PRODUCTION SCHEDULING PROJECT

Hospital Emergency Room Scheduling

Xiaoyu Guo  xg2164
Adrian Luk  acl2150
Project Description... never been studied in class

- Simulated from limited data (4 Sundays) gathered at St. Luke’s Hospital ER
  - Determined the distributions for each process to represent the situation.
  - Defined scheduling problem and proposed solutions
  - Conducted computational analysis and analyzed results in MATLAB

- Processes in sequence:
  - Triage – Patients see a nurse first
  - Treatment – After receiving treatment by a doctor, patients leave the ER
    - Nurse/Doctor’s processing times depend on the criticality of each patient

- Facilities:
  - 2 treatment rooms in the ER
  - 1 nurse
  - 2 doctors

- Patients (jobs):
  - Arrive according to Poisson distribution / Exponential arrival rate
  - 5 criticalities: (5-Critical; 4-High; 3-Medium; 2-Low; 1-Clear/non-critical)
  - Death metrics: determined by release time and a uniformly distributed random variable with expectation that’s associated with criticality
Model Assumptions

- Patients arrive according to exponential distribution with lambda = 0.2
  - tA = -log(rand)/lambda
- Each type of patient has a determined chance to arrive
  - 10% cleared patient; 20% low criticality
  - 20% medium criticality; 30% high criticality; 20% critical
- Processing times:
  - Nurse processing time is random with expected duration of 10 minutes for all patients
  - Doctor processing time depending on criticality of patients
    - Uniformly distributed within doctor processing time bracket, i.e. critical patients has processing time between 75 -95 minutes.
- Death Metrics:
  - Releasing time + Processing time with doctor + Scalar + 10* Rand ()
  - * Rand () has expected values which vary with criticality of each type of patient and range is between +/-5
Scheduling Problem Setup - $J_2 | R_j | \sum U_j$

- Modeled after a Job Shop model
  - 5 Job types
    - Job 1 – criticality 0 (cleared) – 10%
    - Job 2 – criticality 1 (low) – 20%
    - Job 3 – criticality 3 (Medium) – 20%
    - Job 4 – criticality 4 (High) – 30%
    - Job 5 – criticality 5 (Critical) – 20%

- 2 Machines:
  - triage – Machine 1
  - treatment – Machine 2

- Death Time Metric:
  - 50 % chance
  - 100% chance

<table>
<thead>
<tr>
<th>Job</th>
<th>Infinity</th>
<th>Infinity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job 2</td>
<td>$R_j + A_{90} + P_{22}$</td>
<td>$R_j + A_{180} + P_{22}$</td>
</tr>
<tr>
<td>Job 3</td>
<td>$R_j + A_{36} + P_{23}$</td>
<td>$R_j + A_{180} + P_{22}$</td>
</tr>
<tr>
<td>Job 4</td>
<td>$R_j + A_{18} + P_{24}$</td>
<td>$R_j + A_{36} + P_{24}$</td>
</tr>
<tr>
<td>Job 5</td>
<td>$R_j + A_{9} + P_{25}$</td>
<td>$R_j + A_{18} + P_{25}$</td>
</tr>
</tbody>
</table>

$R_j$: Patients arrival time
$U_j$: Number of dead patients from the system
$A_9$: random variable constant with expected value of 9, ranging from +/- 5.
Proposed Algorithms

- **First In First Out:**
  - Prioritize the patients with earliest arrival time
  - Process earliest arrived patient when machine becomes available
  - Same for both nurse queue and doctor queue (sorted on arrival time)

- **Highest Criticality First / EDD:**
  - Assign priority to the patients with the highest criticality
  - Higher criticality is associated with earliest time in death metrics, both for 50% and 100%

<table>
<thead>
<tr>
<th>Job</th>
<th>85 + 10*rand()</th>
<th>175 + 10*rand()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job 2</td>
<td>31 + 10*rand()</td>
<td>67 + 10*rand()</td>
</tr>
<tr>
<td>Job 3</td>
<td>13 + 10*rand()</td>
<td>31 + 10*rand()</td>
</tr>
<tr>
<td>Job 4</td>
<td>4 + 10*rand()</td>
<td>23 + 10*rand()</td>
</tr>
</tbody>
</table>

- **Advanced Expected Death EDD:**
  - Comparing completion time with death metrics for patients in queue
  - Discard the patients with 100% certainty to die regardless received treatment or not, then advance the process by taking the next patient in queue
  - Feasible Lower Bound
FIFO Model & Output Analysis

- Store patients in queues for doctor and nurse
- Sort doctor queue based on the arrival time
  - \([Y,I] = \text{sort(queueD}(1,:), \text{'descend'})\)
    - \% queueD stores patients waiting to be treated by doctor
  - Check if the patient is dead before service starts
    - if (serviceD1(3,Ncustomers+1)>queueD(4,1))
      - serviceD1(2,NCustomers+1) = 0.001;
      - serviceD1(3,NCustomers+1) = serviceD1(1,NCustomers+1) + serviceD1(2,NCustomers+1);
      - NDeath = NDeath+1;
    - end
- Take the next on sorted queue when additional machine becomes available

Result: (20 exits in each simulation; 1000 simulations)

<table>
<thead>
<tr>
<th>Number of 100% Death</th>
<th>Number of 50% Death</th>
<th>Total Number Completed</th>
<th>Make Span (minutes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.08</td>
<td>2.12</td>
<td>20</td>
<td>299.55</td>
</tr>
</tbody>
</table>
Highest Criticality First/EDD

Similar set up as the FIFO except when queueD >0 & doctor or nurse becomes available, use a different sorting schema

```matlab
if (NextDoctorAvailable == 1) % if doctor 1 is the next available doctor
    EDDQueueD = EDDQueueD - 1;
    [Y,I] = sort(queueD(1,:), 'ascend'); % sort by criticality
    queueD = queueD(:,I);
    [Y,I] = sort(queueD(2,:), 'descend'); % sort by arrival time
    queueD = queueD(:,I);
...```

Result: (20 exits in each simulation; 1000 simulations)

<table>
<thead>
<tr>
<th>Number of 100% Death</th>
<th>Number of 50% Death</th>
<th>Total Number Completed</th>
<th>Make Span (minutes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.28</td>
<td>2.08</td>
<td>20</td>
<td>299.83</td>
</tr>
</tbody>
</table>
Advanced EDD Output Analysis

% check is job dead before arrival
if serviceD1(1,QCustomers+1)>queueD(4,1) % If job is dead before service
  serviceD1(2,QCustomers+1) = 0.001; % Set dead jobs as cleared
  serviceD1(3,QCustomers+1) = serviceD1(1,QCustomers+1) + serviceD1(2,QCustomers+1); % end time
  NDeath = NDeath+1;
end

% check is the job expected to die before service end
if serviceD1(3,QCustomers+1)>queueD(4,1)
  serviceD1(2,QCustomers+1) = 0.001; % Set dead jobs as cleared
  serviceD1(3,QCustomers+1) = serviceD1(1,QCustomers+1) + serviceD1(2,QCustomers+1); % end time
  NDeath = NDeath+1;
end

Result: (20 exits in each simulation; 1000 simulations)

<table>
<thead>
<tr>
<th>Number of 100% Death</th>
<th>Number of 50% Death</th>
<th>Total Number Completed</th>
<th>Make Span (minutes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.19</td>
<td>3.07</td>
<td>20</td>
<td>272.32</td>
</tr>
</tbody>
</table>
Results & Conclusions

- Advanced Expected Death EDD output can be used as a feasible LB
- EDD is not necessarily better than FIFO
- Make Span from FIFO and EDD are very close to each other
- EDD can result in large queues
  - i.e. for a simulation with 20 jobs, EDD can end up having >100 people in queue

<table>
<thead>
<tr>
<th></th>
<th>Number of 100% Death</th>
<th>Number of 50% Death</th>
<th>Make Span (minutes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIFO</td>
<td>7.08</td>
<td>2.12</td>
<td>299.55</td>
</tr>
<tr>
<td>EDD</td>
<td>7.28</td>
<td>2.08</td>
<td>299.83</td>
</tr>
<tr>
<td>Advanced EDD</td>
<td>4.19</td>
<td>3.07</td>
<td>272.32</td>
</tr>
</tbody>
</table>

Advanced EDD has the least number of death in every type
In all three algorithms, most death happens in “Critical”
EDD yields higher number of death in all three types of patients
Also keep in mind ... sanity check

- Advanced EDD = Let patients die before being seen by doctors or nurses?
  - How practical that really is?

- Who determine the “criticality” of the patients?
  - Possibilities of making mistake on assigning “criticality” to patients?
  - How do we know before patients going through arbitrage?

- Advanced EDD discard 100% certainty death, what about 50% death?

- Offline Scheduling
  - Know the arrival of each patients then schedule the jobs
  - Will this be a better LB?
Thank you!