Optimization and Simulation
Simulation Exercise 2: Event-Based Simulation

Bilal Farooq

March 08, 2013

TRANSP-OR

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Topic: Event-based Queueing Simulation

- 2.1: Simulate a single road
- 2.2: Simulate a road network

Roads Intersection as Queue

2.1 Basics: Queue State and Event

- Single queue has one state variable
  - # of jobs in waiting
- Event list of a single queue has always 3 entries
  - Next arrival
  - Next service/departure
  - Termination
- But for a network, the event list can be arbitrary size…
- In problem 2.1 we program the event queue for 1 road network

2.1 Basics: Data structure

- Queue event in case of a road network
  - Time of the event (*real*)
  - Road/Link on which the event occurred (*integer*)
  - Type of the event (*integer*)
- Convention for integer event types
  - ARRIVAL = 1
  - DEPARTURE = 2
- Termination when the event queue is empty

2.1 Creating Events

- We use `NewEvent` function to create new ‘event’ object
- `help NewEvent`

```
function event = NewEvent(time, link,
                                  type)
    event.time = time;
    event.link  = link;
    event.type = type;
end
```

2.1 Queue Management

- We use `UpdatedQueue` function to manage a sorted row vector of events
- `help UpdatedQueue`

- Try creating and adding events to the queue…

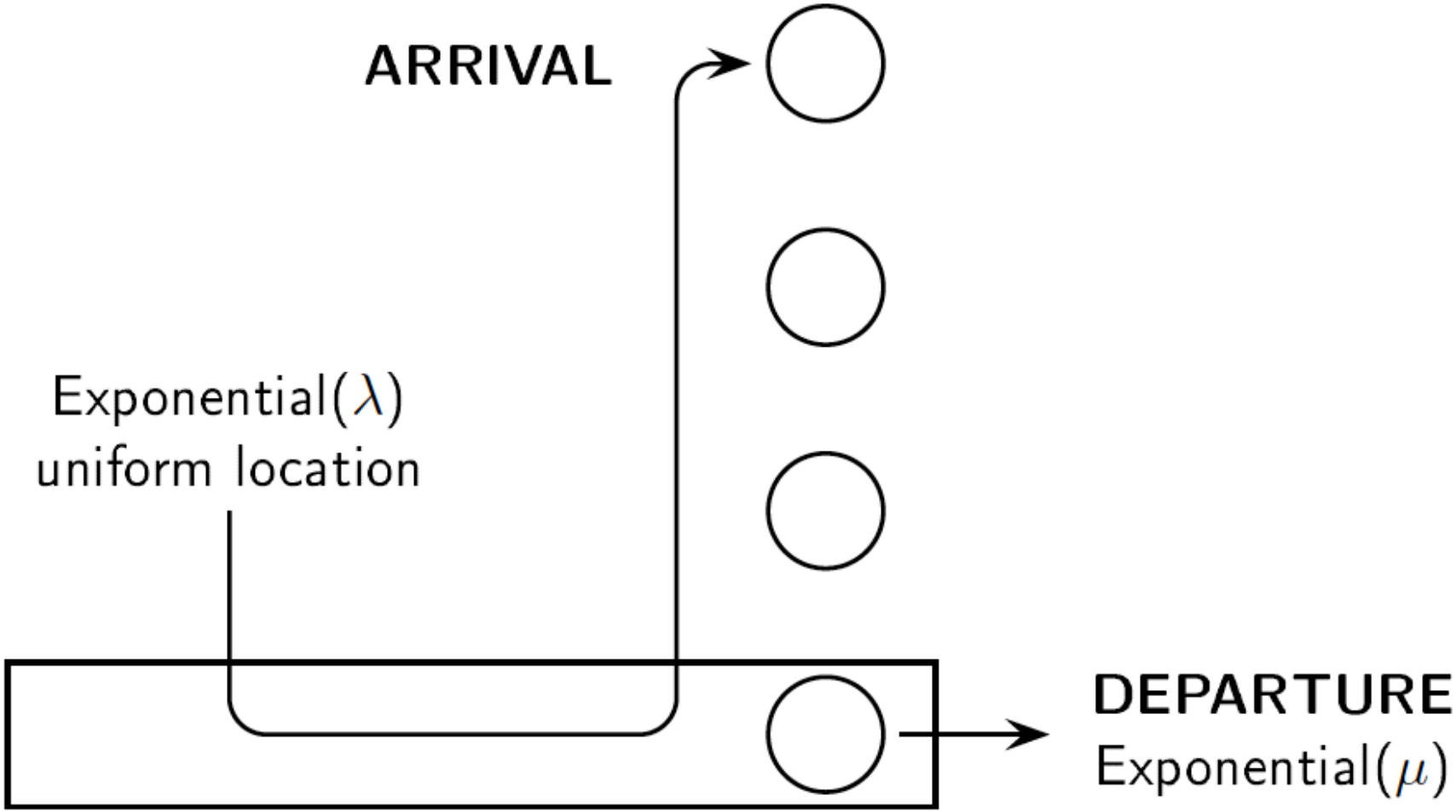
2.1 Queue Management Example

```
time = 42
link = 5
type = 1
event1 = NewEvent(time, link, type)
queue = struct([])
queue = UpdatedQueue(queue, event1)
event2 = NewEvent(43, 4, 1)
event3 = NewEvent(39, 3, 1)
queue = UpdatedQueue(queue, event2)
queue = UpdatedQueue(queue, event3)
```

2.1 Properties of a Road

- Defined in `NewRoad1()`
- Free-flow travel time t_0
- Rate of arrival λ of vehicles
  - Imagine vehicles pulling out of their driveways
  - Locations of the driveways are uniformly distributed
- Service rate μ (road capacity)

2.1 Modelling Road as a Queue



2.1 Vehicle Entry → ARRIVAL

- New vehicle enters on road i at time t_{arrv}
- Vehicle entry point is uniformly distributed over the road
- Free-flow travel time on road i is $t_0(i)$
- Vehicle arrives downstream after $t_{free} \sim U(0, t_0(i))$
- Create ARRIVAL event on road i at time $t = t_{arrv} + t_{free}$

2.1 ARRIVAL → DEPARTURE

- ARRIVAL event occurs on road i at time t_{arrv} and reaches the queue in t_{free}
- Increase queue on road i by 1
- If queue size now is exactly 1
  - Service time for vehicle is $t_{service} \sim \text{Exp}(\mu_i)$
  - Create DEPARTURE event on i at time $t + t_{service}$

2.1 DEPARTURE → DEPARTURE

- DEPARTURE event occurs on road i at time t
- Reduce queue on road i by 1
- If the queue on road i is still larger than 0
  - Service time for vehicle is $t_{service} \sim \text{Exp}(\mu_i)$
  - Create DEPARTURE event on i at time $t + t_{service}$

Note: This specification applied for a single road
Will later be extended for a road network

2.1 Simulate a Single Road

- `Scenario = NewRoad1()` creates a single road
- Implement the function `QueueingSimulation1`
  - Description of the function will help you in exactly what to do
- For testing run `Road1Test`
  - Create a scenario, feed it into the simulation, analyze
  - Use the new `DrawQueues` and `DrawNetwork` functions

2.2 Properties of a Road Network

- Now, road Network has more than one road
- Roads have the same properties as before
- Roads are connected at intersections
- Every intersection has the following properties
  - Turning probability p_{ij} from i to j
  - Network exit probability p_{i*}
  - $p_{i*} + \Sigma_j p_{ij} = 1$

2.2 DEPARTURE → ARRIVAL/DEPARTURE

- DEPARTURE event occurs on road i at time t
- Reduce queue size on road i by one
- If the queue on road i is still larger than zero
  - Service time for next vehicle is $t_{service} \sim \mathrm{Exp}(\mu_i)$
  - Create DEPARTURE event on road i at time $t + t_{service}$
- Vehicle
  - Turns into road j with probability p_{ij}
  - Leave network with probability p_{i*}
- If vehicle turns into road j (i.e. doesn't leave the network)
  - Create ARRIVAL event on road j at time $t + t_0(j)$

2.2 Simulate a Road Network

- `scenario = NewNetwork1()` create a small network
  - The description can be found in the function
  - It is an extension of `NewRoad1` for more than 1 road
- Implement the function `QueueingSimulation2`
  - Again, it is the extension of `QueueingSimulation1` for more than 1 road
  - Reuse the code from 2.1
- For testing run `Network1Test`

Appendix A: Implemented Functions

- `function event = NewEvent(time, link, type)`
- `function newQueue = UpdatedQueue(oldQueue, newEvent)`
- `function scenario = NewRoad1()`
- `function scenario = NewNetwork1()`
- `function DrawNetwork(scenario, times, queues)`
- `function DrawQueues(times, queues)`
- **`function [times, queues] = QueueingSimulation1(scenario)`**
- **`function [times, queues] QueueingSimulation2(scenario)`**
  - Test functions…