

WYDZIAŁ NAUK ŚCISŁYCH I TECHNICZNYCH

Symulacje Komputerowe

Sprawozdanie “Symulacja zdarzeń dyskretnych”

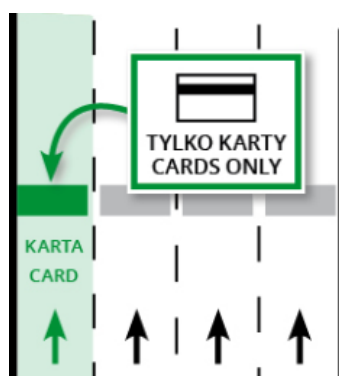
Adam Talarczyk, Mateusz Wrzoł

Uniwersytet Śląski, Sosnowiec, 2021

1 Zadanie 1

Wykonać symulację bramek autostradowych uwzględniając następujące założenia:

- Dostępne są cztery bramki. Na trzech bramkach kierowcy mogą płacić kartą i gotówką. Na jednej bramce kierowcy mogą płacić tylko kartą.
- Czas trwania obsługi na bramce w przypadku płatności gotówką jest opisany rozkładem normalnym o średniej M1 minuty i odchyleniu standardowym SD1 minuty.
- Czas trwania obsługi na bramce w przypadku płatności kartą jest opisany rozkładem normalnym o średniej M2 minuty i odchyleniu standardowym SD2 minuty.
- Odstęp czasu pomiędzy nadjeżdżającymi samochodami jest opisany rozkładem wykładniczym o parametrze $\lambda = L$ (wartość oczekiwana wynosi $1/L$ minuty, L odpowiada średniej liczbie pojazdów na minutę).
- Połowa kierowców zamierza dokonać płatności kartą a druga połowa gotówką. Nadjeżdżający kierowcy wybierają dostępną bramkę z najkrótszą kolejką. Płacący gotówką mają do wyboru 3 bramki. Płacący kartą wybierają spośród 4 bramek. Wartości parametrów



Rysunek 1: Wizualizacja

Wartości parametrów SD1, SD2, M1 i M2 należy przyjąć według własnego uznania.

Wyznaczyć symulacyjnie zależność pomiędzy średnią liczbą pojazdów na minutę (L) i średnim czasem oczekiwania na przejazd przez bramki. Przedstawić tę zależność na wykresie.

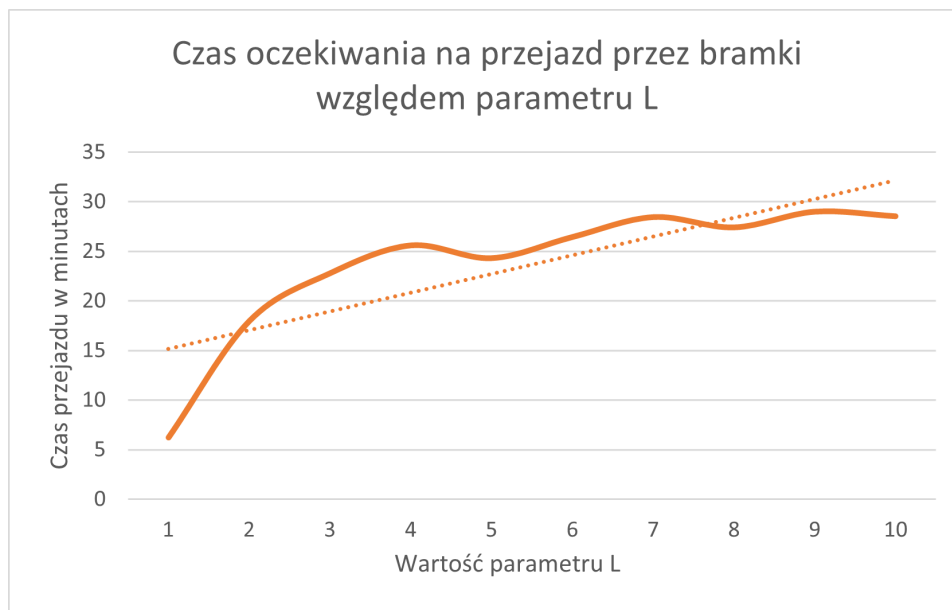
W sprawozdaniu należy zamieścić treść zadania, kod źródłowy rozwiązania z opisem, wyniki symulacji i wykres.

1.1 Rozwiązanie

Zachowanie kierowców zostało zdefiniowane w osobnej, parametryzowanej funkcji. Każdemu kierowcy przypisać można alias służący do wyświetlania logów, dostępne bramki (parametr `gates`) oraz czas przejazdu przez bramki wraz z odchyleniem standardowym. Na podstawie funkcji `driver` (Listing 1) utworzono obiekty dla kierowców płacących gotówką i kartą oraz samą gotówką. W założeniach przyjęto, że dostępne bramki autostradowe oznaczane są jako `gate1`, `gate2`, `gate3`, `gate4`. Kierowcy płacący tylko i wyłącznie gotówką mają do dyspozycji bramki `gate2`, `gate3` oraz `gate4`, ponieważ `gate1` obsługuje tylko karty. Przy założeniu, że płatność kartą odbywa się szybciej, przyjęto parametr `M1=3`, `SD1=2` dla płatności gotówką, oraz `M2=2`, `SD1=1` dla płatności kartą.

Symulacja jest również sparametryzowaną funkcją (Listing 2), w której można ustawić parametr `L` (`cars_per_minute`), obiekty kierowców (`cash/card_drivers_trajectory`) oraz łączną liczbę kierowców. Symulacja wykonywana jest pętli, dla parametru `L` z przedziału od 1 do 10, przy założeniu 50 kierowców płacących kartą jak i 50 kierowców płacących gotówką. Dla każdego parametru `L` wyniki zostały powtórzone 10 razy a następnie uśrednione. Jak widać na listingu 2, definiowane są dostępne bramki oraz ilość kierowców wraz z ich typem (`cash/card_drivers_trajectory`). Wyniki, czyli średni czas oczekiwania do przejazdu oraz średni czas płacenia przy bramkach zostały wyeksportowane do arkusza kalkulacyjnego.

Na podstawie danych utworzony wykres (Rysunek 2), który przedstawia czas oczekiwania na przejazd względem parametru `L`, przy łącznej ilości 100 kierowców. Uśrednione dane z 10 prób przedstawione są w tabeli 1.



Rysunek 2: Wykres przedstawiający czas oczekiwania na przejazd przez bramki względem parametru L przy 100 kierowcach

Parametr L	Średni czas przejazdu przez bramki
1	6.229830525
2	17.96801664
3	22.78144309
4	25.58758123
5	24.29923589
6	26.43225118
7	28.43579926
8	27.39886657
9	28.97951878
10	28.52141024

Tablica 1: Tablica z wynikami uzyskanymi w symulacji

Na podstawie powyższych danych i wykresu widać, że czas oczekiwania na przejazd przez bramki rośnie wraz z parametrem L.

1.2 Kod źródłowy

```

1 # Title      : Discrete events simulation
2 # Objective  : -
3 # Created by : Adam Talarczyk, Mateusz Wrzol
4 # Created on : 01.05.2021

```

```

5 library(simmer)
6
7 driver <- function(alias, gates, proceed_time, standard_
  deviation) {
8   return(
9     trajectory("Drivers' path") %>%
10      log_(paste(alias, "driver has arrived")) %>%
11      select(gates, policy = "shortest-queue") %>%
12      seize_selected() %>%
13      timeout(function() abs(rnorm(1, proceed_time, standard_
deviation))) %>%
14      release_selected() %>%
15      log_(paste(alias, "driver drove away"))
16   )
17 }

```

Listing 1: Funkcja związana z zachowaniem kierowcy

```

1 # Title      : Discrete events simulation
2 # Objective : -
3 # Created by: Adam Talarczyk, Mateusz Wrzol
4 # Created on: 01.05.2021
5
6 simulate_gates <- function(cars_per_minute = 2,
7                             cash_drivers_trajectory,
8                             card_drivers_trajectory,
9                             card_drivers_num,
10                             cash_drivers_num) {
11   gates <- simmer("gates") %>%
12     add_resource("gate1", 1) %>%
13     add_resource("gate2", 1) %>%
14     add_resource("gate3", 1) %>%
15     add_resource("gate4", 1) %>%
16     add_generator("CASH event ",
17                  cash_drivers_trajectory,
18                  function() { c(0, rexp(cash_drivers_num -
19    1, cars_per_minute), -1) }) %>%
20     add_generator("CARD event ",
21                  card_drivers_trajectory,
22                  function() { c(0, rexp(card_drivers_num -
23    1, cars_per_minute), -1) })
24
25   gates %>% run(until = 2000)
26
27   return(
28     result <- gates %>%
29       get_mon_arrivals() %>%
30       transform(waiting_time = end_time - start_time -
activity_time)
31   )

```

30 }

Listing 2: Implementacja symulacji

```
1 # Title      : Discrete events simulation
2 # Objective  : -
3 # Created by: Adam Talarczyk, Mateusz Wrzol
4 # Created on: 01.05.2021
5
6 library(simmer)
7 library("xlsx")
8
9 source('simulation/driver.R')
10 source('simulation/simulate_gates.R')
11
12 gates_for_card_drivers <- c("gate1", "gate2", "gate3", "
   gate4")
13 gates_for_cash_drivers <- c("gate2", "gate3", "gate4")
14
15 cash_drivers_trajectory <- driver("CASH", gates_for_cash_
   drivers, 3, 2)
16 card_drivers_trajectory <- driver("CARD", gates_for_card_
   drivers, 2, 1)
17
18 results <- data.frame("drivers", "lambda", "avg_waiting_time
   ", "avg_activity_time")
19
20 for (j in seq(1, 10, by = 1)) {
21   cars_per_minute <- j
22   drivers <- 50
23   for (i in seq(1, 10, by = 1))
24   {
25     result <- simulate_gates(cars_per_minute,
26       cash_drivers_trajectory,
27       card_drivers_trajectory,
28       cash_drivers_num = drivers,
29       card_drivers_num = drivers)
30
31     results <- rbind(results, c(sum(result$finished), cars_
   per_minute, mean(result$waiting_time), mean(result$
   activity_time)))
32     print(paste("Avg time for", sum(result$finished), "
   drivers:", mean(result$waiting_time), "minutes"))
33   }
34 }
35
36 export_dataset <- function(dataset) {
37   write.xlsx(dataset, file = "simulation/export/data.xlsx",
   sheetName = "discrete_events")
38 }
```

```
39  
40 export_dataset(results)
```

Listing 3: Wywołanie symulacji kilkakrotnie dla różnych wartości i zapis danych do pliku