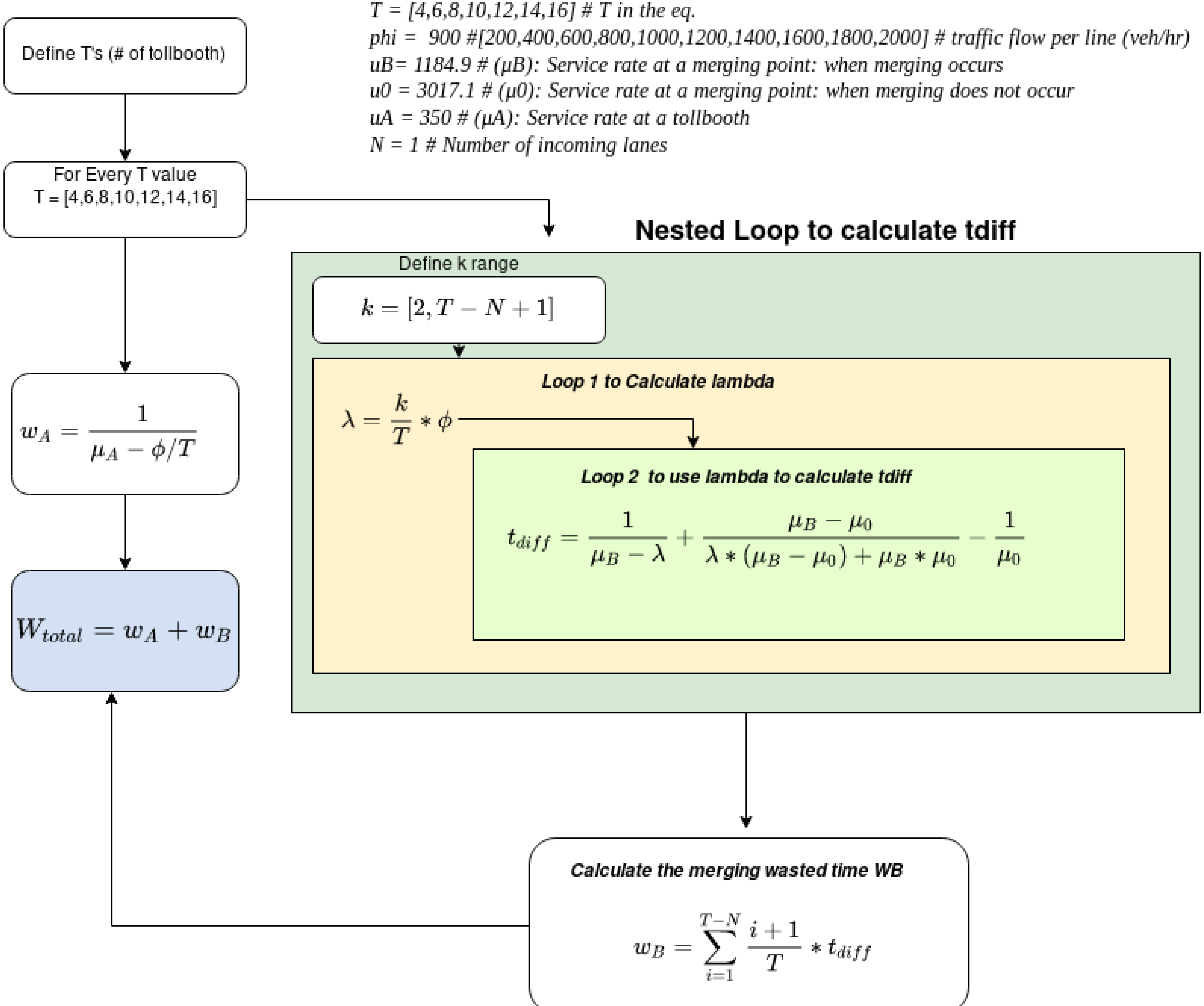
$$w_B = \sum_{i=1}^{T-N} \frac{i+1}{T} \times t_{diff}\left(\frac{i+1}{T} \times \Phi\right)$$

- Wasted Time - Total Merging Process

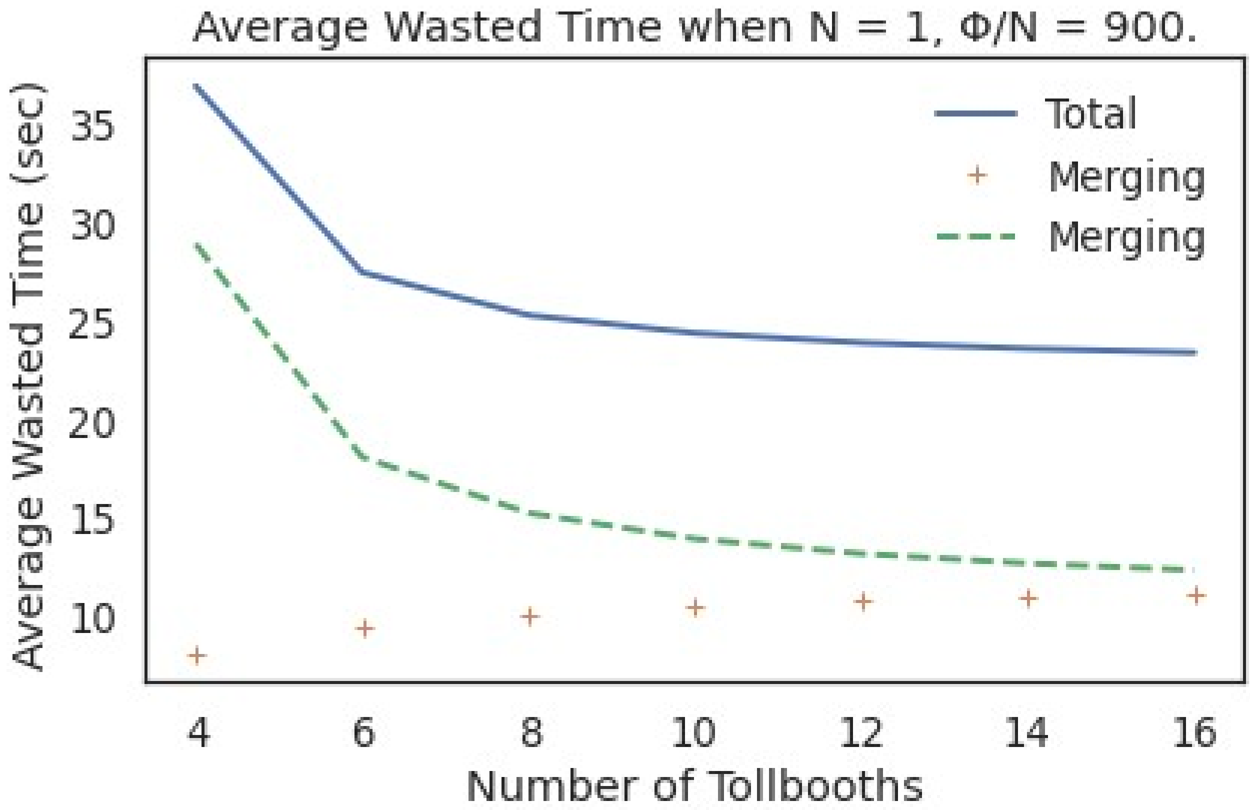
$$w_{total} = w_A + w_B = \frac{1}{\mu_A - \Phi/T} + \sum_{i=1}^{T-N} \frac{i+1}{T} \times t_{diff}\left(\frac{i+1}{T} \times \Phi\right)$$

Simulation I

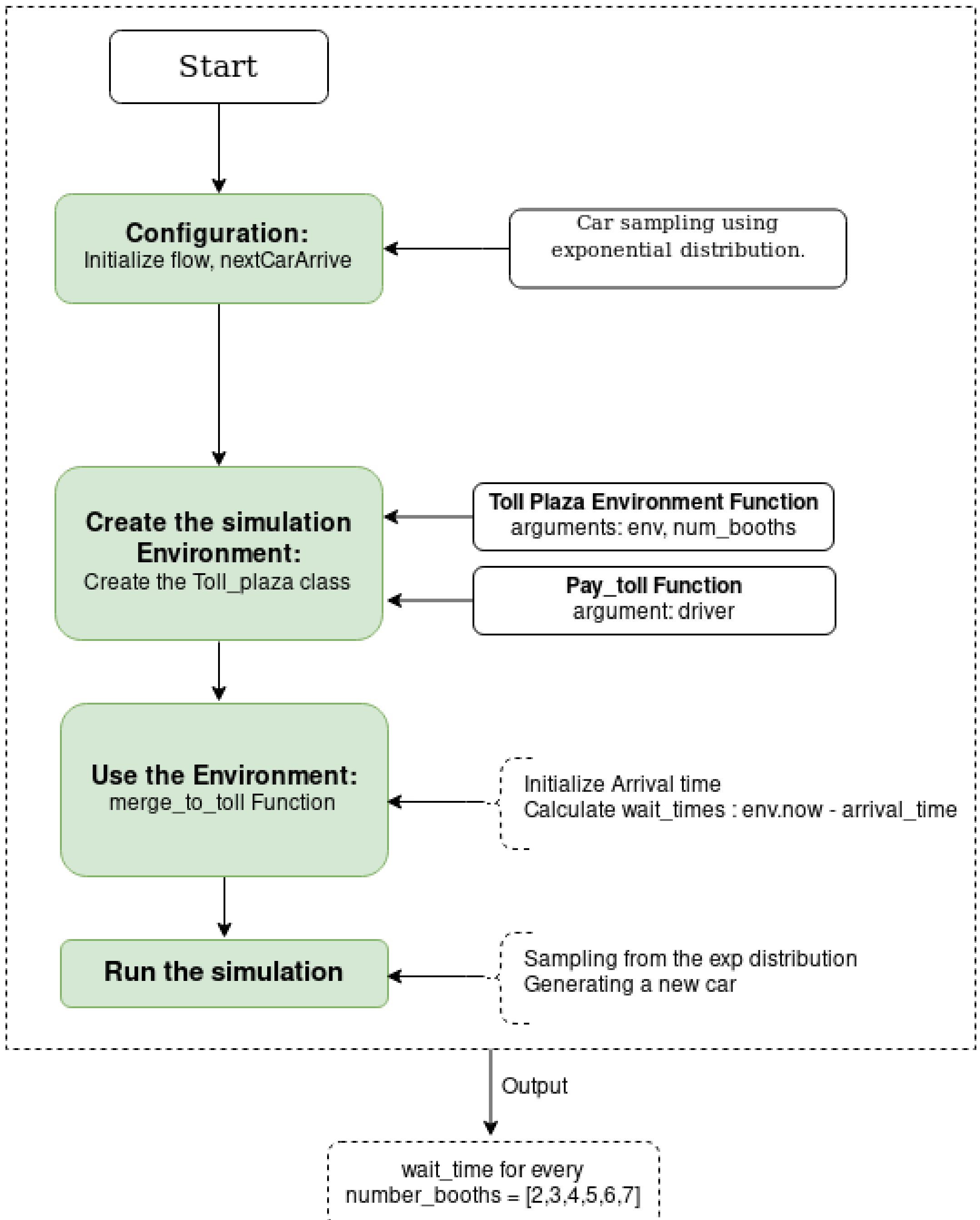
```
T = [4,6,8,10,12,14,16] # T in the eq.
phi =  900 #[200,400,600,800,1000,1200,1400,1600,1800,2000] # traffic flow per line (veh/hr)
uB= 1184.9 # (μB): Service rate at a merging point: when merging occurs
u0 = 3017.1 # (μ0): Service rate at a merging point: when merging does not occur
uA = 350 # (μA): Service rate at a tollbooth
N = 1 # Number of incoming lanes
```



Simulation I



Simulation II – Using Simpy



Simulation II – Using Simpy

```python
#####################################

def main():

    # Setup
    random.seed(42)
    num_booths = get_user_input()   # The number of toll booths to put at the exit is in questio

    # Run the simulation

    env = simpy.Environment()
    env.process(run_toll_plaza(env, num_booths))
    env.run(until=90)

    # View the results
    mins, secs = get_average_wait_time(wait_times)
    print(
            "Running simulation...",
            f"\nThe average wait time is {mins} minutes and {secs} seconds.",
        )

if __name__ == "__main__":
    main()
```

```
Input # of booths working: 4
Running simulation...
The average wait time is 33.0 minutes and 43.0 seconds.
```

Conclusion

- Explore more Simpy
- Improving the Model

Characteristics	Symbol	Description
Arrival pattern(A)	M	Exponential distribution
Service pattern(B)	M	Exponential distribution
	G	General distribution
Number of servers(X)	$1, 2, \ldots, \infty$	
System capacity(Y)	$1, 2, \ldots, \infty$	
Queue discipline(Z)	FCFS	First come, first served

References

- Modeling toll plaza behavior using queuing theory, 2011