Exponential closed networks

- Workstations 1, \ldots, M
- Workstation $m$ has $c_m$ parallel identical machines
- $N$ circulating jobs ($N$ is the population size)
- Processing times in workstation $m$ are exponential with rate $\mu_m$
- Processing order is FCFS
- Buffers are unlimited
- Markovian routing:
  job moves from workstation $m$ to $n$ with probability $p_{mn}$

This network is also called Closed Jackson network
Exponential closed networks

Fixed population size $N$

Network of $M$ work stations

1. $\mu_1$

2. $\mu_2$

$\cdots$

$p_{12}$

$p_{1M}$

1. $\mu_1$

2. $\mu_2$

$\cdots$

$p_{12}$

$p_{1M}$
Example: Robotic barn
Example: Robotic barn

How to design a robotic barn? How many robots?
Closed network with $K$ circulating cows and 6 workstations:

1. Milking robot,
2. Concentrate feeder,
3. Forage lane,
4. Water trough,
5. Cubicle and
6. (artifical one) Walking.
Example: Robotic barn

Closed network with $K$ circulating cows and 6 workstations:

- Milking
- Concentrate
- Forage
- Water
- Cublices
- Walking
Example: Zone-Picking
Issues in design:

- What should be the layout of the network?
- Size of zones?
- Where to locate items?
- What number of pickers and zones?
- Required CONWIP level?
Example: Single Zone

- Storage Zone
- Buffer
- Weight check
- System entrance/exit
- Recirculation
- Tote
- Order picker
Example: Single Zone

Closed network with $K$ totes and 6 workstations
Example: KIVA robots
Example: KIVA robots

How to design a KIVA system? How many robots?
Example: KIVA robots

Closed queueing network model with $K$ circulating robots
Example: Container terminal

How many AGVs needed for unloading ship?
One of the examples shows how the method can be applied to solve a queueing network drawn from a model of a sea container terminal. In this model, a fixed number of transportation vehicles (AGVs) go round, which represent the jobs in a closed system. The container handling is done in two separate phases, connected by a transportation phase. The first phase and last phase represent stacking operations of the cranes. For unloading, the first phase will be the retrieval of a container from the vessel together with the placement of the container on the AGV. The last phase would represent the unloading of the container from the AGV and the storage operation for further processing. For the loading procedure, the route of the containers is reversed. The transportation process is modeled to be subject to congestion.

The purpose of this model (which is inspired on the Jawaharlal Nehru Port near Mumbai, India) is to predict the performance of the terminal in terms of throughput. A general layout is shown in Figure 1.

This model is translated to the following queuing network. The container handling operations are modeled as a two phase process. One service cycle consists of a setup phase, for which the customer does not need to be present at the server and a service phase, for which the customer has to be present at the server. Both have exponentially distributed service times. The congestion effect is modeled as a load dependent exponential server. The complete model is shown in Figure 2.

Abstract view of load/unload process
Example: Container terminal

Closed queueing network model with $K$ circulating AGVs
States of network \((k_1, \ldots, k_M)\) where \(k_m\) is number of jobs in workstation \(m\)

Note that

\[
\sum_{m=1}^{M} k_m = N
\]

so there are \(\binom{N+M-1}{M-1}\) states!

State probabilities \(p(k_1, k_2, \ldots, k_M)\) satisfy balance equations \((c_m = 1)\)

\[
p(k) \sum_{m=1}^{M} \mu_m \epsilon(k_m) = \sum_{n=1}^{M} \sum_{m=1}^{M} p(k + e_n - e_m) \mu_n p_{nm} \epsilon(k_m)
\]

where \(e_m = (0, \ldots, 1, \ldots, 0)\) with 1 at place \(m\) and \(\epsilon(k) = \begin{cases} 1 & \text{if } k > 0 \\ 0 & \text{else} \end{cases} \)
Product form solution “Jackson’s miracle”

\[ p(k) = C p_1(k_1) p_2(k_2) \cdots p_M(k_M), \]

where \( C \) is normalizing constant and

\[ p_m(k_m) = \left( \frac{\nu_m}{\mu_m} \right)^{k_m}, \quad k_m = 0, 1, \ldots \]

with \( \nu_m \) the “arrival rate” to workstation \( m \)

This is again the product of \( M/M/1 \) solutions, where the number in station \( m \) follows the \( M/M/1 \) solution with arrival rate \( \nu_m \) and service rate \( \mu_m \)!
$v_m$ is the relative arrival rate or visiting frequency to $m$, satisfying

$$v_m = \sum_{n=1}^{M} v_n p_{nm}, \quad m = 1, \ldots, M$$

Remarks:

- Equations above determine $v_m$’s up to a multiplicative constant
- Set $v_1 = 1$, then $v_m$ is the expected number of visits to $m$ in between two successive visits to station 1
- Product form result also valid for fixed routing
- Although $p(k)$ is again a product, the queues at stations are dependent!
Let

\[ C(m, n) = \sum_{k_1, \ldots, k_m \geq 0} \left( \frac{v_1}{\mu_1} \right)^{k_1} \left( \frac{v_2}{\mu_2} \right)^{k_2} \cdots \left( \frac{v_m}{\mu_m} \right)^{k_m} \]

So \( C(m, n) \) is sum of products in network with stations 1, \ldots, m and population n. Clearly \( C = 1/C(M, N) \)

Recursion (Buzen’s algorithm):

\[ C(m, n) = C(m - 1, n) + \frac{v_m}{\mu_m} C(m, n - 1) \]

with initial conditions

\[ C(0, n) = 0, \quad n = 1, \ldots, N, \quad C(m, 0) = 1, \quad m = 1, \ldots, M, \]
Recursion (Buzen’s algorithm):

\[ C(m, n) = C(m - 1, n) + \frac{\nu_m}{\mu_m} C(m, n - 1) \]

with initial conditions

\[ C(0, n) = 0, \quad n = 1, \ldots, N, \quad C(m, 0) = 1, \quad m = 1, \ldots, M, \]
Mean values

- What is the real arrival rate $\lambda_m$?
  Note that
  \[
  \lambda_M = v_m \frac{C(M, N - 1)}{C(M, N)}
  \]
  and
  \[
  \lambda_m = \frac{v_m}{v_M} \lambda_M
  \]

- What is mean number $E(L_M)$ in station $M$?
  \[
  E(L_M) = \frac{1}{C(M, N)} \sum_{k_M=0}^{N} k_M \left( \frac{v_M}{\mu_M} \right)^{k_M} C(M - 1, N - k_M)
  \]

- What is expected cycle time $E(C)$ between two visits to station 1?
  \[
  E(C) = \frac{N}{\lambda_1} \quad \text{(Little’s law)}
  \]
Exponential multi server network

Product form solution

\[ p(k) = C p_1(k_1) p_2(k_2) \cdots p_M(k_M), \]

where \( C \) is normalizing constant and

\[ p_m(k_m) = \prod_{k=1}^{k_m} \frac{v_m}{\mu_m(k)} \]

where \( \mu_m(k) = \min(k, c_m) \mu_m \) and \( v_m \) the visiting frequency to workstation \( m \)

This is product of \( M/M/c_m \) solutions with arrival rate \( v_m \) and service rate \( \mu_m \)!

Normalizing constant \( C \) can again be calculated via recursion (verify!)
**Question:** What is the state seen by job moving from one station to another?

Total number of jumps per time unit that see the (single server) network in state \( k \in S(N - 1) = \{ k \geq 0 | \sum_{i=1}^{M} k_i = N - 1 \} \)

\[
\sum_{m=1}^{M} p(k + e_m) \mu_m = \frac{1}{C(M, N)} p_1(k_1) \cdots p_M(k_M) \sum_{m=1}^{M} v_m,
\]

where \( p_m(k_m) = \left( \frac{v_m}{\mu_m} \right)^{k_m} \)

Total number of all jumps per time unit in the (single server) network

\[
\sum_{l \in S(N-1)} \sum_{m=1}^{M} p(l + e_m) \mu_m = \frac{1}{C(M, N)} \sum_{l \in S(N-1)} p_1(l_1) \cdots p_M(l_M) \sum_{m=1}^{M} v_m,
\]
Fraction of jumps per time unit that see the network in state $k \in S(N - 1)$

$$\frac{1}{C(M,N)} p_1(k_1) \cdots p_M(k_M) \sum_{m=1}^{M} v_m$$

$$\frac{1}{C(M,N)} \sum_{l \in S(N-1)} p_1(l_1) \cdots p_M(l_M) \sum_{m=1}^{M} v_m = \frac{1}{C(M, N - 1)} p_1(k_1) \cdots p_M(k_M)$$

which is probability that network with $N - 1$ circulating jobs is in state $k$

**Conclusion:**

Arbitrary job moving from one station to another sees the network in equilibrium with a population with one job less (job does not see himself)

**Remarks:**

- Also valid in multi-server networks (verify!)
- Also valid for jobs moving to a specific station (verify!)
- What is the impact of this result?
Define for network with population $k$

\[
E(S_m(k)) = \text{mean production lead time at station } m
\]
\[
\Lambda_m(k) = \text{throughput of station } m
\]
\[
E(L_m(k)) = \text{mean number of jobs in station } m
\]

For population $k = 1, 2, \ldots, N$ in single server network

\[
E(S_m(k)) = E(L_m(k - 1)) \frac{1}{\mu_m} + \frac{1}{\mu_m} \quad \text{(Arrival theorem)}
\]
\[
\Lambda_m(k) = \frac{k \nu_m}{\sum_{n=1}^{M} \nu_n E(S_n(k))} \quad \text{(Little)}
\]
\[
E(L_m(k)) = \Lambda_m(k) E(S_m(k)) \quad \text{(Little)}
\]

with initially $E(L_m(0)) = 0$
In multi server network

\[ E(S_m(k)) = \prod_m (k - 1) \frac{1}{c_m \mu_m} + \left( E(L_m(k - 1)) - \frac{\Lambda_m(k - 1)}{\mu_m} \right) \frac{1}{c_m \mu_m} + \frac{1}{\mu_m} \]

where \( \prod_m (k - 1) \) is probability that all servers are busy

Approximate \( \prod_m (k - 1) \) by probability of waiting in corresponding \( M/M/c_m \)

\[ \prod_m (k - 1) \approx \frac{1}{c_m!} \left( \frac{\Lambda_m(k-1)}{\mu_m} \right)^{c_m} \left( 1 - \frac{\Lambda_m(k-1)}{c_m \mu_m} \right) \sum_{i=0}^{c_m-1} \frac{1}{i!} \left( \frac{\Lambda_m(k-1)}{\mu_m} \right)^i + \frac{1}{c_m!} \left( \frac{\Lambda_m(k-1)}{\mu_m} \right)^{c_m} \]

If \( c_m = \infty \) (no waiting)

\[ E(S_m(k)) = \frac{1}{\mu_m} \]
In multi server station

\[ E(S_m(k)) = \Pi_m(k - 1) \frac{E(R_m)}{c_m} + (E(L_m(k - 1)) - \Lambda_m(k - 1)E(B_m)) \frac{E(B_m)}{c_m} + E(B_m) \]

where \( \Pi_m(k - 1) \) is approximated by probability of waiting in \( M/M/c \)

In single server station this reduces to

\[ E(S_m(k)) = \rho_m(k - 1)E(R_m) + (L_m(k - 1) - \rho_m(k - 1)) E(B_m) + E(B_m) \]

where \( \rho_m(k - 1) = \Lambda_m(k - 1) E(B_m) \)
Closed system with 4 single server stations and 10 circulating pallets:

Processing characteristics:

<table>
<thead>
<tr>
<th>Station</th>
<th>$E(B_m)$</th>
<th>$c_{Bm}^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.25</td>
<td>0.25</td>
</tr>
<tr>
<td>2</td>
<td>1.25</td>
<td>0.50</td>
</tr>
<tr>
<td>3</td>
<td>2.00</td>
<td>0.33</td>
</tr>
<tr>
<td>4</td>
<td>1.60</td>
<td>1.00</td>
</tr>
</tbody>
</table>
Mean value analysis: \( \Lambda_1(10) = 0.736 \) parts per time unit
Simulation: \( \Lambda_1(10) = 0.743 \pm 0.003 \) parts per time unit

<table>
<thead>
<tr>
<th>Station</th>
<th>( E(S_m(10)) )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( \text{amva} )</td>
</tr>
<tr>
<td>1</td>
<td>4.417</td>
</tr>
<tr>
<td>2</td>
<td>5.050</td>
</tr>
<tr>
<td>3</td>
<td>4.181</td>
</tr>
<tr>
<td>4</td>
<td>4.086</td>
</tr>
</tbody>
</table>
Example

Production system:

- $C$ machines
- $N$ pallets
- $M$ operations to be performed
- each operation requires a **specific** tool set
- $r_m$ copies of tool set $m$
- $v_m E(B_m)$ is work load to be handled by tool set $m$
Optimization problem:

\[
\text{max } TH(c_1, c_2, \ldots, c_m) \\
\text{subject to} \\
\sum_{m=1}^{M} c_m \leq C, \\
1 \leq c_m \leq r_m, \quad m = 1, 2, \ldots, M.
\]
Optimization problem:

\[
\max TH(c_1, c_2, \ldots, c_m)
\]
subject to
\[
\sum_{m=1}^{M} c_m \leq C,
\]
\[
1 \leq c_m \leq r_m, \quad m = 1, 2, \ldots, M.
\]

Heuristic solution:

- Subsequently allocate tool sets to machines
- allocate tool set with \textbf{maximum increase in throughput}
Example: Robotic barn

Closed network with $K$ circulating cows and 6 workstations:

1. Milking robot,
2. Concentrate feeder,
3. Forage lane,
4. Water trough,
5. Cubicle and
6. (artifical one) Walking.
Example: Robotic barn

Histogram of the processing time (in min.) in the milking robot:
### Example: Robotic barn

Processing times in the facilities of the barn:

<table>
<thead>
<tr>
<th>Facility</th>
<th>Routing probability</th>
<th>Processing time (in min.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Milking robot</td>
<td>0.164</td>
<td>8.41</td>
</tr>
<tr>
<td>Concentrate feeder</td>
<td>0.155</td>
<td>6.38</td>
</tr>
<tr>
<td>Forage lane</td>
<td>0.235</td>
<td>15.0</td>
</tr>
<tr>
<td>Water trough</td>
<td>0.170</td>
<td>3.18</td>
</tr>
<tr>
<td>Cubicle</td>
<td>0.276</td>
<td>38.9</td>
</tr>
</tbody>
</table>
General closed network model of robotic dairy barn.
type cow = tuple (real arr; int stat);
type cow_walk = tuple(cow x; timer t);
proc B(chan? cow a; chan! cow b):
    list cow xs;
    cow x;

    while true:
        select
            a?x:
                xs = xs + [x]
        alt
            size(xs) > 0, b!xs[0]:
                xs = xs[1:]
        end
    end
end
proc M(chan? cow a; chan! cow b, c; dist real u):
cow x;

while true:
a?x;
b!x;
delay sample u;
c!x;
end

end
proc W(chan? cow a; chan! cow b, c; dist real u; int m):
    chan cow d;

run B(a,d),
    unwind j in range(m):
        M(d, b, c, u)
    end
end
proc WLK(chan? cow a; list chan! cow b; real walk):
    list cow_walk xst; cow x; list(1000) int dest;
    for i in range(1000):
        if i < 164:
            dest[i] = 0;
        elif i < 319:
            dest[i] = 1;
        ...
    while true:
        select
            a?x:
                x.arr = time + walk;
                x.stat = dest[sample uniform(0, 1000)];
                xst = xst + [(x, timer(walk))]
        alt
            not empty(xst) and ready(xst[0].t),
            b[xst[0].x.stat]!xst[0].x:
                xst = xst[1:]
model DairyBarn():
chan cow a, c;
    list(5) chan cow b;

run G(a, 10),
    WLK(a, b, 5.0),
    W(b[0], c, a, exponential(8.41), 1),
    W(b[1], c, a, exponential(6.38), 1),
    W(b[2], c, a, exponential(15.0), 1),
    W(b[3], c, a, exponential(3.18), 1),
    W(b[4], c, a, exponential(38.9), 1),
    E(c, 100000)
end
• $n_m$ distinct types of operations at (single server) work station $m$
• $v_{mr}$ visits to work station $m$ for type $r$ operation
• mean processing time for type $r$ operation at work station $m$ is $E(B_{mr})$
• mean residual processing time is $E(R_{mr})$
Define

\[ E(S_{mr}(k)) = \text{mean production lead time at station } m \text{ for job of type } r \text{ operation} \]
\[ \Lambda_{mr}(k) = \text{arrival rate at station } m \text{ of jobs for type } r \text{ operation} \]
\[ E(L_{mr}(k)) = \text{mean number of jobs at station } m \text{ for type } r \text{ operation} \]

Then

\[ E(S_{mr}(k)) = \sum_{s=1}^{n_m} \rho_{ms}(k-1) E(R_{ms}) + \sum_{s=1}^{n_m} (E(L_{ms}(k-1)) - \rho_{ms}(k-1)) E(B_{ms}) + E(B_{mr}) \]

where \( \rho_{ms}(k-1) = \Lambda_{ms}(k-1) E(B_{ms}) \) and

\[ \Lambda_{mr}(k) = \frac{k v_{mr}}{\sum_{n=1}^{M} \sum_{s=1}^{n_m} v_{ns} E(S_{ns}(k))} \]
\[ E(L_{mr}(k)) = \Lambda_{mr}(k) E(S_{mr}(k)) \]
Closed multi-class networks

- Workstations 1, \ldots, \textit{M}
- Workstation \textit{m} has \textit{c}_m parallel identical machines
- \textit{R} job types
- \textit{N}_r circulating jobs of type \textit{r}
- Processing times in workstation \textit{m} are exponential with rate \textit{\mu}_m (so processing times are job-type independent!)
- Processing order is FCFS
- Buffers are unlimited
- \textbf{Markovian routing:}
  - type \textit{r} job moves from workstation \textit{m} to \textit{n} with probability \textit{p}^{\textit{r}}_{\textit{mn}} (so each job type has its own Markovian routing)

This network is also called \textbf{Closed multi-class Jackson network}
Closed multi-class networks

States of network \((k_1, \ldots, k_M)\) where

- \(k_m = (k_{m1}, \ldots, k_{mR})\) is the aggregate situation in station \(k\)
- \(k_{mr}\) is the number of type \(r\) jobs in workstation \(m\)

Note that for each \(r\)

\[
\sum_{m=1}^{M} k_{mr} = N_r
\]

\(v_{mr}\) is the relative visiting frequency to station \(m\) of type \(r\) jobs satisfying

\[
v_{mr} = \sum_{n=1}^{M} v_{nr} p_{nm}^r, \quad m = 1, 2, \ldots, M.
\]
Jackson’s miracle

\[ p(k) = C p_1(k_1) p_2(k_2) \cdots p_M(k_M), \]

where \( C \) is normalizing constant

If \( c_m = 1 \)

\[ p_m(k_m) = \frac{(k_{m1} + k_{m2} + \cdots + k_{mR})!}{k_{m1}!k_{m2}! \cdots k_{mR}!} \left( \frac{\nu_{m1}}{\mu_m} \right)^{k_{m1}} \left( \frac{\nu_{m2}}{\mu_m} \right)^{k_{m2}} \cdots \left( \frac{\nu_{mR}}{\mu_m} \right)^{k_{mR}} \]

If \( c_m > 1 \)

\[ p_m(k_m) = \frac{(k_{m1} + k_{m2} + \cdots + k_{mR})!}{k_{m1}!k_{m2}! \cdots k_{mR}!} \frac{\nu_m^{k_{m1}} \nu_m^{k_{m2}} \cdots \nu_m^{k_{mR}}}{\mu_m(1) \mu_m(2) \cdots \mu_m(k_{m1} + k_{m2} + \cdots + k_{mR})} \]

where \( \mu_m(k) = \min(k, c_m) \mu_m \)
Arbitrary type $r$ job moving from one station to another sees the network in equilibrium with a population with one job of his own type less (job does not see himself)

$\underline{N} = (N_1, N_2, \ldots, N_R)$ is the population vector

So jumping type $r$ job sees the network in equilibrium with population $\underline{N} - e_r$
Mean value analysis

Define for network with population $N$

\[
E(S_{mr}(N)) = \text{mean production lead time at work station } m \text{ for type } r \text{ job}
\]
\[
\Lambda_{mr}(N) = \text{throughput of type } r \text{ jobs of station } m
\]
\[
E(L_{mr}(N)) = \text{mean number of type } r \text{ jobs in station } m
\]

In single-server network

\[
E(S_{mr}(N)) = \sum_{s=1}^{r} E(L_{ms}(N - e_r)) \frac{1}{\mu_m} + \frac{1}{\mu_m}
\]
\[
\Lambda_{mr}(N) = \frac{N_r v_{mr}}{\sum_{n=1}^{M} v_{nr} E(S_{nr}(N))}
\]
\[
E(L_{mr}(N)) = \Lambda_{mr}(N) E(S_{mr}(N))
\]

with initially $E(L_{ms}(0))$

Recursion over population vector $N$, starting from $k = 0$ to $k = N!$
Define for network with population $N$

\[ E(W_{mr}(N)) = \text{mean waiting time at work station } m \text{ for type } r \text{ job} \]

\[ \Lambda_{mr}(N) = \text{throughput of type } r \text{ jobs of station } m \]

\[ E(Q_{mr}(N)) = \text{mean number of type } r \text{ jobs waiting in station } m \]

In non-preemptive (single server) priority station $m$ (type 1 highest priority)

\[
E(W_{mr}(N)) = \sum_{s=1}^{R} \rho_{ms}(N - e_r) \frac{1}{\mu_m} + \sum_{s=1}^{r} E(Q_{ms}(N - e_r)) \frac{1}{\mu_m} + \sum_{s=1}^{r-1} \Lambda_{ms}(N - e_r) E(W_{mr}(N)) \frac{1}{\mu_m}
\]

where $\rho_{mr}(N) = \Lambda_{mr}(N) \frac{1}{\mu_m}$
Breaking the recursion:

Assume jumping type $r$ job sees the system in equilibrium with population $N$ (instead of $N - e_r$)

In FCFS single server station $m$

$$E(S_{mr}(N)) = \sum_{s=1}^{r} E(L_{ms}(N)) \frac{1}{\mu_m} + \frac{1}{\mu_m}$$

So mean number seen on arrival is mean number in system **including himself**
Breaking the recursion:

Assume jumping type $r$ job sees the system in equilibrium with population $N$ (instead of $N - e_r$)

In FCFS single server station $m$

$$E(S_{mr}(N)) = \sum_{s=1}^{r} E(L_{ms}(N)) \frac{1}{\mu_m} + \frac{1}{\mu_m}$$

So mean number seen on arrival is mean number in system including himself

To avoid self queueing

$$E(S_{mr}(N)) = \sum_{s \neq r} E(L_{ms}(N)) \frac{1}{\mu_m} + \frac{N_r - 1}{N_r} E(L_{mr}(N)) \frac{1}{\mu_m} + \frac{1}{\mu_m}$$
3MR equations for 3MR unknowns $E(S_{mr}(N))$, $\Lambda_{mr}(N)$ and $E(L_{mr}(N))$

\[
E(S_{mr}(N)) = \sum_{s \neq r} E(L_{ms}(N)) \frac{1}{\mu_m} + \frac{N_r - 1}{N_r} E(L_{mr}(N)) \frac{1}{\mu_m} + \frac{1}{\mu_m}
\]

\[
\Lambda_{mr}(N) = \frac{N_r v_{mr}}{\sum_{n=1}^{M} v_{nr} E(S_{nr}(N))}
\]

\[
E(L_{mr}(N)) = \Lambda_{mr}(N) E(S_{mr}(N))
\]

Solution by successive substitutions