MFS605/EE605
Systems for Factory Information and Control

Lecture 5
Fall 2005

Larry Holloway
Dept. of Electrical and Computer Engineering and
UK Center for Manufacturing

- **References:**
  - Modeling and Analysis of Mfg. Systems, Askin and Standridge (John Wiley, 1993)
  - Factory Physics, by Hopp and Spearman (McGraw Hill 1996)
  - Simulation Modeling and Analysis, 2nd ed. Law and Kelton, 1991 (McGraw Hill)
  - Probability and Random Processes for Electrical Engineering, 2nd edition, A. Leon-Garcia, 1994 (Addison-Wesley)
  - Nelson, "Stochastic Modeling, Analysis, and Simulation", 1995
  - Hillier & Lieberman, "Intro to Oper. Research, 6th edition", 1995

Other News

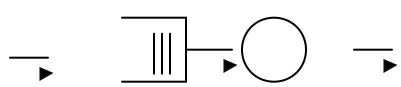
Quiz:

Next class… (tentative)

- **Needed: Student computer lab account**
  - **Sign up at www.uky.edu/scs**

Queuing Theory: Basic Terms

- **Queuing Theory: Study of lines, waiting in lines**



- **Buffer or Queue**
- **"Server" -- service rate =** m
- **"Customers" --- arrival rate =** l
- **Utilization is** r = l/m --- **required to be less than 1.**

Questions we can answer:

- **What is avg. number of "customers" in the queue?**
- **What is avg. number of "customers" in the system?**
- **What is avg. wait of "customers" in the system?**
- **What percentage of time does queue exceed some number?**

Summary of results for M/M/1 queues

M/M/1 queue (for r <1):

- **P_0 = 1 -** l/m = 1 - r
- **P_n = (**l/m**)^n (1-** l/m) = r^n(1-r) = r^n **P_0** **(for n>0)**
- **L: = mean number of customers *in the system***

$$L = E[n] = \sum_{n=0}^{\infty} nP_n = \frac{l}{(m-l)} = \frac{r}{1-r}$$

- **W:= mean wait time in the system**

$$W = \frac{L}{l} = \frac{1}{(m-l)} = \frac{1}{m(1-r)}$$

- **W_Q : mean waiting time in just the queue**

  W_Q = W - $1/m$

- **L_Q : mean number of customers in the queue**

  L_Q = $l\ W_Q$

Multiple servers: M/M/c

$$r = \frac{l}{cm} < 1$$

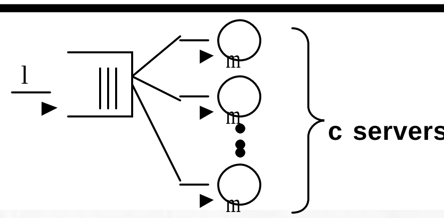


Table 11.1 *M/M/c* Queueing Results

	$M/M/1$	$M/M/c$
$p(0)$	$1-\rho$	$\left[\frac{(c\rho)^c}{c!(1-\rho)} + \sum_{n=0}^{c-1}\frac{(c\rho)^n}{n!}\right]^{-1}$
L_q	$\frac{\rho^2}{1-\rho}$	$\frac{\rho(c\rho)^c p(0)}{c!(1-\rho)^2}$
L	$\frac{\rho}{1-\rho}$	$L_q + \frac{\lambda}{\mu}$
W_q	$\frac{\rho}{\mu(1-\rho)}$	$\frac{(c\rho)^c p(0)}{c!c\mu(1-\rho)^2}$
W	$\frac{1}{\mu(1-\rho)}$	$W_q + \mu^{-1}$

Example: comparison of M/M/1 vs. M/M/2

Values of L for M/M/c model

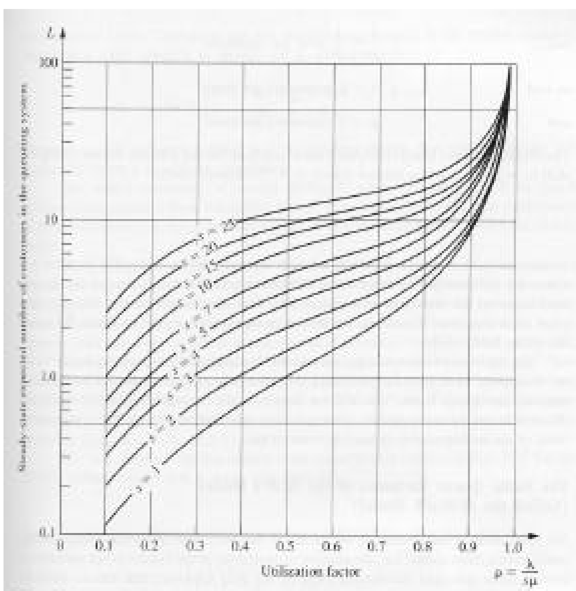


Figure 15.7 Values for L for the M/M/s model (Sec. 15.6).

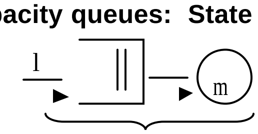
From Hillier and Lieberman, 6th edition

Effect of many small (many queue) vs. single machine

- **(example done last class ---- *not an M/M/2*)**

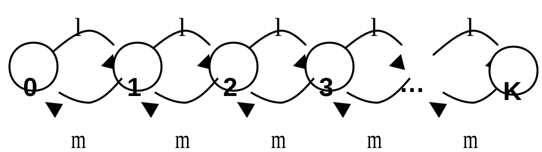
M/M/1/K

- **Finite capacity queues: State probabilities and # in system**



System capacity of K

- **Birth-death model**



$$P_i = r^i P_0$$

$$1 = \sum_{i=0}^{K} P_i = \sum_{i=0}^{K} r^i P_0 \longrightarrow P_0 = \frac{1-r}{1-r^{K+1}}$$

1 server -- Finite Capacity Queues: M/M/1/K

$$L = E[n] = \sum_{n=0}^{K} nP_n = \begin{cases} \dfrac{r}{1-r} - \dfrac{(K+1)r^{K+1}}{1-r^{K+1}} & for\ r \neq 1 \\ \dfrac{K}{2} & for\ \ r = 1 \end{cases}$$

- **M/M/1/K queues: mean wait in system:**
  - **Note some customers get turned away.**
  - **For Little's Law, we need *effective arrival rate***

$$P_i = r^i \left(\frac{1-r}{1-r^{K+1}} \right)$$

<u>offered load</u> is measure of demand on system: λ/μ

<u>carried load</u> is actual demand met by system: λ_a/μ

$$l_a = effective\ arrival\ rate = l\ (1 - P_K)$$

$$W = \frac{L}{l_a}$$

$$W_Q = W - \frac{1}{m}$$

$$L_Q = W_Q l_a \qquad note\ it\ can\ be\ shown\ this\ is\ L_Q = L - (1 - P_0)$$

G/G/1 Queues

- An approximation (From Kingman 1961, as given in Hopp and Spearman 1996) for the expected wait time in the queue in a G/G/1 system is given by the following:

$$W_{q(G/G/1)} = \left(\frac{c_a^2 + c_e^2}{2} \right) \left(\frac{r}{1-r} \right) \frac{1}{m}$$

where c_a is the coefficient of variation of the arrival time distribution and c_e is the coefficient of variation of the processing times distribution.

- Approx. is exact for M/M/1 and M/G/1
- Variability term, Utilization term, Capacity term
- → variability in arrival times or in processing time give congestion.
- → increasing variability increases congestion
- → increasing utilization increases congestion ("blow up")

Example for G/G/1 approx.

13

Summary

- **M/M/1, M/M/c, M/M/k represent “practical worst case” situations with “maximum randomness” (memoryless property).**
- **Use the results as bounds.**
  - **M/M/1 results are very simple**
  - **M/M/c results:**
    - Multiple servers with shared queues
- **Queues explode as utilization approaches 1**
- **G/G/1 results: reducing variability reduces congestion**
    - Variability term, utilization term, capability term

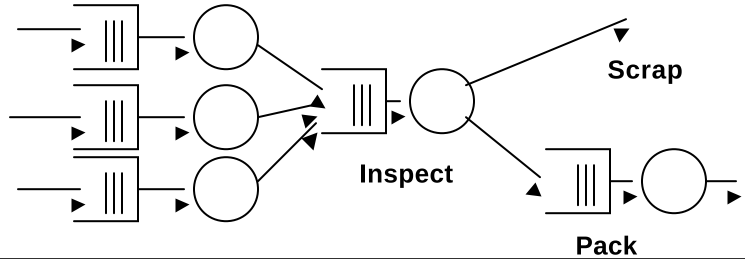
14

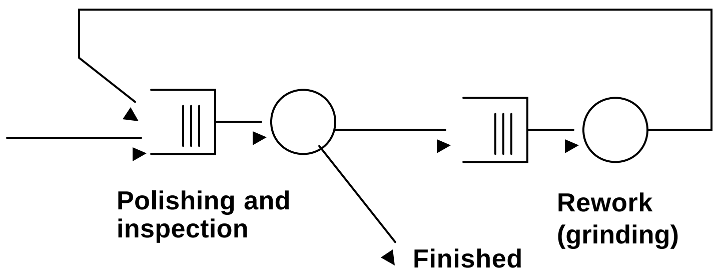
Systems with multiple queues

- Open networks vs. closed networks

- ***open Jackson networks:***
  - open network
  - Poisson external arrivals
  - exponential service times (possibly c>1) at each node
  - unlimited buffers at each node
  - probabilistic routing
  - First come – First Serve
  - one type of customer

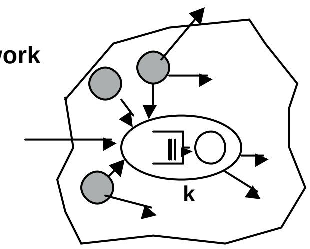
Examples





Open Jackson Networks

- **Why Jackson networks?**
  - **Probability of system state can be calculated from looking at the individual queues in the system.**
  - ***Under steady state conditions, each facility in a Jackson network behaves as if it were an <u>independent</u> M/M/c queuing system with effective arrival rate:***
    - $\lambda_k = a_k + \Sigma_{i=1...m} \lambda_i p_{ik}$
    - p_{ik} **probability of routing from i to k**
    - a_k **arrival rate in from outside**
    - λ_k **arrival rate at node k**
    - **m number of nodes in network**



Key facts leading to Jackson networks

- **Sum of independent Poisson random vars. is Poisson.**
- **Poisson arrival process means exponential distribution of inter-arrival times**
- **Inter-departure times from M/M/c system (infinite queue capacity) is exponential**

Jackson Network Solution Procedure

1. **Solve for arrival rates using**

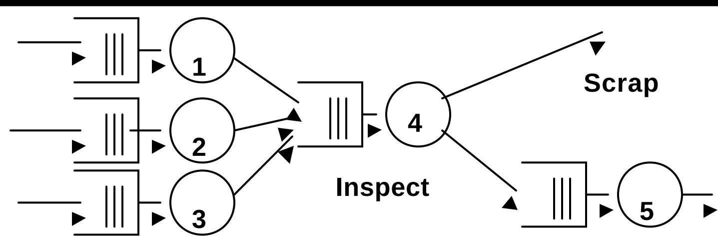
   $$\lambda_k = a_k + \Sigma_{i=1..m}\ \lambda_i\ p_{ik}$$

2. **Analyze each workstation as a M/M/1 or M/M/c queue**

   Use standard table of L, W, L_Q, W_Q formulas

3. **Combine results across workstations to get performance measures of the entire system.**

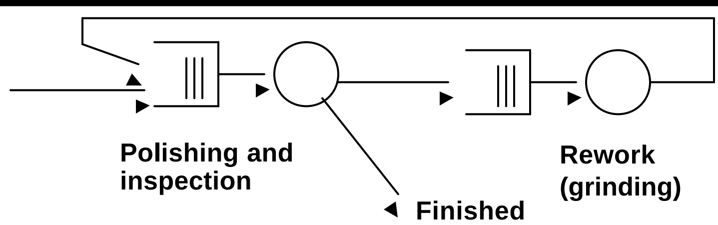
Queueing Network example 1



a_1= 20/day
a_2= 10/day
a_3= 10/day
$\mu_1 = \mu_2 = \mu_3$ = 25/day
μ_4 = 60/day
μ_5 = 45/day
Scrap $rate_4$ = 5%

Example 1 -- continued

Queueing Network example 2



a_1= 2 parts/hr.

μ_1 = 0.5 parts/hr.

μ_2 = 2 parts/hr.

Rejection rate = 1.%

Example 2: continued

Aggregate Parts

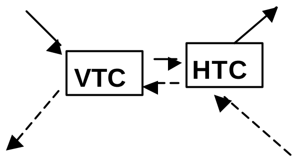
- **Jackson Networks assume one part type**
  - **What if we actually have more types?**
    - → create an aggregate part type

- **Aggregate arrival rates are sum of individual arrival rates**
  - **5 parts/hr for part A and 8 parts/hr for part B gives 13 parts/hr aggregate**

- **Average processing times found by averaging the *times* (*not averaging the rates!*)**
  - **Example: 20% are part A which are 5/hr, and 80% are part B which has rate 2 per hr.**
  - → **average time = 0.20(12 min) + 0.80(30min) = 26.4min.**

    so rate is (1/26.4min) = 2.27per hour.
- **(*this is <u>not</u> same as .2*5 + .8*2 = 2.6/hr.)***

Queuing network example 3



	Part A	***Part B***
Path	**VTC-HTC-exit**	**HTC-VTC-exit**
Demand	**Poisson 30/day**	**Poisson 60/day**
VTC time (min.)	**uniform(8,12)**	**uniform(8,12)**
HTC time (min.)	**uniform(10,14)**	**uniform(6,10)**

Assume 960 min/day (2 shifts@8hrs.)

Example 3 (cont.)

Jackson summary

- Network of systems with exponential service times and probabilistic routings can be decomposed into individual M/M/1 or M/M/c problems
- Requires single part type, Poisson external arrivals
- Relatively fast and simple

Extensions

Closed Queueing Networks

- Examples:
  - Constant work in process
  - Fixed number of pallets
- Solution method: Mean Value Analysis (MVA)
  - Iterative – computer based
  - P part different part types – each with total N_P in the system
  - Find steady-state solution relating throughput times, throughput rates, and queue lengths
  - Solution done by iterative search

Modeling Review

- **Queueing models: Considers uncertainty in steady state: single machine and Jackson networks.**
  - **How long is the average wait?**
  - **How many parts on average in the system?**
  - **Problem: easy answers require strong assumptions**
    - too much randomness?

Modeling Review

- **Analysis of unpaced serial lines *with processing time variation***
  - **No Buffers:**
    - Significant capacity lost due to processing time variation
  - **Infinite Buffers:**
    - Long-term: No capacity lost
  - **Finite Buffers:?**
    - Can use the coefficient of variation to characterize the amount of variation.
      - For exponential RV, cv = 1.
    - Portions of lost capacity can be recovered using finite buffers
      - For identical distributions for each machine, can recover:
        - 80% capacity recovery if buffer size = 10*cv
        - 90% capacity recovery if buffer size = 20*cv

Reductions in variation allows reduction in buffers (and thus throughput time and WIP)

Discrete-Event Simulation

Simulation: running model of system on a computer
Simulation allows specific questions and models

- **How is behavior during startup? During steadystate?**
- **What if we have a new inventory management policy?**
- **What if we have different kinds of parts circulating?**

- **Cheaper and faster than prototype**
- **Less restrictive than deterministic or queueing models**

- **Disadvantages:**
  - **more detailed model --> more model-building time**
  - **no analytical relationships --> "try this and see"**
  - **potential pitfalls (to cover later)**

Example Process

Terminology

- **Event: instantaneous occurrence that changes the state of the system**

- **State: Complete description of the system at a given time.**
  - **Example: buffer states, machine states, etc.**

- **Key points from example:**

  - ***State changes occur at discrete points in time.***
    ***These are* “events”.**

  - ***<u>Nothing</u> happens between events***

- ***Great idea of discrete event simulation:***
  - ***since nothing happens between events, then just jump between them!***

- **What we need:**
  - **simulation clock**
  - **system state and tracking variables**
  - **event list**
  - **statistical counters**

Example continued

- **At each event, we:**
  - **update the state variables**
  - **forecast (schedule) new events**
  - **delete events that are no longer relevant**

Computer “Random” numbers

- **Simulation requires “random” number sequences**
- **Computer is deterministic, so we actually have “psuedo-random” number sequence**
  - **Start with a “random” “seed”**
  - **Perform math function on it to get next random number.**
  - **Use it as your next seed and repeat**
  - **Issues:**
    - reuse of same seed gives same simulation
    - some seeds may wrap around and appear very non-random
    - what is distribution of the resulting sequence?

Example of random number generation

Mapping to different distributions

- **Once we have a psuedo-random number generator with uniform distribution, we can map this onto any other distribution (even with experimentally derived distributions)**
- **Inverse Transform method:**

Classes of Simulations

- **General Purpose languages**

- **Simulation packages**

- **Generic model/GUI methods**

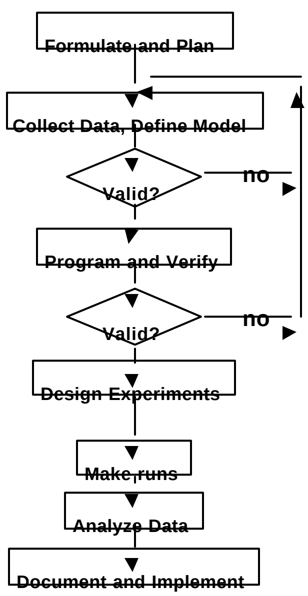
Review

- **Discrete Event Simulation**
  - **run computer model of system instead of prototype**
  - **"Discrete-Event" because state changes at event occurrences, not every time step**
  - **Key idea: since events occur only infrequently, we can skip clock forward in time to next event**
    - maintain event list
    - maintain state variables
    - maintain clock
    - etc.

Classes of simulation tools

- general purpose computer languages (write it yourself in C, etc.)
- simulation languages (SLAM, SIMAN, GPSS, …)
- graphical simulation modeling tools (Promodel, Arena, …)

Steps of Successful Simulation Study



- Formulate the problem and plan the study
- Collect data and model

Issues in Modeling

- How detailed?
- What is the scope of the model?
- What assumptions will be made?
- What random variables to be modeled?

How complex a model?

- ***Start simple, then add complexity only as needed***
- Complex models:
  - harder to understand, debug, validate, modify, document, explain, etc.
  - may be less accurate due to difficulty to debug and find errors
- Smaller models allow more replications and longer runs
- Judge simplifications in terms of impact on our performance measure: "will they increase or decrease it"?

Common Modeling Problems

- Selection of inappropriate distributions

Common Modeling Problems (cont.)

- Removing Randomness by using mean instead of distribution

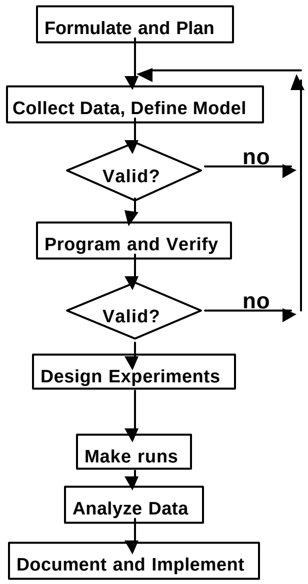
How do we choose distributions?

- Use collected data directly as input to the simulation
  - Problems:
- Use data to define empirical distribution
- Fit a theoretical distribution form to the data
  - may smooth out irregularities in an empirical model
  - able to generate extremes that may not be in original samples
  - sometimes physical reason to assume distribution

Example: dangers in modeling

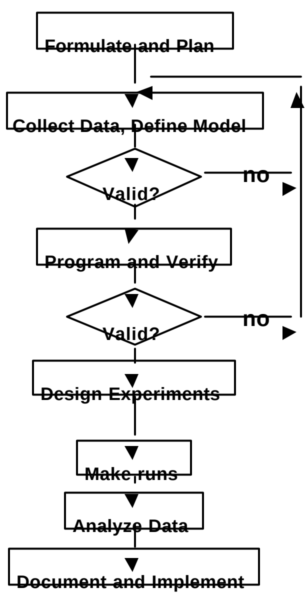
- Single machine, broken 10% of time
  - Model 1: machine breaks on avg of 540 min., then down for 60 minutes
  - Model 2: machine breaks down avg. of 54 minutes, then down for avg. of 6 min.
  - Model 3: Production rate is just decreased by 10% (thus machine breaks down constantly)

Steps of Successful Simulation Study



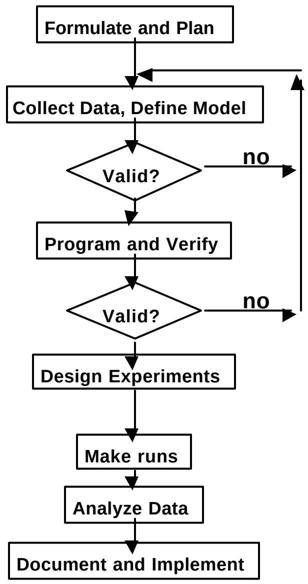
- **Validating:**
  - **Does the model appropriately represent reality??**
- **Verifying:**
  - **Is the program faithful to the model??**

Steps of Successful Simulation Study



- **Program and Verify**
  - ***Programming should only be 30-40% of time for study!***

Steps of Successful Simulation Study



- **Design Experiments**
  - **How long a run?**
  - **Initial Conditions?**
  - **# of runs?**
- **Keys:**
  - **Don't consider a single run valid**
  - **Don't neglect warmup period**