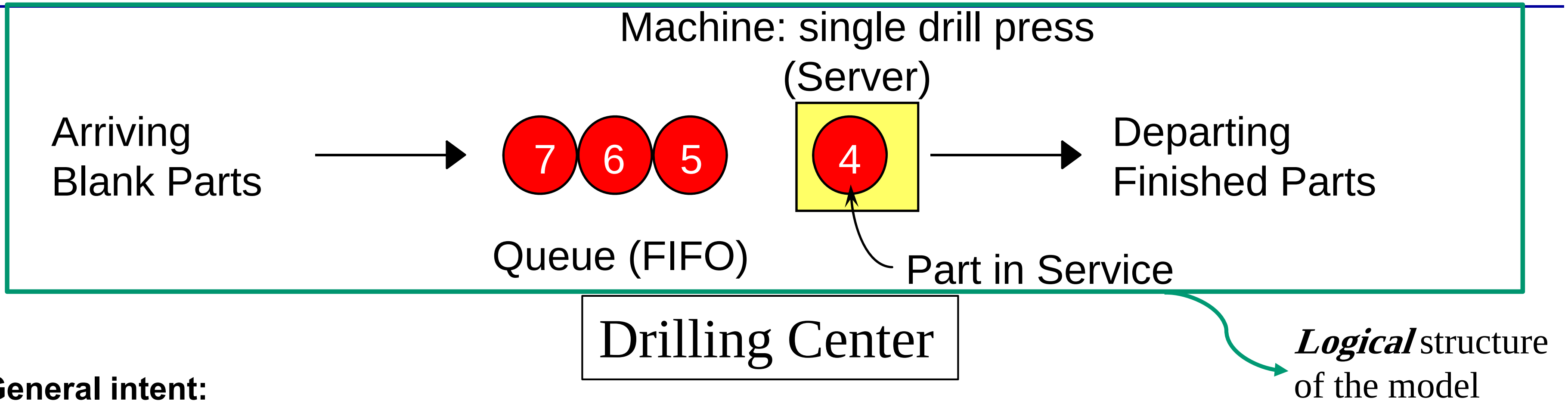
0405324: Stochastic System Simulation

Lecture 2: Fundamental simulation concept (simulation of the single server queue)

Content

- **Underlying ideas, methods, and issues in simulation**
- **Software-independent (setting up for Arena)**
- **Example of a simple processing system**
  - Decompose problem
  - Terminology
  - Simulation by hand
  - Some basic statistical issues
- **Overview of a simulation study**

The System:
A Simple Processing System



- **General intent:**
  - Estimate *expected production*
  - *Waiting time in queue*, *queue length*, *proportion of time machine is busy*

- **Specify the numerical aspects including how the simulation starts and stops and time units**
  - Example: the system starts at time 0 minutes with no parts in the queue and the drill press is idle: ***empty-and-idle*** assumption.

- **Time units**
  - Declare *base time units* with which time will be measured for internal calculations, outputs
  - Can use different units in different places … must declare: use the most appropriate, familiar and convenient time units: **minutes for service time**, **hours for machine up times**
  - Arena allows you to define **time units for inputs** but you must declare the **base time unit** to which will be converted all times are converted during the simulation runs and for outputs.
  - Be reasonable (interpretation, roundoff error): do not use seconds for a simulation of 20 years

Model Specifics

- **Initially (time 0) empty and idle**
- **Base time units: minutes**
- **Input data (assume given for now …), in minutes:**

Will not be given in the exam

Part Number	Arrival Time	Interarrival Time	Service Time
1	0.00	1.73	2.90
2	1.73	1.35	1.76
3	3.08	0.71	3.39
4	3.79	0.62	4.52
5	4.41	14.28	4.46
6	18.69	0.70	4.36
7	19.39	15.52	2.07
8	34.91	3.15	3.36
9	38.06	1.76	2.37
10	39.82	1.00	5.38
11	40.82	.	.
.	.	.	.
.	.	.	.

- **Stop when 20 minutes of (simulated) time have passed: any units present in the system at that time are never finished**

Goals of Study: Output Performance Measures

- ***Total production*** **of parts (number of completed units) over the simulation run (*P*), i.e. during the 20 min of simulation**
- ***Average waiting time*** **of parts in queue (W_q):**

$$\frac{\sum_{i=1}^{N} WQ_i}{N}$$

N = no. of parts completing queue wait
WQ_i = waiting time in queue of ith part
Know: WQ_1 = 0 (why?)
$N \geq 1$ (why?)

→ discrete-time (discrete-parameter) statistic since it refers to data : based on the first, second, … observations

- ***Maximum waiting time*** **of parts in queue (W_q^*):**

$$\max_{i=1,\ldots,N} WQ_i$$

worst case
small is good

Goals of Study: Output Performance Measures (cont'd.)

- ***Average total time*** **of parts** ***in the system*** **or** ***Cycle time*** **(W_s),**

$$\frac{\sum_{i=1}^{P} TS_i}{P} \qquad TS_i = \text{total time in system of } i\text{th part}$$

P**: total units completed**

(Cycle time = part's waiting time in the queue + service time)

- ***Maximum total time*** **of parts** ***in system*** **(W_s^*),**

$$\max_{i=1,\ldots,p} TS_i$$

small is good

Goals of Study: Output Performance Measures (cont'd.)

- *Time-average number* of parts *in queue* (L_q),

$$\frac{\int_0^T Q(t)\,dt}{T}$$

$Q(t)$ = number of parts in queue at time t
T = *Simulation time*
$\int_0^T Q(t)\,dt$: area under the curve of $Q(t)$

L_q is a *time-persistent* statistic.

- *Maximum number* of parts *in queue* (L_q^*),

$$\max_{0 \le t \le T} Q(t)$$

worst case (indication of how much floor space is needed)
small is good

Goals of Study: Output Performance Measures (cont'd.)

- ***Utilization*** **of the machine or proportion of time busy (ρ),**

$$\frac{\int_0^T B(t)\,dt}{T} \qquad B(t) = \begin{cases} 1, \text{if the machine is busy at time } t \\ 0, \text{if the machine is idle at time } t \end{cases}$$

ρ is also a *time-persistent* statistic.

Very important in simulation: high (ρ=1) is good (low excess capacity), but maybe congested system

- **Many other measures are possible (too many → Information overload?)**

Analysis Options

- **The objective of simulation is to obtain the outputs by transforming the inputs according to the model's logic**
- **We can also use "educated guessing" or "queuing theory"**
- **Educated guessing**
  - Guessing calculations can sometimes provide qualitative insights.
  - Example: look at the average inflow and processing rates
    - Average interarrival time = 4.08 minutes
    - Average service time = 3.46 minutes
    - So (on average) parts are being processed faster than they arrive
      - System has a chance of operating in a stable way in long run, i.e., might not "explode"
      - If all interarrivals and service times were exactly at their mean, there would never be a queue
      - But data clearly exhibit variability, so a queue could form
    - If we'd had average interarrival < average service time, and this persisted, then queue would explode
    - Truth — between these extremes: for instance, if during a period of time, the interarrival times are short and the same time the service times are long→ queue
  - Guessing has its limits …

Analysis Options (cont'd.)

- **Queueing theory**
  - Requires additional assumptions about model
  - Popular, simple model: *M/M/1 queue*
    - Interarrival times ~ exponential
    - Service times ~ exponential, independent of interarrivals
    - Must have E(service time) < E(interarrival time)
    - Steady-state (long-run, forever)
    - Exact analytic results; e.g., average waiting time in queue is

$$W_q = \frac{\lambda}{\mu \cdot (\mu - \lambda)}, \qquad \lambda = \mathrm{E}(\text{arrival rate}) \qquad \mu = \mathrm{E}(\text{service rate})$$

See other formulas on slide #45

  - Problems: ***validity*** (Markovian times: not necessarily the case), ***estimating means*** (λ and μ), ***time frame*** (formula for the long run and not the simulation time), the formula does not provide any information about the ***variability in the system***
  - Often useful as first-cut approximation and allow to guess which type of simulation is appropriate

M/M/1: First 'M'- states that arrival process is *Markovian*, i.e., the interarrival times are independent and identically distributed (iid) draws from an exponential distribution. Second 'M' stands for service time distribution and here it is also exponential. '1' indicates there is a single server.

Mechanistic Simulation

- **The issues with “educated guessing” and “queuing theory” lead us to “simulation” again**
- **“*Mechanistic*” → individual operations (arrivals, service by the drill press) happen as they would in reality → imitation of reality**
- **Movements, changes in the simulation model occur at right “times,” in right order and have the right effect on each other**
- **Therefore, the different pieces interact**
- **Install “observers” (statistical-accumulator variables) to get output performance measures**
- **Concrete, “brute-force” analysis approach for dealing with the model**
- **Nothing mysterious or subtle**
  - But a lot of details, bookkeeping
  - Simulation software keeps track of things for you

Pieces of a Simulation Model

- ***Entities***
  - "**Players**" that move around, change status, affect and are affected by other entities (an object of interest in the system)
  - *Dynamic objects* — get created, move around, leave/are disposed of (maybe)
  - Usually represent "real" (physical) things
    - **In Our model: entities are parts → created when they arrive, wait in the queue (if necessary), are served by the drill press, and are then disposed of as they leave.**
  - Can have "fake" (or "logic") entities for modeling "tricks"
    - Breakdown demon (to model a machine failure: an entity that holds the server for a while), break angel (similar idea)

      Arena has built-in ways to model these examples directly
  - Usually have multiple "copies" or *realizations* (of the entity) floating around
  - Can have different types of entities concurrently (different processing or priority)
  - Usually, identifying types of entities is first thing to do in building a model

Pieces of a Simulation Model (cont'd.)

- ***Attributes***
  - Characteristic (property) of all entities: describe, differentiate
  - All entities have same attribute "slots" but different values for different entities, for example:
    - Time of arrival
    - Due date
    - Priority
    - Color
  - Attribute value tied to a specific entity
  - Similar to a tag attached to each entity, but what is written on the tag differs across entities
  - Like "local" (to entities) variables
  - Some automatic in Arena, some you define

- ***Activities***
  - What entities do over a specified length of time period.

Pieces of a Simulation Model (cont'd.)

- **(Global or State)** ***Variables***
  - Reflects a characteristic of whole model, not of specific entities
  - Used for many different kinds of things
    - Travel time between all station pairs
    - Number of parts in system
    - Simulation clock (built-in Arena variable)
  - Not tied to entities (in contrast to attributes)
  - Entities can access, change variables
  - If attributes are tags attached to entities floating in a room, think of variables as writings on the walls of the room (that are rewriteable)
  - Some built-in by Arena (number in the queue, number of busy servers,…), you can define others (mean service time, travel time, …)
  - Useful in practice: for example, if the time to move between any two stations in the model is constant, define a variable "***Transfer Time***" and set it to the appropriate value and use it whenever necessary. Its value can be changed if necessary.
  - A variable's value can also change during the simulation.
  - Can be scalars, vectors and matrices
- ***Event***
  - Is defined as an instantaneous occurrence that might change the state (variable) of the system
    - Arrival: a new part enters the system
    - Departure: a part finishes its service and leaves the system
    - The End: the simulation is stopped (at 20 min in the example)

Pieces of a Simulation Model (cont'd.)

- ***Resources***
  - What entities compete for
    - People
    - Equipment
    - Space
  - Entity *seizes* a resource, uses it, *releases* it
  - Think of a *resource being assigned to an entity*, rather than an entity "belonging to" a resource (because an entity may need different resources at the same time: a machine and a worker)
  - "A" resource can have several *units* of capacity
    - Seats at a table in a restaurant
    - Identical ticketing (parallel) agents at an airline counter
  - Number of units of resource can be changed during simulation (agents going on break for instance)

Pieces of a Simulation Model (cont'd.)

- ***Queues***
  - Place for entities to wait when they can't move on (maybe since the resource they want to seize is not available)
  - Have names, often tied to a corresponding resource
  - Can have a finite capacity to model limited space — You have to define as part of your simulation model what to do if an entity shows up to a queue that's already full
  - Usually watch length of a queue, waiting time in it

Pieces of a Simulation Model (cont'd.)

- ***Statistical accumulators***
  - In order to get output performance measures, we need to keep track of different intermediate ***statistical-accumulator*** variables, as the simulation progresses.
  - Variables that "watch" what's happening
  - Depend on output performance measures desired
  - "Passive" in the model — do not participate in the simulation, but rather just watch
  - Most are automatic in Arena, but some you may have to set up and maintain during simulation
  - At end of simulation, they are used to compute final output performance measures
  - Examples:
    - The number of parts produced so far
    - The total of the waiting times in the queues so far
    - The highest level Q(t) has reached so far
  - All these accumulators should be initialized to 0.
  - When an event happens in the system, the affected accumulators should be updated.

Pieces of a Simulation Model (cont'd.)

- **Statistical accumulators for simple processing system**
  - Number of parts produced so far (P)
  - Total of waiting times spent in queue so far ($\sum_{i=1}^{N} WQ_i$)
  - No. of parts that have gone through queue (N)
  - Max time in queue we've seen so far (W_q^*)
  - Total of times spent in system by all parts so far ($\sum_{i=1}^{P} TS_i$)
  - Max time in the system we've seen so far (W_s^*)
  - Area so far under queue-length curve $Q(t)$-- $\int Q$
  - Max of $Q(t)$ so far-- L_q^*
  - Area so far under server-busy curve $B(t)$--$\int B$

Pieces of a Simulation Model (cont'd.)

- ***Example of components of a system***

Table 1 Examples of Systems and Their Components

System	*Entities*	*Attributes*	*Activities*	*Events*	*State Variables*
Banking	Customers	Checking-account balance	Making deposits	Arrival; departure	Number of busy tellers; number of customers waiting
Rapid rail	Riders	Origin; destination	Traveling	Arrival at station; arrival at destination	Number of riders waiting at each station; number of riders in transit
Production	Machines	Speed; capacity; breakdown rate	Welding; stamping	Breakdown	Status of machines (busy, idle, or down)
Communications	Messages	Length; destination	Transmitting	Arrival at destination	Number waiting to be transmitted
Inventory	Warehouse	Capacity	Withdrawing	Demand	Levels of inventory; backlogged demands

Source: Carson

Simulation Dynamics: Event-Scheduling “World View”

- **When using a simulation package, a modeler adopts a world view for developing a model to advance the simulation and guarantee that event happen in the correct chronological order: “event-scheduling”.**
- **Identify characteristic *events*: for instance an entity that leaves the queue and start the service because of another entity’s departure (which is already an event)**
- **Decide on a *logic* for each type of events to:**
  - Effect *state changes* for each event type
  - Observe simulation statistics
  - Update times of future events (maybe of this type, other types)
- **Keep a simulation *clock*, future *event calendar* (entity involved, type of event, event time). The next (soonest) event is always at the top of the calendar.**
- ***Jump* from one event to the next (no need to waste (real) time looking at (simulated times) that don’t matter), process, observe statistics, update event calendar**
- **Must specify an appropriate *stopping rule* (20 min, or as soon as 100 finished parts are produced)**
- **Usually done with general-purpose programming language (C++, Java, Matlab, FORTRAN, etc.)**

Events for the Simple Processing System

- ***Arrival*** **of a new part to system**
  - Schedule the next part to arrive later, at the next arrival time, by placing a new event record in the event calendar
  - Update time-persistent statistical accumulators (from last event to now)
    - Area under $Q(t)$
    - Max of $Q(t)$
    - Area under $B(t)$
  - "Mark" arriving part with current time (use later) as an attribute: store the arriving part's time of arrival (current time)
  - If machine is idle:
    - Start processing (schedule departure), Make machine busy, Tally waiting time in queue (0)
  - Else (machine is busy):
    - Put part at end of queue, increase queue-length variable

Events for the Simple Processing System (cont'd.)

- ***Departure*** **(when a service is completed)**
  - Increment number-produced stat accumulator
  - Compute & tally time in system (now – time of arrival)
  - Update time-persistent statistics (as in arrival event)
  - If queue is non-empty:
    - Take first part out of queue, compute & tally its waiting time in queue, begin service (schedule departure event)
  - Else (queue is empty):
    - Make machine idle (Note: there will be no departure event scheduled on future events calendar, which is as desired)

Events for the Simple Processing System (cont'd.)

- ***The End***
  - Update time-persistent statistics (to end of simulation)
  - Compute final output performance measures using current (= final) values of statistical accumulators

- **After each event (other than *The End*), event calendar's top record is removed to see what to do (what event will happen next), and what time it is (the time of next event)**
- **The simulation clock is advance to that time**
- **The appropriate logic is carried out**

- **Also must initialize everything**

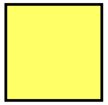
Some Additional Specifics for the Simple Processing System

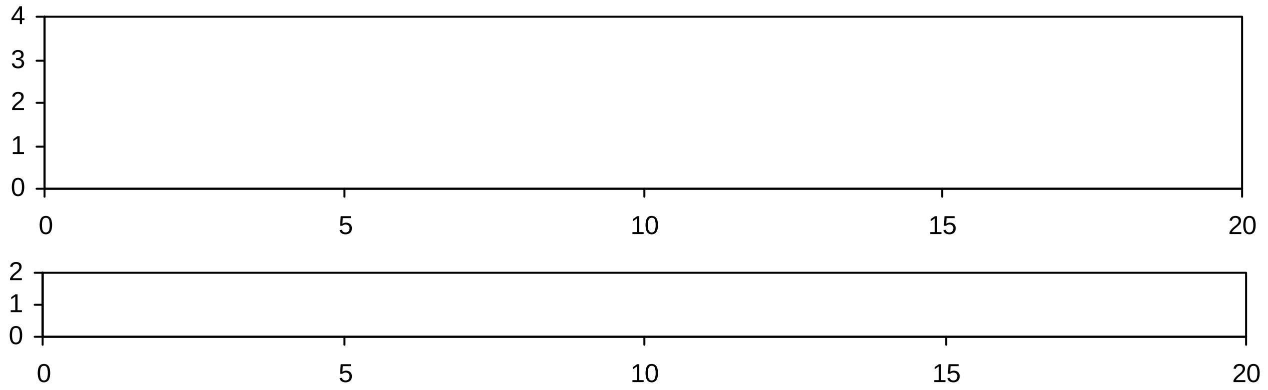
- **Simulation clock variable (internal in Arena)**
- **Event calendar: list of event *records*:**
  - [Entity No., Event Time, Event Type]
  - Keep *ranked* in increasing order on Event Time
  - Next event always in top record
  - Initially, schedule first Arrival, The End (Dep.?)
- **State variables: describe current status**
  - Server status $B(t)$ = 1 for busy, 0 for idle
  - Number of customers in queue $Q(t)$
  - Times of arrival of each customer now in queue (a list of random length)

Simulation by Hand

- **Manually track state variables, statistical accumulators**
- **Use "given" interarrival, service times**
- **Keep track of event calendar**
- **Move clock from one event to next**
- **Will omit times in system, "max" computations here (see text for complete details)**

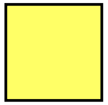
Simulation by Hand: Setup

<table>
<tr><td>System
</td><td>Clock</td><td>B(t)</td><td>Q(t)</td><td>Arrival times of custs. in queue</td><td>Event calendar</td></tr>
<tr><td>Number of completed waiting times in queue</td><td colspan="2">Total of waiting times in queue</td><td colspan="2">Area under Q(t)</td><td>Area under B(t)</td></tr>
<tr><td>Q(t) graph

B(t) graph</td><td colspan="5">

Time (Minutes)</td></tr>
<tr><td>Interarrival times</td><td colspan="5">1.73, 1.35, 0.71, 0.62, 14.28, 0.70, 15.52, 3.15, 1.76, 1.00, ...</td></tr>
<tr><td>Service times</td><td colspan="5">2.90, 1.76, 3.39, 4.52, 4.46, 4.36, 2.07, 3.36, 2.37, 5.38, ...</td></tr>
</table>

Simulation by Hand:
t = 0.00, Initialize

<table>
<tr><td>System
</td><td>Clock

0.00</td><td>B(t)

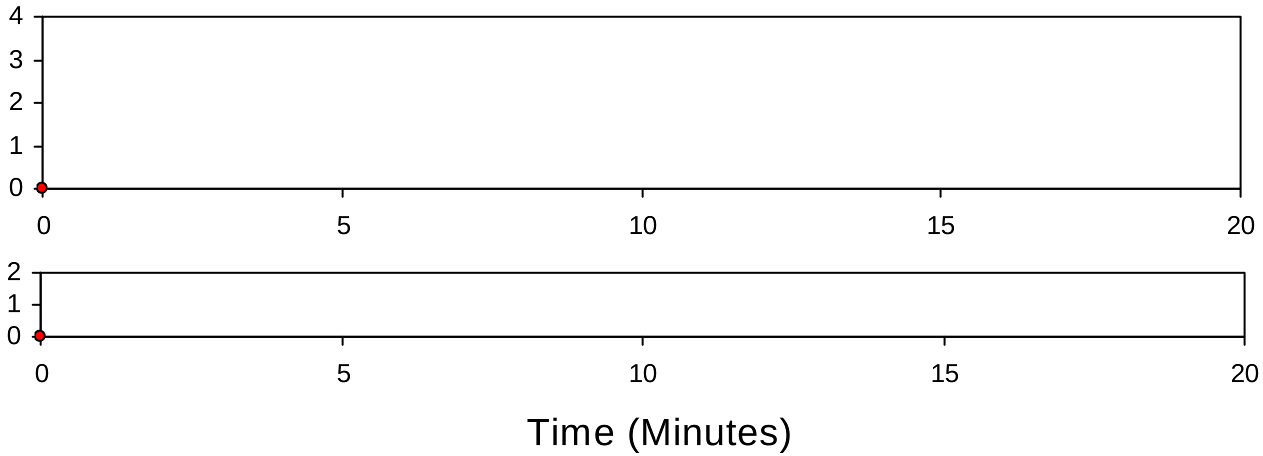
0</td><td>Q(t)

0</td><td>Arrival times of custs. in queue
<empty></td><td>Event calendar
[1, 0.00, Arr]
[–, 20.00, End]</td></tr>
<tr><td>Number of completed waiting times in queue
0</td><td colspan="2">Total of waiting times in queue

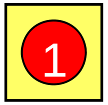
0.00</td><td colspan="2">Area under Q(t)

0.00</td><td>Area under B(t)

0.00</td></tr>
<tr><td>Q(t) graph

B(t) graph</td><td colspan="5"></td></tr>
<tr><td>Interarrival times</td><td colspan="5">1.73, 1.35, 0.71, 0.62, 14.28, 0.70, 15.52, 3.15, 1.76, 1.00, ...</td></tr>
<tr><td>Service times</td><td colspan="5">2.90, 1.76, 3.39, 4.52, 4.46, 4.36, 2.07, 3.36, 2.37, 5.38, ...</td></tr>
</table>

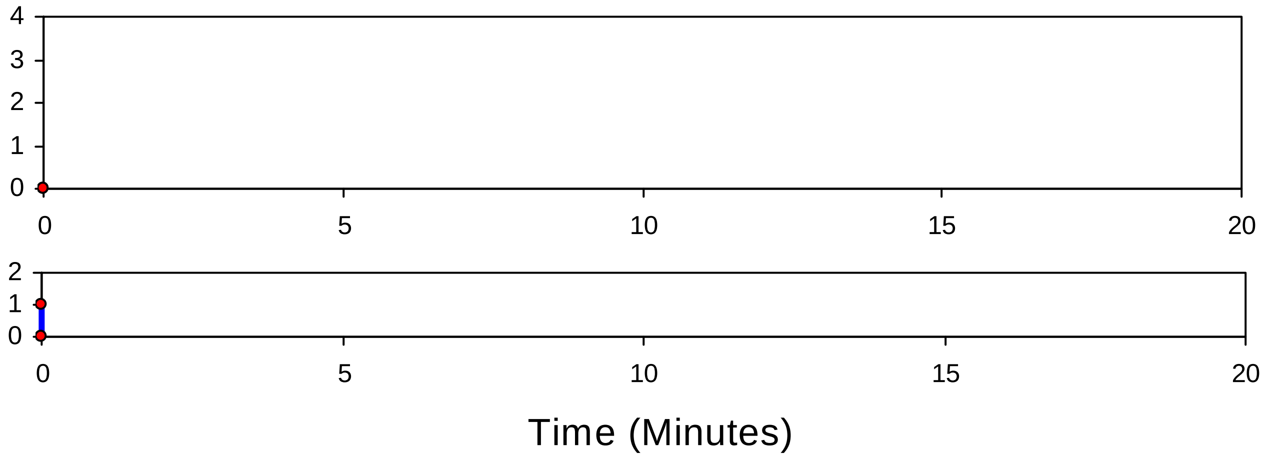
Simulation by Hand:
t = 0.00, Arrival of Part 1

System	Clock	$B(t)$	$Q(t)$	Arrival times of custs. in queue	Event calendar
	0.00	1	0	<empty>	[2, 1.73, Arr] [1, 2.90, Dep] [–, 20.00, End]

Number of completed waiting times in queue	Total of waiting times in queue	Area under $Q(t)$	Area under $B(t)$
1	0.00	0.00	0.00

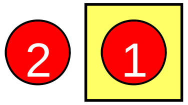
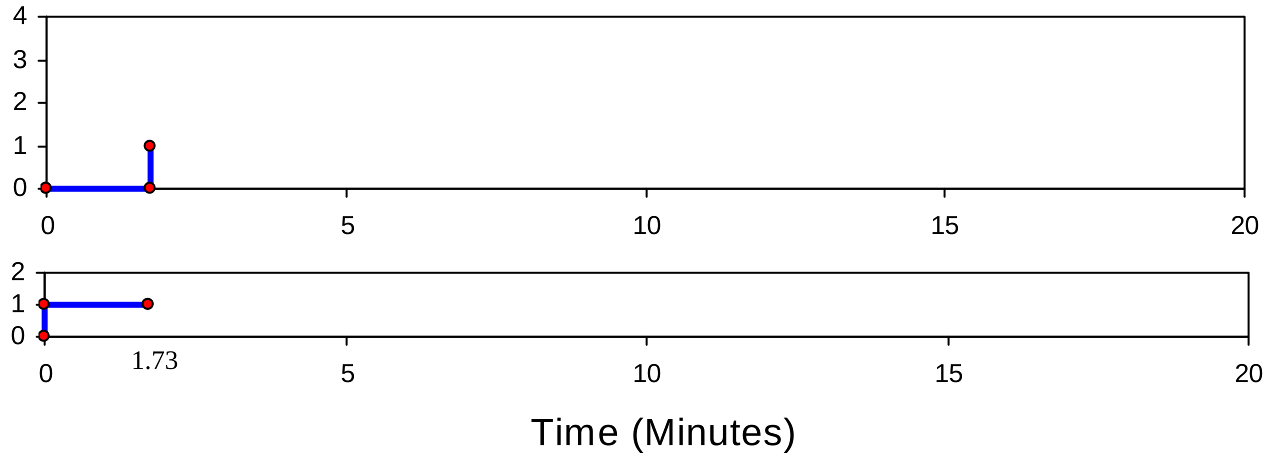
$Q(t)$ graph

$B(t)$ graph



Interarrival times	~~1.73~~, 1.35, 0.71, 0.62, 14.28, 0.70, 15.52, 3.15, 1.76, 1.00, ...
Service times	~~2.90~~, 1.76, 3.39, 4.52, 4.46, 4.36, 2.07, 3.36, 2.37, 5.38, ...

Simulation by Hand: t = 1.73, Arrival of Part 2

System	Clock	$B(t)$	$Q(t)$	Arrival times of custs. in queue	Event calendar
	1.73	1	1	(1.73)	[1, 2.90, Dep] [3, 3.08, Arr] [–, 20.00, End]
Number of completed waiting times in queue 1	Total of waiting times in queue 0.00		Area under $Q(t)$ 0.00		Area under $B(t)$ 1.73
$Q(t)$ graph $B(t)$ graph					
Interarrival times	~~1.73~~, ~~1.35~~, 0.71, 0.62, 14.28, 0.70, 15.52, 3.15, 1.76, 1.00, ...				
Service times	~~2.90~~, 1.76, 3.39, 4.52, 4.46, 4.36, 2.07, 3.36, 2.37, 5.38, ...				

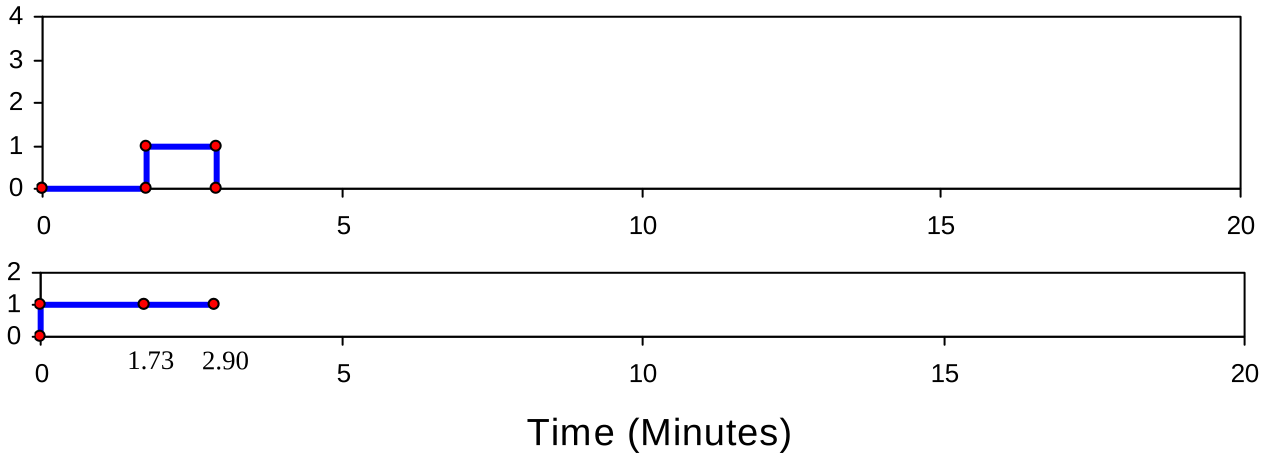
Simulation by Hand:
t = 2.90, Departure of Part 1

System	Clock	$B(t)$	$Q(t)$	Arrival times of custs. in queue	Event calendar
	2.90	1	0	<empty>	[3, 3.08, Arr] [2, 4.66, Dep] [–, 20.00, End]

Number of completed waiting times in queue	Total of waiting times in queue	Area under $Q(t)$	Area under $B(t)$
2	1.17	1.17	2.90

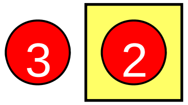
Q(*t*) graph

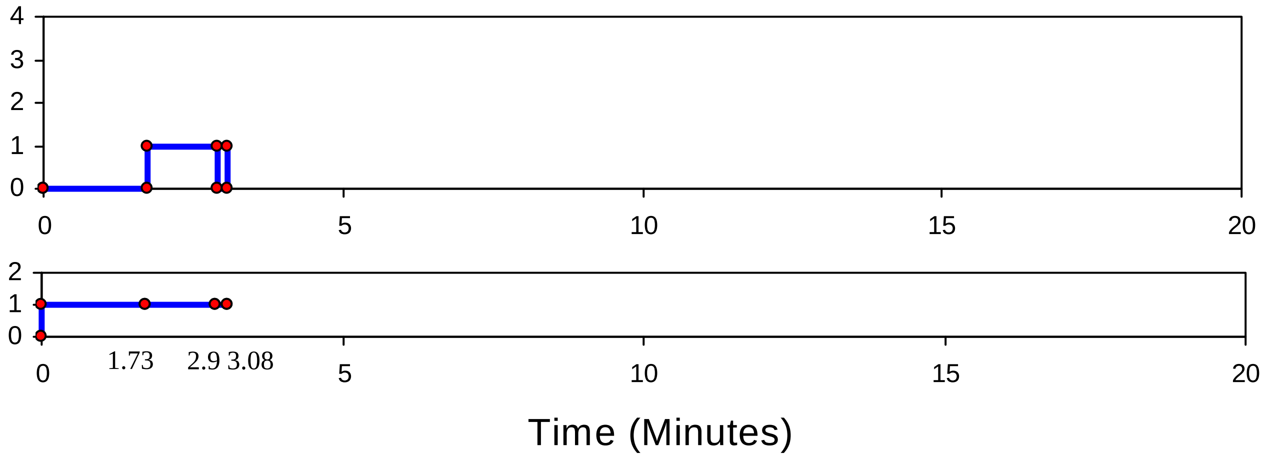
B(*t*) graph



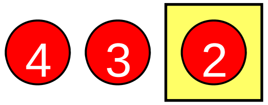
Interarrival times	~~1.73~~, ~~1.35~~, 0.71, 0.62, 14.28, 0.70, 15.52, 3.15, 1.76, 1.00, ...
Service times	~~2.90~~, ~~1.76~~, 3.39, 4.52, 4.46, 4.36, 2.07, 3.36, 2.37, 5.38, ...

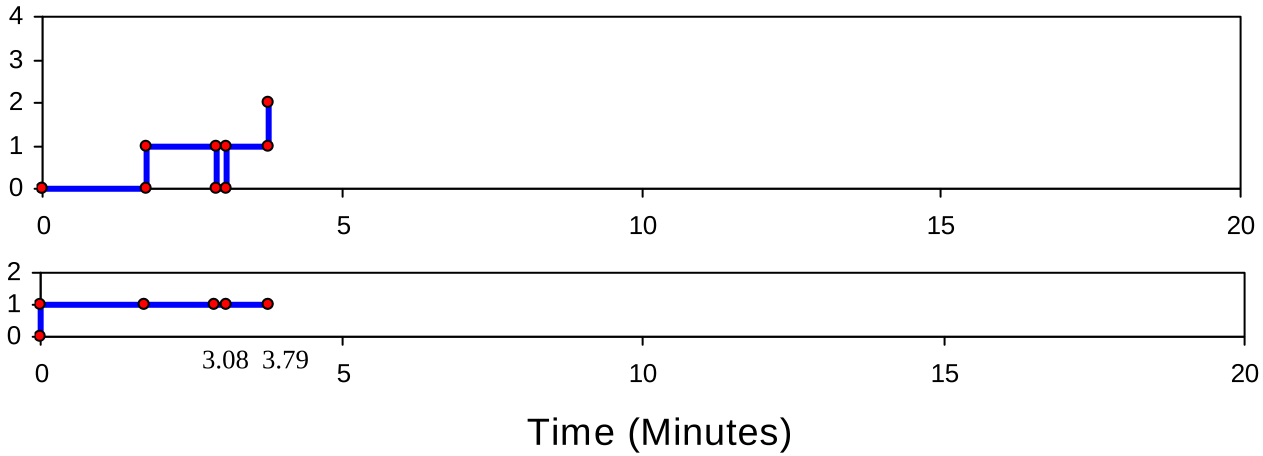
Simulation by Hand:
t = 3.08, Arrival of Part 3

<table>
<tr><td>System
</td><td>Clock
3.08</td><td>B(t)
1</td><td>Q(t)
1</td><td>Arrival times of custs. in queue
(3.08)</td><td>Event calendar
[4, 3.79, Arr]
[2, 4.66, Dep]
[–, 20.00, End]</td></tr>
<tr><td>Number of completed waiting times in queue
2</td><td colspan="2">Total of waiting times in queue
1.17</td><td colspan="2">Area under Q(t)
1.17</td><td>Area under B(t)
3.08</td></tr>
<tr><td>Q(t) graph

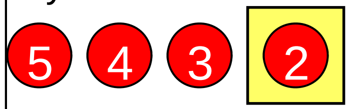
B(t) graph</td><td colspan="5"></td></tr>
<tr><td>Interarrival times</td><td colspan="5">~~1.73~~, ~~1.35~~, ~~0.71~~, 0.62, 14.28, 0.70, 15.52, 3.15, 1.76, 1.00, ...</td></tr>
<tr><td>Service times</td><td colspan="5">~~2.90~~, ~~1.76~~, 3.39, 4.52, 4.46, 4.36, 2.07, 3.36, 2.37, 5.38, ...</td></tr>
</table>

Simulation by Hand:
t = 3.79, Arrival of Part 4

<table>
<tr><td>System
</td><td>Clock
3.79</td><td>B(t)
1</td><td>Q(t)
2</td><td colspan="2">Arrival times of custs. in queue
(3.79, 3.08)</td><td>Event calendar
[5, 4.41, Arr]
[2, 4.66, Dep]
[–, 20.00, End]</td></tr>
<tr><td>Number of completed waiting times in queue
2</td><td colspan="3">Total of waiting times in queue
1.17</td><td colspan="2">Area under Q(t)
1.88</td><td>Area under B(t)
3.79</td></tr>
<tr><td>Q(t) graph

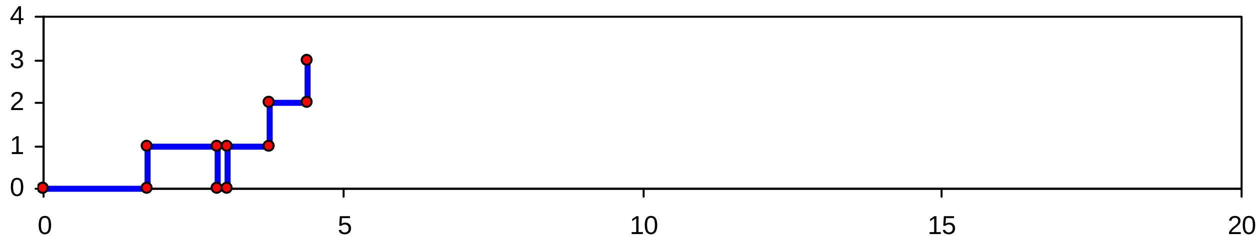
B(t) graph</td><td colspan="6"></td></tr>
<tr><td>Interarrival times</td><td colspan="6">~~1.73~~, ~~1.35~~, ~~0.71~~, ~~0.62~~, 14.28, 0.70, 15.52, 3.15, 1.76, 1.00, ...</td></tr>
<tr><td>Service times</td><td colspan="6">~~2.90~~, ~~1.76~~, 3.39, 4.52, 4.46, 4.36, 2.07, 3.36, 2.37, 5.38, ...</td></tr>
</table>

Simulation by Hand:
t = 4.41, Arrival of Part 5

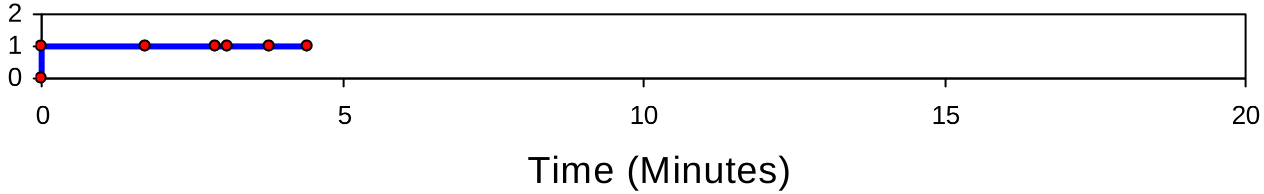
System	Clock	*B*(*t*)	*Q*(*t*)	Arrival times of custs. in queue	Event calendar
	4.41	1	3	(4.41, 3.79, 3.08)	[2, 4.66, Dep] [6, 18.69, Arr] [–, 20.00, End]

Number of completed waiting times in queue	Total of waiting times in queue	Area under *Q*(*t*)	Area under *B*(*t*)
2	1.17	3.12	4.41

Q(*t*) graph

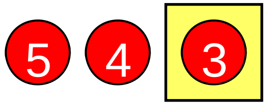


B(*t*) graph

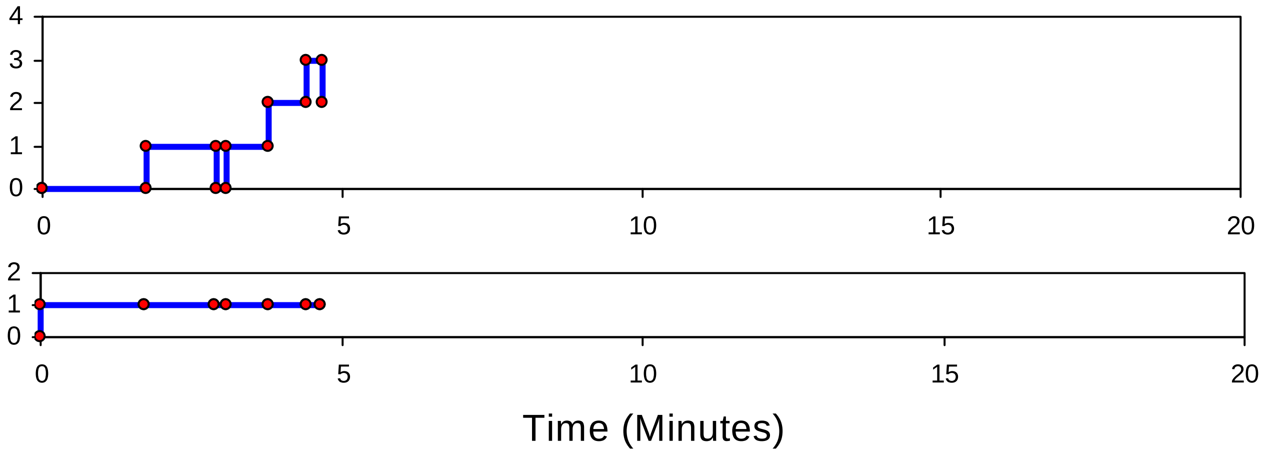


Interarrival times	~~1.73~~, ~~1.35~~, ~~0.71~~, ~~0.62~~, ~~14.28~~, 0.70, 15.52, 3.15, 1.76, 1.00, ...
Service times	~~2.90~~, ~~1.76~~, 3.39, 4.52, 4.46, 4.36, 2.07, 3.36, 2.37, 5.38, ...

Simulation by Hand:
t = 4.66, Departure of Part 2

System	Clock	$B(t)$	$Q(t)$	Arrival times of custs. in queue	Event calendar
	4.66	1	2	(4.41, 3.79)	[3, 8.05, Dep] [6, 18.69, Arr] [–, 20.00, End]

Number of completed waiting times in queue	Total of waiting times in queue	Area under $Q(t)$	Area under $B(t)$
3	2.75	3.87	4.66

$Q(t)$ graph $B(t)$ graph	
Interarrival times	~~1.73~~, ~~1.35~~, ~~0.71~~, ~~0.62~~, ~~14.28~~, 0.70, 15.52, 3.15, 1.76, 1.00, ...
Service times	~~2.90~~, ~~1.76~~, ~~3.39~~, 4.52, 4.46, 4.36, 2.07, 3.36, 2.37, 5.38, ...

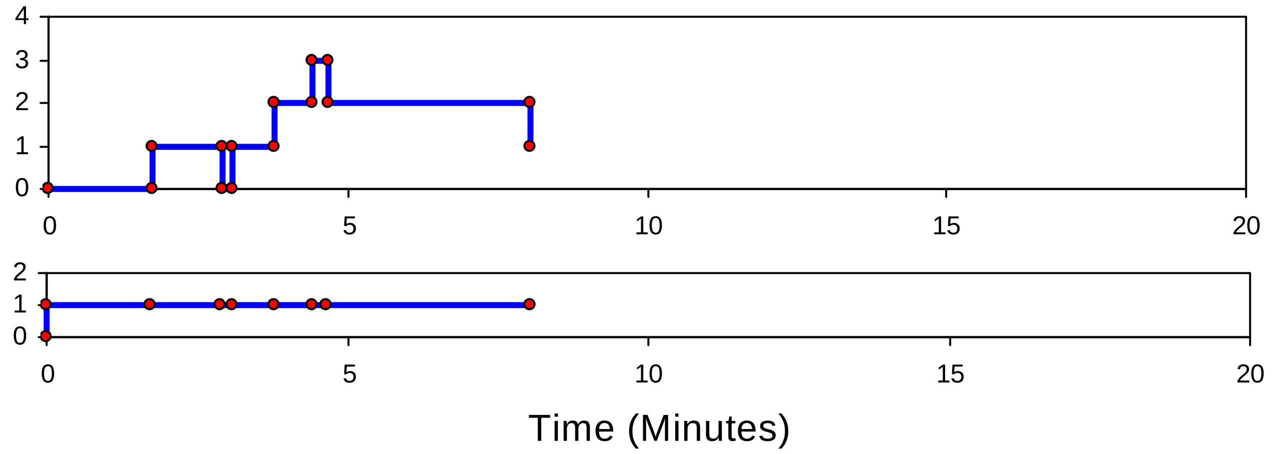
Simulation by Hand:
t = 8.05, Departure of Part 3

System	Clock	$B(t)$	$Q(t)$	Arrival times of custs. in queue	Event calendar
5 [4]	8.05	1	1	(4.41)	[4, 12.57, Dep] [6, 18.69, Arr] [–, 20.00, End]

Number of completed waiting times in queue	Total of waiting times in queue	Area under $Q(t)$	Area under $B(t)$
4	7.01	10.65	8.05

$Q(t)$ graph

$B(t)$ graph



Interarrival times	~~1.73~~, ~~1.35~~, ~~0.71~~, ~~0.62~~, ~~14.28~~, 0.70, 15.52, 3.15, 1.76, 1.00, ...
Service times	~~2.90~~, ~~1.76~~, ~~3.39~~, ~~4.52~~, 4.46, 4.36, 2.07, 3.36, 2.37, 5.38, ...

Simulation by Hand:
t = 12.57, Departure of Part 4

<table>
<tr><td>System

</td><td>Clock

12.57</td><td>B(t)

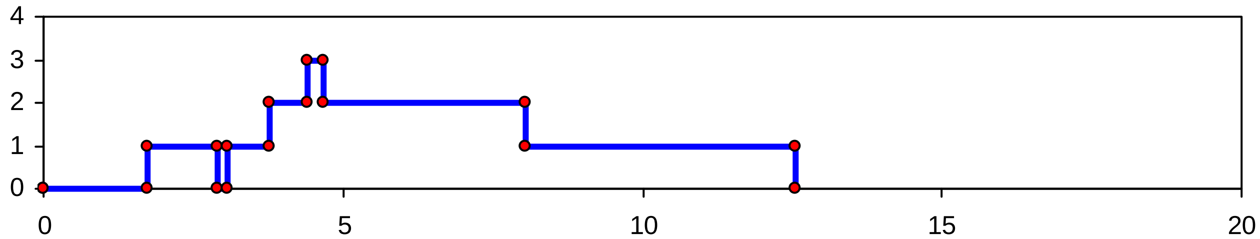
1</td><td>Q(t)

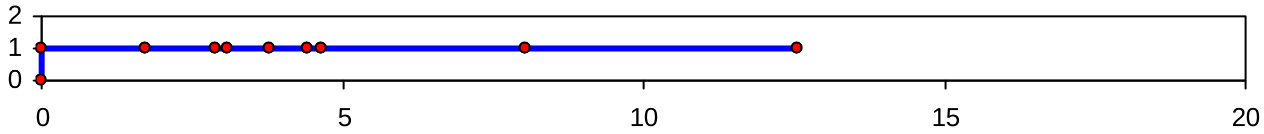
0</td><td>Arrival times of custs. in queue
()</td><td>Event calendar
[5, 17.03, Dep]
[6, 18.69, Arr]
[–, 20.00, End]</td></tr>
<tr><td>Number of completed waiting times in queue
5</td><td colspan="2">Total of waiting times in queue

15.17</td><td colspan="2">Area under Q(t)

15.17</td><td>Area under B(t)

12.57</td></tr>
<tr><td>Q(t) graph

B(t) graph</td><td colspan="5">



Time (Minutes)</td></tr>
<tr><td>Interarrival times</td><td colspan="5">~~1.73~~, ~~1.35~~, ~~0.71~~, ~~0.62~~, ~~14.28~~, 0.70, 15.52, 3.15, 1.76, 1.00, ...</td></tr>
<tr><td>Service times</td><td colspan="5">~~2.90~~, ~~1.76~~, ~~3.39~~, ~~4.52~~, ~~4.46~~, 4.36, 2.07, 3.36, 2.37, 5.38, ...</td></tr>
</table>

Simulation by Hand:
t = 17.03, Departure of Part 5

<table>
<tr><td>System</td><td>Clock

17.03</td><td>B(t)

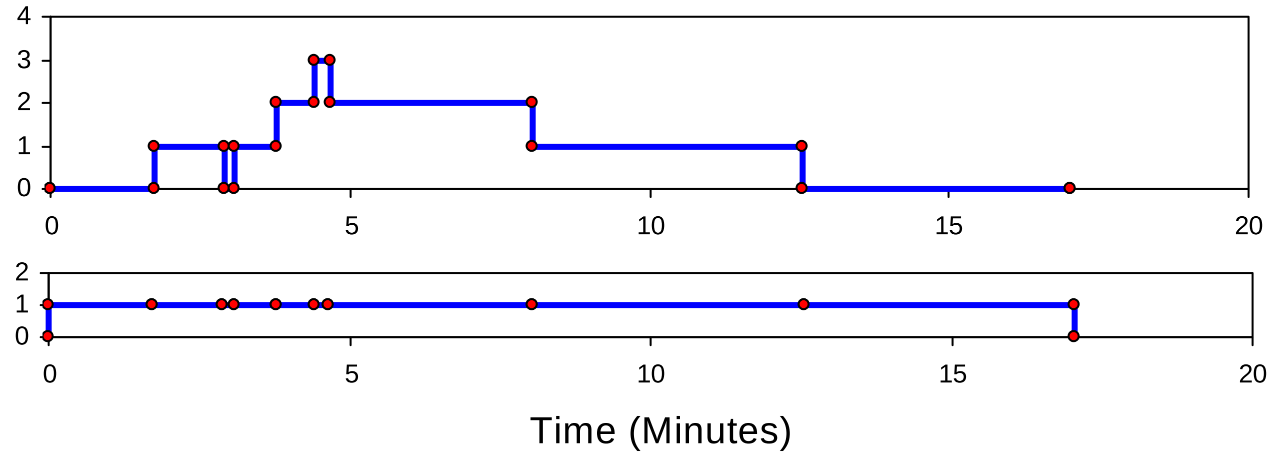
0</td><td colspan="2">Q(t)

0</td><td>Arrival times of custs. in queue
()</td><td>Event calendar
[6, 18.69, Arr]
[–, 20.00, End]</td></tr>
<tr><td>Number of completed waiting times in queue
5</td><td colspan="3">Total of waiting times in queue

15.17</td><td colspan="2">Area under Q(t)

15.17</td><td>Area under B(t)

17.03</td></tr>
<tr><td>Q(t) graph

B(t) graph</td><td colspan="6">
</td></tr>
<tr><td>Interarrival times</td><td colspan="6">~~1.73~~, ~~1.35~~, ~~0.71~~, ~~0.62~~, ~~14.28~~, 0.70, 15.52, 3.15, 1.76, 1.00, ...</td></tr>
<tr><td>Service times</td><td colspan="6">~~2.90~~, ~~1.76~~, ~~3.39~~, ~~4.52~~, ~~4.46~~, 4.36, 2.07, 3.36, 2.37, 5.38, ...</td></tr>
</table>

Simulation by Hand:
t = 18.69, Arrival of Part 6

<table>
<tr><td>System
6</td><td>Clock

18.69</td><td>$B(t)$

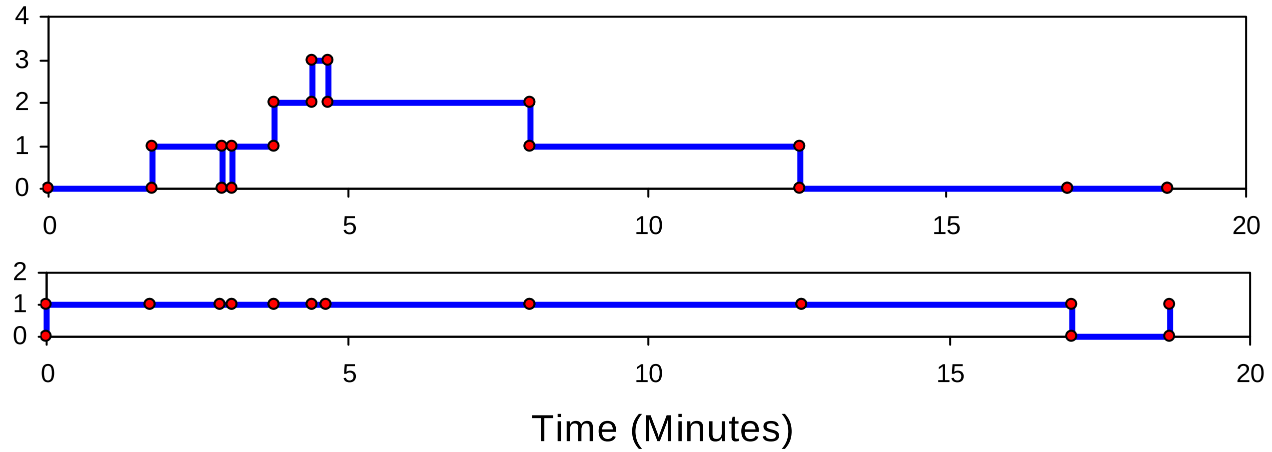
1</td><td>$Q(t)$

0</td><td colspan="2">Arrival times of custs. in queue
()</td><td>Event calendar
[7, 19.39, Arr]
[–, 20.00, End]
[6, 23.05, Dep]</td></tr>
<tr><td>Number of completed waiting times in queue
6</td><td colspan="3">Total of waiting times in queue

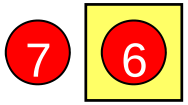
15.17</td><td colspan="2">Area under $Q(t)$

15.17</td><td>Area under $B(t)$

17.03</td></tr>
<tr><td>$Q(t)$ graph

$B(t)$ graph</td><td colspan="6">
</td></tr>
<tr><td>Interarrival times</td><td colspan="6">~~1.73~~, ~~1.35~~, ~~0.71~~, ~~0.62~~, ~~14.28~~, ~~0.70~~, 15.52, 3.15, 1.76, 1.00, ...</td></tr>
<tr><td>Service times</td><td colspan="6">~~2.90~~, ~~1.76~~, ~~3.39~~, ~~4.52~~, ~~4.46~~, ~~4.36~~, 2.07, 3.36, 2.37, 5.38, ...</td></tr>
</table>

Simulation by Hand: *t* = 19.39, Arrival of Part 7

<table>
<tr><td colspan="2">System
</td><td>Clock

19.39</td><td>B(t)

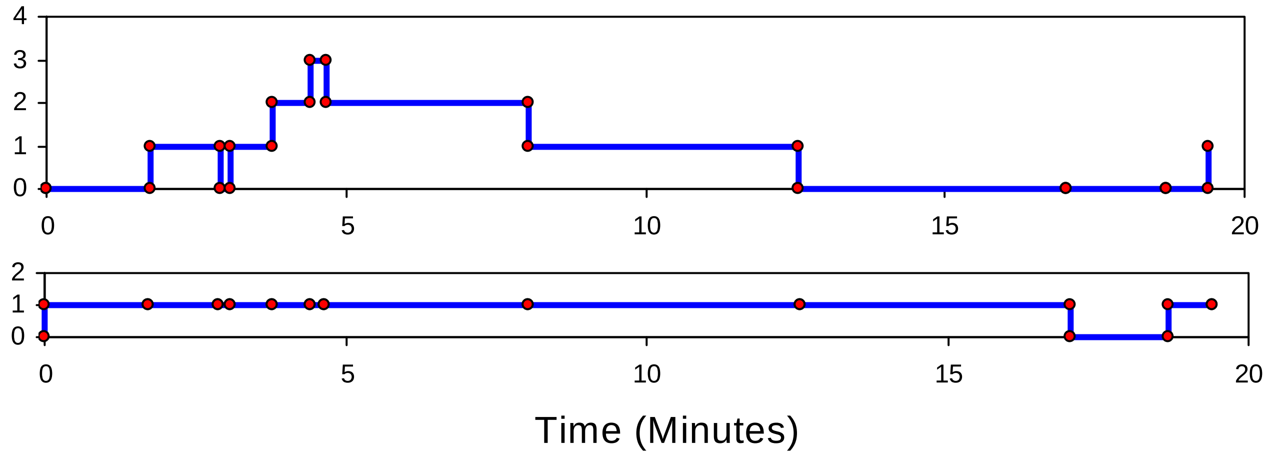
1</td><td>Q(t)

1</td><td>Arrival times of custs. in queue
(19.39)</td><td>Event calendar
[–, 20.00, End]
[6, 23.05, Dep]
[8, 34.91, Arr]</td></tr>
<tr><td>Number of completed waiting times in queue
6</td><td colspan="3">Total of waiting times in queue

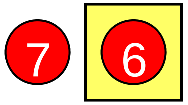
15.17</td><td colspan="2">Area under Q(t)

15.17</td><td>Area under B(t)

17.73</td></tr>
<tr><td>Q(t) graph

B(t) graph</td><td colspan="6"></td></tr>
<tr><td>Interarrival times</td><td colspan="6"><s>1.73</s>, <s>1.35</s>, <s>0.71</s>, <s>0.62</s>, <s>14.28</s>, <s>0.70</s>, <s>15.52</s>, 3.15, 1.76, 1.00, ...</td></tr>
<tr><td>Service times</td><td colspan="6"><s>2.90</s>, <s>1.76</s>, <s>3.39</s>, <s>4.52</s>, <s>4.46</s>, <s>4.36</s>, 2.07, 3.36, 2.37, 5.38, ...</td></tr>
</table>

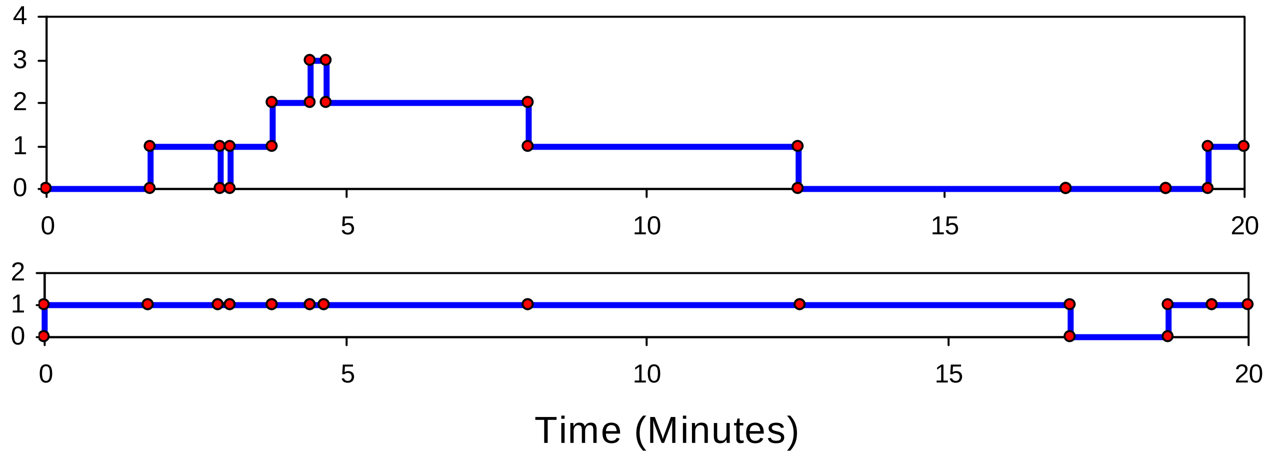
Simulation by Hand: *t* = 20.00, The End

System	Clock	$B(t)$	$Q(t)$	Arrival times of custs. in queue	Event calendar
	20.00	1	1	(19.39)	[6, 23.05, Dep] [8, 34.91, Arr]

Number of completed waiting times in queue	Total of waiting times in queue	Area under $Q(t)$	Area under $B(t)$
6	15.17	15.78	18.34

$Q(t)$ graph

$B(t)$ graph



Interarrival times	~~1.73~~, ~~1.35~~, ~~0.71~~, ~~0.62~~, ~~14.28~~, ~~0.70~~, ~~15.52~~, 3.15, 1.76, 1.00, ...
Service times	~~2.90~~, ~~1.76~~, ~~3.39~~, ~~4.52~~, ~~4.46~~, ~~4.36~~, 2.07, 3.36, 2.37, 5.38, ...

Event-Scheduling Logic via Programming

- **Clearly well suited to standard programming language (C, C++, Java, etc.)**
- **Often use "utility" libraries for:**
  - List processing
  - Random-number generation
  - Random-variate generation
  - Statistics collection
  - Event-list and clock management
  - Summary and output
- **Main program ties it together, executes events in order**

Complete Record of the Hand Simulation (cont..)

Statistical accumulator:

P : the total number of parts produced so far

N : the number of entities that have passed through the queue

$\sum_{i=1}^{N} WQ_i$: the sum of the queue times that have been observe so far

W_q^*: the maximum time in queue observed so far

$\sum_{i=1}^{p} TS_i$: the sum of the total times in system that have been observed so far

W_S^* : the maximum total time in system observed so far

$\int Q$: area under the $Q(t)$ curve so far

L_q^* : max no of parts in the queue (max value of $Q(t)$ so far)

$\int B$: the area under the $B(t)$ curve so far

Complete Record of the Hand Simulation

Just-Finished Event			Variables		Attributes		Statistical Accumulators									Event Calendar
Entity No.	Time t	Event Type	$Q(t)$	$B(t)$	Arrival Times: (In Queue)	In Service	P	N	ΣWQ	WQ^*	ΣTS	TS^*	$\int Q$	Q^*	$\int B$	[Entity No., Time, Type]
–	0.00	Init	0	0	( )	–	0	0	0.00	0.00	0.00	0.00	0.00	0	0.00	[1, 0.00, Arr] [–, 20.00, End]
1	0.00	Arr	0	1	( )	0.00	0	1	0.00	0.00	0.00	0.00	0.00	0	0.00	[2, 1.73, Arr] [1, 2.90, Dep] [–, 20.00, End]
2	1.73	Arr	1	1	(1.73)	0.00	0	1	0.00	0.00	0.00	0.00	0.00	1	1.73	[1, 2.90, Dep] [3, 3.08, Arr] [–, 20.00, End]
1	2.90	Dep	0	1	( )	1.73	1	2	1.17	1.17	2.90	2.90	1.17	1	2.90	[3, 3.08, Arr] [2, 4.66, Dep] [–, 20.00, End]
3	3.08	Arr	1	1	(3.08)	1.73	1	2	1.17	1.17	2.90	2.90	1.17	1	3.08	[4, 3.79, Arr] [2, 4.66, Dep] [–, 20.00, End]
4	3.79	Arr	2	1	(3.79, 3.08)	1.73	1	2	1.17	1.17	2.90	2.90	1.88	2	3.79	[5, 4.41, Arr] [2, 4.66, Dep] [–, 20.00, End]
5	4.41	Arr	3	1	(4.41, 3.79, 3.08)	1.73	1	2	1.17	1.17	2.90	2.90	3.12	3	4.41	[2, 4.66, Dep] [6, 18.69, Arr] [–, 20.00, End]
2	4.66	Dep	2	1	(4.41, 3.79)	3.08	2	3	2.75	1.58	5.83	2.93	3.87	3	4.66	[3, 8.05, Dep] [6, 18.69, Arr] [–, 20.00, End]
3	8.05	Dep	1	1	(4.41)	3.79	3	4	7.01	4.26	10.80	4.97	10.65	3	8.05	[4, 12.57, Dep] [6, 18.69, Arr] [–, 20.00, End]
4	12.57	Dep	0	1	( )	4.41	4	5	15.17	8.16	19.58	8.78	15.17	3	12.57	[5, 17.03, Dep] [6, 18.69, Arr] [–, 20.00, End]
5	17.03	Dep	0	0	( )	–	5	5	15.17	8.16	32.20	12.62	15.17	3	17.03	[6, 18.69, Arr] [–, 20.00, End]
6	18.69	Arr	0	1	( )	18.69	5	6	15.17	8.16	32.20	12.62	15.17	3	17.03	[7, 19.39, Arr] [–, 20.00, End] [6, 23.05, Dep]
7	19.39	Arr	1	1	(19.39)	18.69	5	6	15.17	8.16	32.20	12.62	15.17	3	17.73	[–, 20.00, End] [6, 23.05, Dep] [8, 34.91, Arr]
–	20.00	End	1	1	(19.39)	18.69	5	6	15.17	8.16	32.20	12.62	15.78	3	18.34	[6, 23.05, Dep] [8, 34.91, Arr]

Simulation by Hand: Finishing Up

- Average waiting time in queue (W_q) = $\sum_{i=1}^{N} WQ_i / N = 15.17/6 = 2.53\,\textbf{min/part}$
- Average total time in system (W_s) = $\sum_{i=1}^{p} TS_i / P = 32.20/5 = \textbf{6.44 min/part}$
- Time-average number in queue (L_q) = $\int_0^T Q(t)dt / T = 15.78/20 = 0.79\ part$
- Server utilization (ρ) = $\int_0^{20} B(t)dt / 15 = 18.34/20 = 0.92\,\textbf{(dimensionless)}$

- Total production = P = 5 parts
- Simulation time = T = 20 min.
- Maximum waiting time in queue = W_q^* = 8.16 min
- Maximum total time in system = W_S^* = 12.62 min
- Maximum no. of parts in queue = L_q^* = 3 parts

Simulation by Hand:
Comparison with classical M/M/1 model (Exact analytical Sol)

- **Average interarrival time = 4.08 minutes**
- **Average service time = 3.46 minutes**
  - **Arrival rate, λ = ?**
  - **Service rate, μ = ?**
- **Average waiting time in system,** $W_s = \frac{1}{\mu - \lambda} = ?$
- **Average waiting time in queue,** $W_q = W_s - \frac{1}{\mu} = \frac{\lambda}{\mu \cdot (\mu - \lambda)} = ?$
- **Average number of parts in system,** $L_s = \frac{\rho}{1-\rho} = \frac{\lambda}{\mu - \lambda} = ?$
- **Average number of parts in queue,** $L_q = \frac{\rho^2}{1-\rho} = \frac{\lambda^2}{\mu \cdot (\mu - \lambda)} = ?$
- **Server utilization,** $\rho = \frac{\lambda}{\mu} = ?$

Compare with the results obtained from simulation model and comment on them.

Randomness in Simulation

- Above was just one "replication" — a sample of size one (not worth much)
- However, we may be interested in knowing more about a "typical" morning, and how results vary from day to day.
- Since inputs are random, the measure of perf. would also vary (random)
- However, random input model maybe real uncertainty. Therefore, instead of using random numerical inputs, like in the example, use probability distributions (from which observations are generated) → Input Analyzer
- Probab. Distribution → generate different observations → more realistic; more simulation than what we would have observed (real data)
- In the previous example: the random numbers were generated from an exponential dist. with a mean of 5 min (interarrival times) and a triangular distr. with a minimum of 1 min, maximum of 6 min and a mode of 3 min (service time)

Randomness in Simulation

- **Made a total of five *replications* (IID):**

Performance Measure	1	2	3	4	5	Sample Avg.	Sample Std. Dev.	95% Half Width
Total production	5	3	6	2	3	3.80	1.64	2.04
Average waiting time in queue	2.53	1.19	1.03	1.62	0.00	1.27	0.92	1.14
Maximum waiting time in queue	8.16	3.56	2.97	3.24	0.00	3.59*	2.93*	3.63*
Average total time in system	6.44	5.10	4.16	6.71	4.26	5.33	1.19	1.48
Maximum total time in system	12.62	6.63	6.27	7.71	4.96	7.64*	2.95*	3.67*
Time-average number of parts in queue	0.79	0.18	0.36	0.16	0.05	0.31	0.29	0.36
Maximum number of parts in queue	3	1	2	1	1	1.60*	0.89*	1.11*
Drill-press utilization	0.92	0.59	0.90	0.51	0.70	0.72	0.18	0.23

Substantial variability across replications

- **Confidence intervals on true μ (ex- 95% CI on total production)**

$$\bar{X} \pm t_{\alpha/2,n-1}\frac{s}{\sqrt{n}} = \bar{X} \pm t_{0.025,5-1}\frac{s}{\sqrt{n}}, \text{ or, } 3.80 \pm 2.776\frac{1.64}{\sqrt{5}}$$

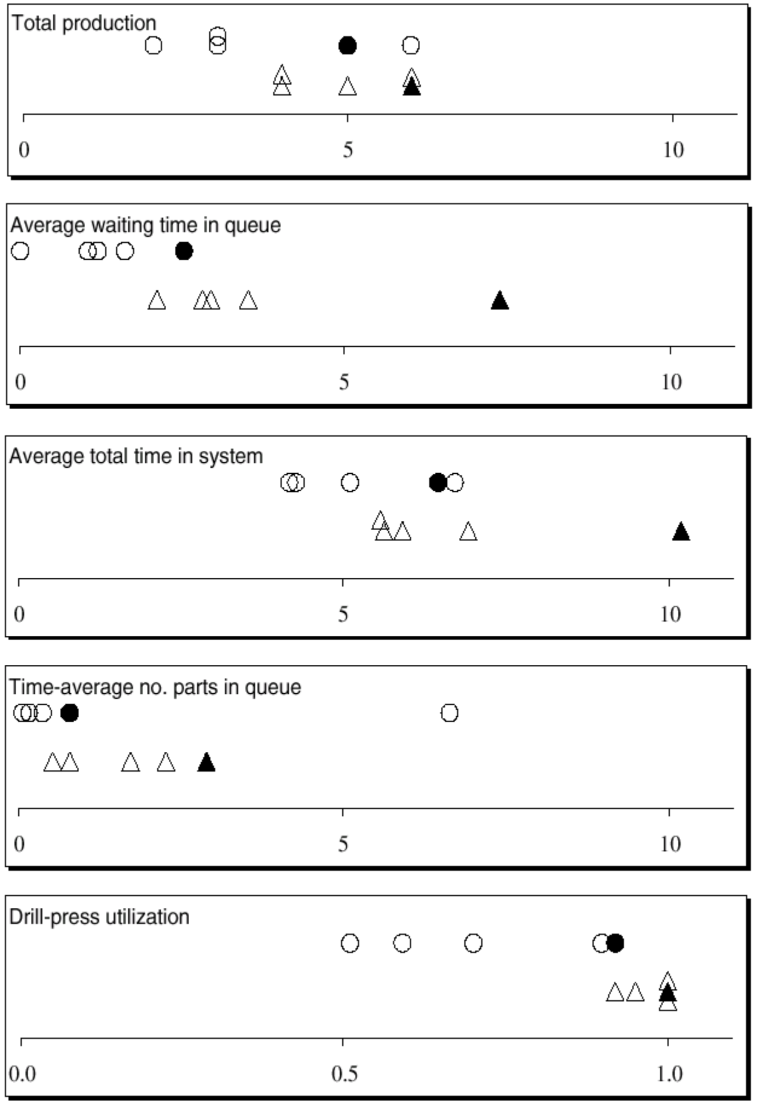
$$\text{or, } 3.80 \pm 2.04$$

Precision?

Comparing Alternatives

- **Usually, simulation is used for more than just a single model "configuration"**
- **Often want to compare alternatives, select or search for best (via some criterion)**
- **Simple processing system: What would happen if arrival rate doubled?**
  - Cut interarrival times in half
  - Rerun model for double-time arrivals
  - Make five replications

Results: Original vs. Double-Time Arrivals



- **Original – circles**
- **Double-time – triangles**
- **Replication 1 – filled in**
- **Replications 2-5 – hollow**
- **Note variability**
- **Danger of making decisions based on one (first) replication**
- **Hard to see if there are really differences**
- **Need: Statistical analysis of simulation output data**

Overview of a Simulation Study

- Understand system
- Be clear about goals
- Formulate model representation
- Translate into modeling software
- Verify “program”
- Validate model
- Design experiments
- Make runs
- Analyze, get insight, document results

More: Lecture 1

See Tutorial 1--Lecture 2

Continued in Lecture 3