

# Optimization and Simulation

## Discrete Events Simulation

Michel Bierlaire

Transport and Mobility Laboratory  
School of Architecture, Civil and Environmental Engineering  
Ecole Polytechnique Fédérale de Lausanne



# Simulation of a system

## Keep track of variables

- Time variable  $t$ : amount of time that has elapsed.
- Counter variables: count events having occurred by  $t$
- System state variables.

## Events

- List of future events sorted in chronological order
- Process the next event:
  - remove the first event in the list,
  - update the variables,
  - generate new events, if applicable (keep the list sorted),
  - collect statistics.

# Discrete Event Simulation: an example

## Yuki at Satellite

- Yuki has applied to be a waiter at Satellite
- According to his experience, he pretends to be able to serve in average one customer per minute.
- In order to make the decision to hire Yuki or not, the manager wants to know:
  - In average, how much time will a customer wait after her arrival, until being served?
  - If Yuki will need extra hours to serve everybody?



**SATELLITE**  
bar · concerts · cafés-théâtres

# Discrete Event Simulation: an example

## Context

- When a customer arrives, she is served if Yuki is free. Otherwise, she joins the queue.
- Customers are served using a “first come, first served” logic.
- When Yuki has finished serving a customer,
  - he starts serving the next customer in line, or
  - waits for the next customer to arrive if the queue is empty.
- The amount of time required by Yuki to serve a customer is a random variable  $X_s$  with pdf  $f_s$ .
- The amount of time between the arrival of two customers is a random variable  $X_a$  with pdf  $f_a$ .
- Satellite does not accept the arrival of customers after time  $T$ .

# Discrete Event Simulation: an example

## Variables

Time:	$t$	
Counters:	$N_A$	number of arrivals
	$N_D$	number of departures
System state:	$n$	number of customers in the system

## Event list

- Next arrival. Time:  $t_A$
- Service completion for the customer currently being served. Time:  $t_D$  ( $\infty$  if no customer is being served).
- The bar closes. Time:  $T$ .

## List management

- The number of events is always 3 in this example.
- We just need to update the times, and keep them sorted.

# Initialization

## Variables

- Time:  $t = 0$ .
- Counters:  $N_A = N_D = 0$ .
- State:  $n = 0$ .
- First event: arrival of first customer: draw  $r$  from  $f_a$ .
- Events list:
  - $t_A = r$ ,
  - $t_D = \infty$ ,
  - $T$  (bar closes).

## Statistics to collect

- $A(i)$  arrival of customer  $i$ .
- $D(i)$  departure of customer  $i$ .
- $T_p$  time after  $T$  that the last customer departs.

## Case 1: arrival of a customer

If  $t_A = \min(t_A, t_D, T)$

- Time  $t = t_A$ : we move along to time  $t_A$ .
- Counter  $N_A = N_A + 1$ : one more customer arrived.
- State  $n = n + 1$ : one more customer in the system.
- Next arrival:
  - draw  $r$  from  $f_a$ ,
  - $t_A = t + r$ .
- Service time: if  $n = 1$  (she is served immediately)
  - draw  $s$  from  $f_s$ ,
  - $t_D = t + s$ .
- Statistics:  $A(N_A) = t$ .

## Case 2: departure of a customer

If  $t_D = \min(t_A, t_D, T)$ ,  $t_D < t_A$

- Time  $t = t_D$ : we move along to time  $t_D$ .
- Counter  $N_D = N_D + 1$ : one more customer departed.
- State  $n = n - 1$ : one less customer in the system.
- Service time: if  $n = 0$ , then  $t_D = \infty$ . Otherwise,
  - draw  $s$  from  $f_s$ ,
  - $t_D = t + s$ .
- Statistics:  $D(N_D) = t$ .



## Case 3: after hours

If  $T < \min(t_A, t_D)$

- 1 Customers are still waiting:  $n > 0$ 
  - Time  $t = t_D$ : we move along to time  $t_D$ .
  - Counter  $N_D = N_D + 1$ : one more customer departed.
  - State  $n = n - 1$ : one less customer in the system.
  - Service time: if  $n > 0$ , then
    - draw  $s$  from  $f_s$ ,
    - $t_D = t + s$ .
  - Statistics:  $D(N_D) = t$ .
- 2 No more customers:  $n = 0$ 
  - Statistics:  $T_p = \max(t - T, 0)$ .

# An instance

## Scenario

- Service time: exponential with mean 1.0
- Inter-arrival time: exponential with mean 1.0
- Closing time: 10.0

## An instance (ctd.)

Event	t	NA	ND	n	tA	tD	T
Arrival	0.94	1	0	1	1.48	3.22	10.0
Arrival	1.48	2	0	2	2.01	3.22	10.0
Arrival	2.01	3	0	3	3.16	3.22	10.0
Arrival	3.16	4	0	4	3.44	3.22	10.0
Departure	3.22	4	1	3	3.44	3.49	10.0
Arrival	3.44	5	1	4	3.81	3.49	10.0
Departure	3.49	5	2	3	3.81	3.91	10.0
Arrival	3.81	6	2	4	7.22	3.91	10.0
Departure	3.91	6	3	3	7.22	5.84	10.0
Departure	5.84	6	4	2	7.22	5.88	10.0
Departure	5.88	6	5	1	7.22	6.49	10.0
Departure	6.49	6	6	0	7.22	$\infty$	10.0
Arrival	7.22	7	6	1	7.42	7.38	10.0
...							

## An instance (ctd.)

Event	t	NA	ND	n	tA	tD	T
...							
Departure	7.38	7	7	0	7.42	$\infty$	10.0
Arrival	7.42	8	7	1	8.58	8.42	10.0
Departure	8.42	8	8	0	8.58	$\infty$	10.0
Arrival	8.58	9	8	1	9.64	9.91	10.0
Arrival	9.64	10	8	2	10.7	9.91	10.0
Departure	9.91	10	9	1	10.7	10.7	10.0
After hours	10.7	10	10	0	10.7	10.7	10.0
Finish	10.7	10	10	0	10.7	10.7	10.0

## An instance (ctd.)

Statistics for each customer (rounded)

Cust.	Arrival	Departure	Time
1	0.94	3.22	2.28
2	1.48	3.49	2.02
3	2.01	3.91	1.9
4	3.16	5.84	2.68
5	3.44	5.88	2.45
6	3.81	6.49	2.68
7	7.22	7.38	0.165
8	7.42	8.42	1.0
9	8.58	9.91	1.33
10	9.64	10.7	1.02

## An instance (ctd.)

### Aggregate indicators

- Average time in the system: 1.75
- Yuki leaves Satellite at 10.7

### Realizations

- This represents one draw from the random variables.
- Multiple draws are necessary.
- Remember the pitfalls of simulation.

## Another instance

### Scenario: Yuki works faster

- Service time: exponential with mean 0.2
- Inter-arrival time: exponential with mean 1.0
- Closing time: 10.0

## An instance (ctd.)

Event	t	NA	ND	n	tA	tD	T
Arrival	1.02	1	0	1	3.14	1.38	10.0
Departure	1.38	1	1	0	3.14	$\infty$	10.0
Arrival	3.14	2	1	1	6.97	3.25	10.0
Departure	3.25	2	2	0	6.97	$\infty$	10.0
Arrival	6.97	3	2	1	7.08	7.26	10.0
Arrival	7.08	4	2	2	7.24	7.26	10.0
Arrival	7.24	5	2	3	10.0	7.26	10.0
Departure	7.26	5	3	2	10.0	8.32	10.0
Departure	8.32	5	4	1	10.0	8.51	10.0
Departure	8.51	5	5	0	10.0	$\infty$	10.0
Finish	10.0	5	5	0	10.0	$\infty$	10.0



## An instance (ctd.)

### Statistics for each customer (rounded)

Cust.	Arrival	Departure	Time
1	1.02	1.38	0.355
2	3.14	3.25	0.11
3	6.97	7.26	0.296
4	7.08	8.32	1.24
5	7.24	8.51	1.27

### Aggregate indicators

- Average time in the system: 0.654
- Yuki leaves Satellite at 10.0.
- He stops working at 8.51.

# General framework

$$Z = h(X, Y, U) + \varepsilon_z$$

## State variables $X$

- Time
- Number of customers in the system

## External input $Y$

Arrival of customers

## Control $U$

Serving customers

# General framework

## Indicators $Z$

- Time of each customer in the system.
- Average time in the system.
- Time at which Yuki leaves Satellite.

## Statistics

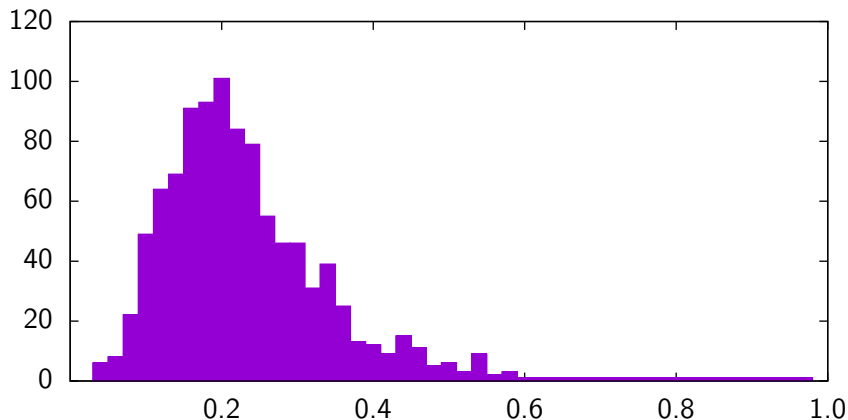
- Numbers reported above are based on one instance.
- Insufficient to draw any conclusion (remember Kid City)
- Their distribution has to be investigated.
- Many realizations are necessary.

## Possible confusion in terminology

- The desired indicator  $Z$  may be a statistic from the simulator:
  - Mean time spent in the system
  - Maximum time spent in the system
  - Number of customers spending more than  $\alpha$  min. in the system
- Still, each of them is a random variable, and statistics must be considered.
  - 5% quantile of the mean time spent in the system
  - Mean of the maximum time spent in the system
  - Mean of the mean time spent in the system
  - Standard deviation of the mean time spent in the system
  - Standard deviation of the number of customers spending more than  $\alpha$  in the system
- Drawing histograms is highly recommended

# Statistics

Mean time spent in the system (service time: 0.2, arrival: 1.0)



Mean: 0.23, but 4.86%  $\geq 0.54$

# Conclusion

## Strengths of discrete event simulation

- Decomposition of a complex system into simple subsystems.
- Easy to mimick a real system

## Challenges

- Importance of book-keeping.
- Easy to be overwhelmed by generated data. Careful statistical analysis is needed.
- Importance to distinguish between an indicator and the statistics of its distribution.