Simulation

Example cases

Examples

- Following cases will be elaborated within the course (to varying degree of detail)
  - A car wash station
    - Complete toy example
  - A logistic/delivery network
    - Draft of more realistic set ups
  - Surgery unit
    - Hands on case to work on

Car wash (1/8)

- System consists of
  - Stream of potential clients
  - Queuing place of limited capacity
  - Car wash machine
- Goal
  - Compare the profit for two machines with different capacity

Car wash (2/8)

- Profit/time unit
  - P= aU –b
  - b = fixed cost/time unit
  - a = profit/time unit during active utilization
  - U = utilization rate
  - a and b are known or can be estimated
  - U is to be simulated

Car wash (3/8)

- We know
    - Behavior of potential clients (distribution of arrival times)
    - Maximal allowable length of the queue
    - Distribution of the service times
- We want
  - Utilization rate U =T_busy/T_total
    - Or 1 – T_idle/T_total
  - Or the difference of rates for different models

Car wash (4/8)

- One variable can describe the state
  - N(t)= number of clients at time t
- Two types of events effect the state
    - Arrival: new client (i) arrives at time t= t_a(i)
    - Departure: client (j) leaves at time t= t_d(j)
- If N=0, system is empty (machine not used). = > Simulation has to provide times when N=0 (or N>0).

Car wash (5/8)

- If N(0), t_a:s and t_d:s are known, N(t) is uniquely determined and computable.
  - Simulation is needed to determine the arrival and (in particular) departure times.
  - Four variables + some counters
    - AT, DT (next arrival/departure time)
    - N (number of clients in the system)
    - t (current time)
    - E, T_idle (counters for collecting the idle time)

Car wash (6/8)

- Set the duration of simulation (T), maximal queue length M. Initialize time t=0, counters (T0=0, E=0), Set N=0 (empty system), DT=maxint
- AT= t+ ”arrival time”
- Repeat until t>T
  - If AT<DT play event ”arrival”, else play event ”departure”
- Report the results

Car wash (7/8)

- ”arrival”
  - t=AT;
  - If N<= M, N=N+1;
  - If N=1
    - DT=t+ ”service time”
    - T0=T0+t-E;
  - AT=t+”arrival time”;

- ”departure”
  - t=DT;
  - N=N-1;
  - If N>0
    - DT=t+ ”service time”
  - Else
    - DT=maxint;
    - E=t;

Car wash (8/8)

- ”Brute force” approach for a simple case.
- Hard to generalize to more complex situations
  - More event types, more complex state, need to follow the clients
- ”Everything” is selfmade
  - Date collection, book keeping of events and system state etc.

More complex examples

- Examples with several components and their interactions
  - Supply/delivery chain with loading/unloading operations and transport delays
  - Hierachical systems
  - Chain of critical services that may block the flow in upstream direction

Harbor network

- Consider traffic in a network of several harbors
- Assume average traffic between the harbors as known and given
- Harbors have different properties (number and capacity of loading docks)
- Ships have various fixed routes

Container harbor

- What can be simulated/varied
  - Utilization rates, waiting times
  - Needed number of ships, durations of routes (average and variability) (impact on crew scheduling)
  - Effects of different routing strategies
  - Effects of different queuing strategies
  - Etc

Container harbor

- Structural components of the model
  - Harbors
  - Docks
  - Ships
  - Containers?
  - Anything else?
- Which components have to be identified

Container harbor

- Events and interactions
  - Ship S arrives to harbor H
  - Loading/unloading of S begins at dock D
  - Loading/unloading of S ends at D
  - Ship S leaves for next harbor H(S)
  - (Ship S is created to the system)
  - (Ship S exits the system)

Container harbour

- Simplified situation (ship arrives to a dock and is unloaded)
- Most simple client-service –model
  - Create a ship and put it to a queue
  - Take a ship from a queue to the dock and start the loading (of known number of containers)
  - Unloading ends, dock becomes free, ship is removed

Container harbor

- Big harbor has several docks
- If docks are similar, no need to model them individually
- Create a pool of empty docks
  - In the beginning all docks in empty-pool
  - On ship's arrival pick a dock from the pool
  - Reinsert the dock to the pool on ship departure

Container harbor

- Details of unloading may be relevant to model
  - Think of unloading using a pool of container carriers
  - Allocation of carriers to different docks/ships can be varied
  - Submodel for unloading phase needed

Container harbor

- Ships have routes.
  - Create a ship, associate a route and place ship to starting position and state (loaded/empty/etc)
  - Ship starts journey to next harbor
  - Ship arrives, gets unloaded and moves forward
  - At the end ot the route the ship exits (or starts the next round)

Surgery unit

- Typical surgery involves three stages
  - (supervised) preparation of the patient (anesthetics etc)
  - Actual operation theatre
  - (supervised) recovery
- For each stage separate facilities are needed
  - How to plan the capacity for each stage

Surgery unit

- Main modelling challenge is the capacity bottlenecks
  - If the next stage is fully booked the patient can not move forward
  - Operation theatre can not be freed without capacity in the recovery
  - Correct modelling is needed