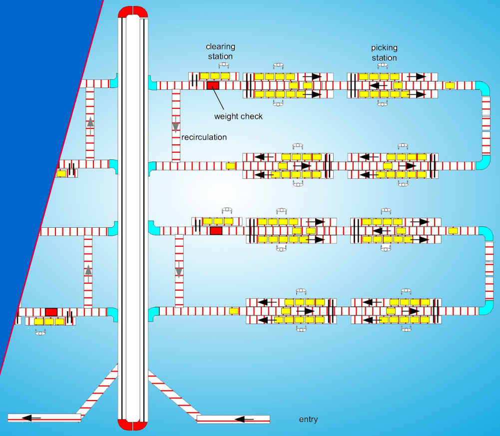
Stochastic Models of Manufacturing Systems

Ivo Adan



TU/e Technische Universiteit Eindhoven University of Technology

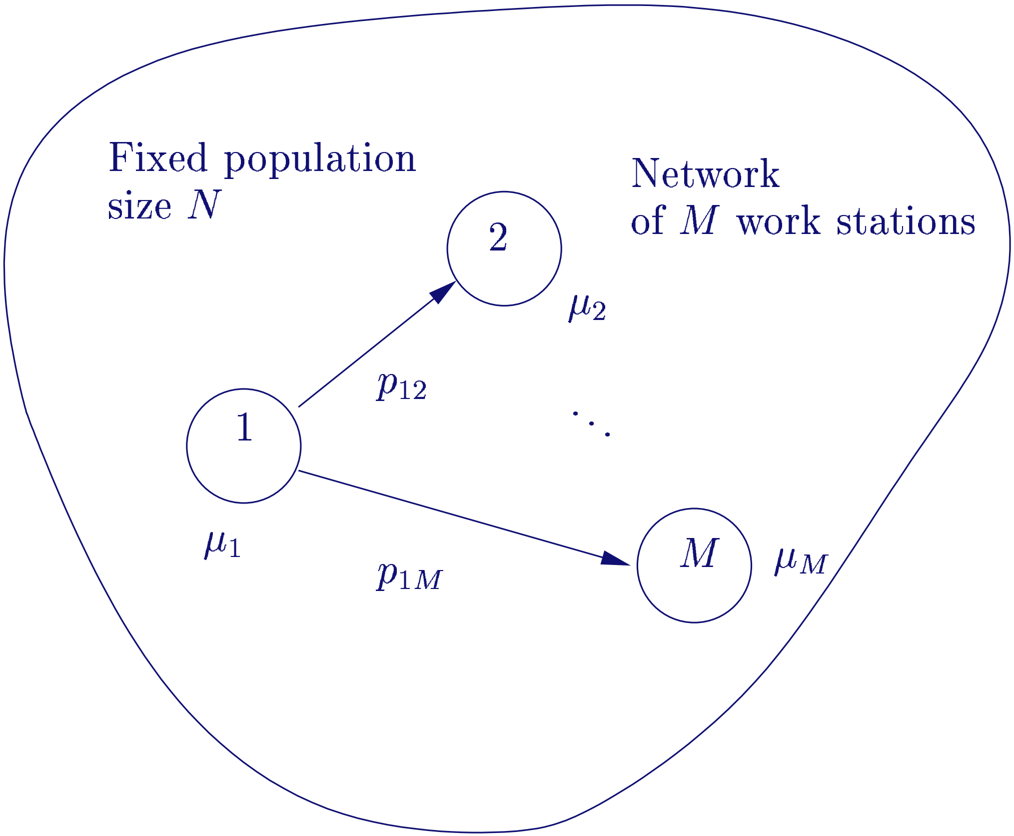
Tuesday June 16

Exponential closed networks

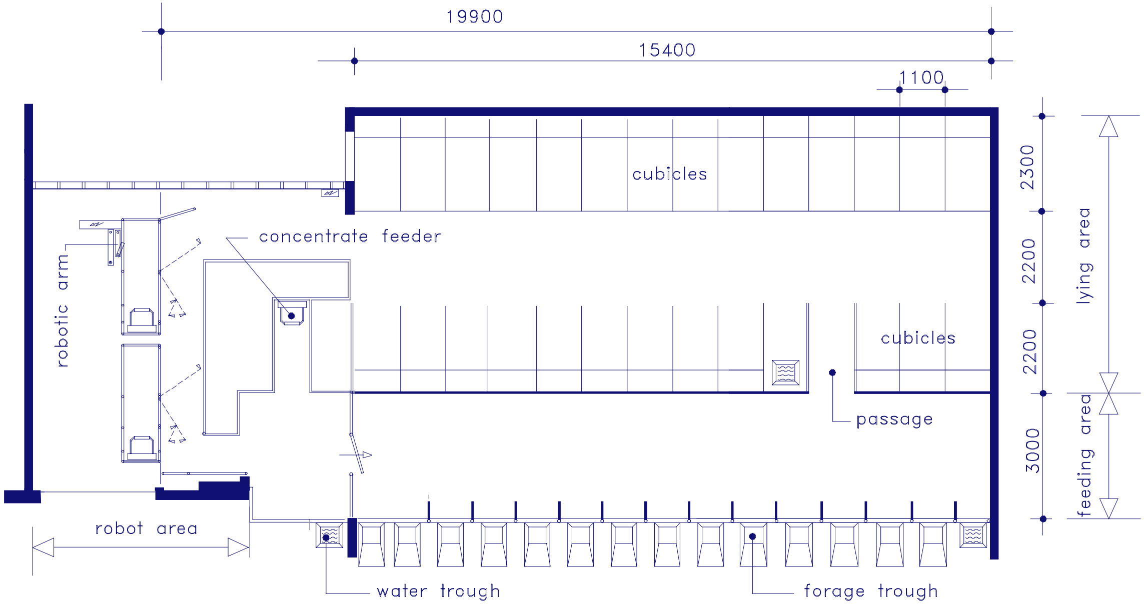
- Workstations $1, \ldots, M$
- Workstation m has c_m parallel identical machines
- N circulating jobs (N is the population size)
- Processing times in workstation m are exponential with rate μ_m
- Processing order is FCFS
- Buffers are unlimited
- Markovian routing:
  job moves from workstation m to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

This network is also called Closed Jackson network

Exponential closed networks

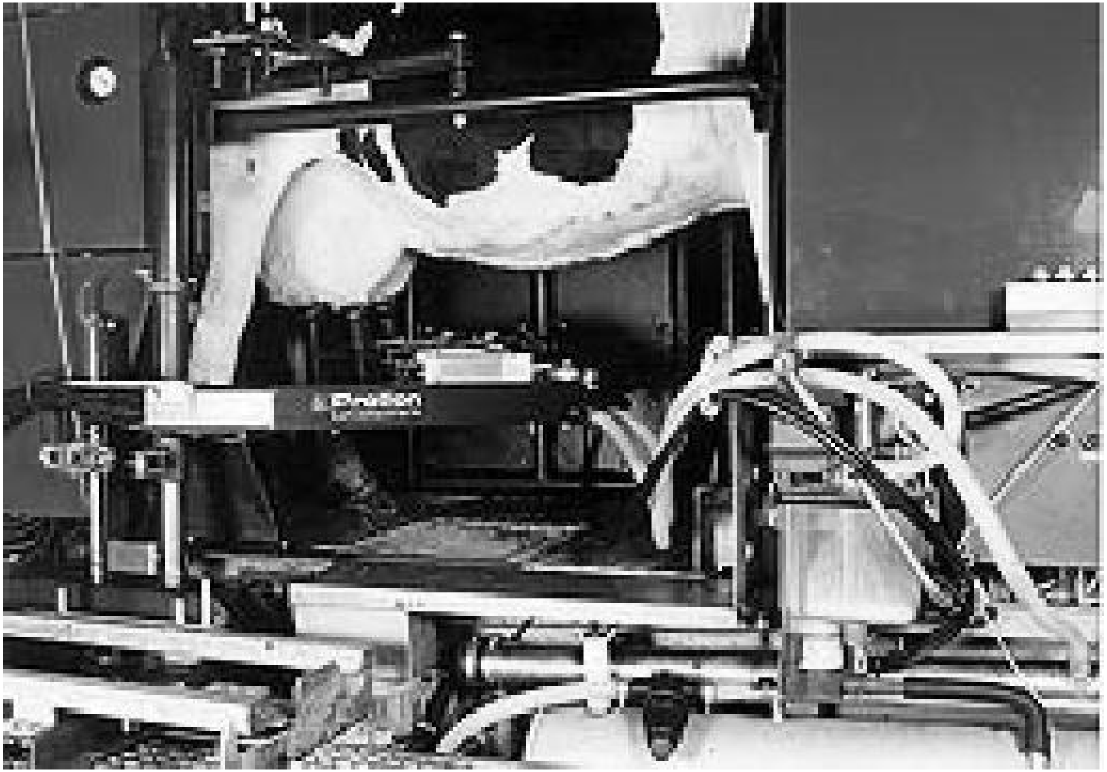


Example: Robotic barn



Technische Universiteit **Eindhoven** University of Technology

Example: Robotic barn



How to design a robotic barn? How many robots?

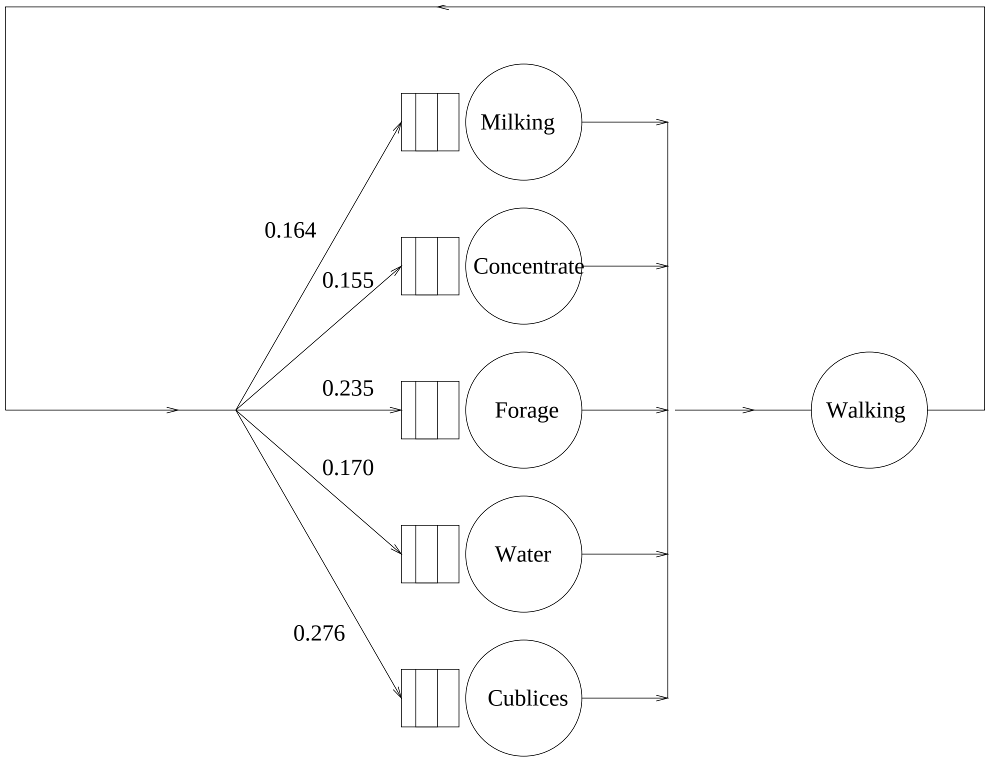
Example: Robotic barn

Closed network with K circulating cows (the herd) and 6 workstations:

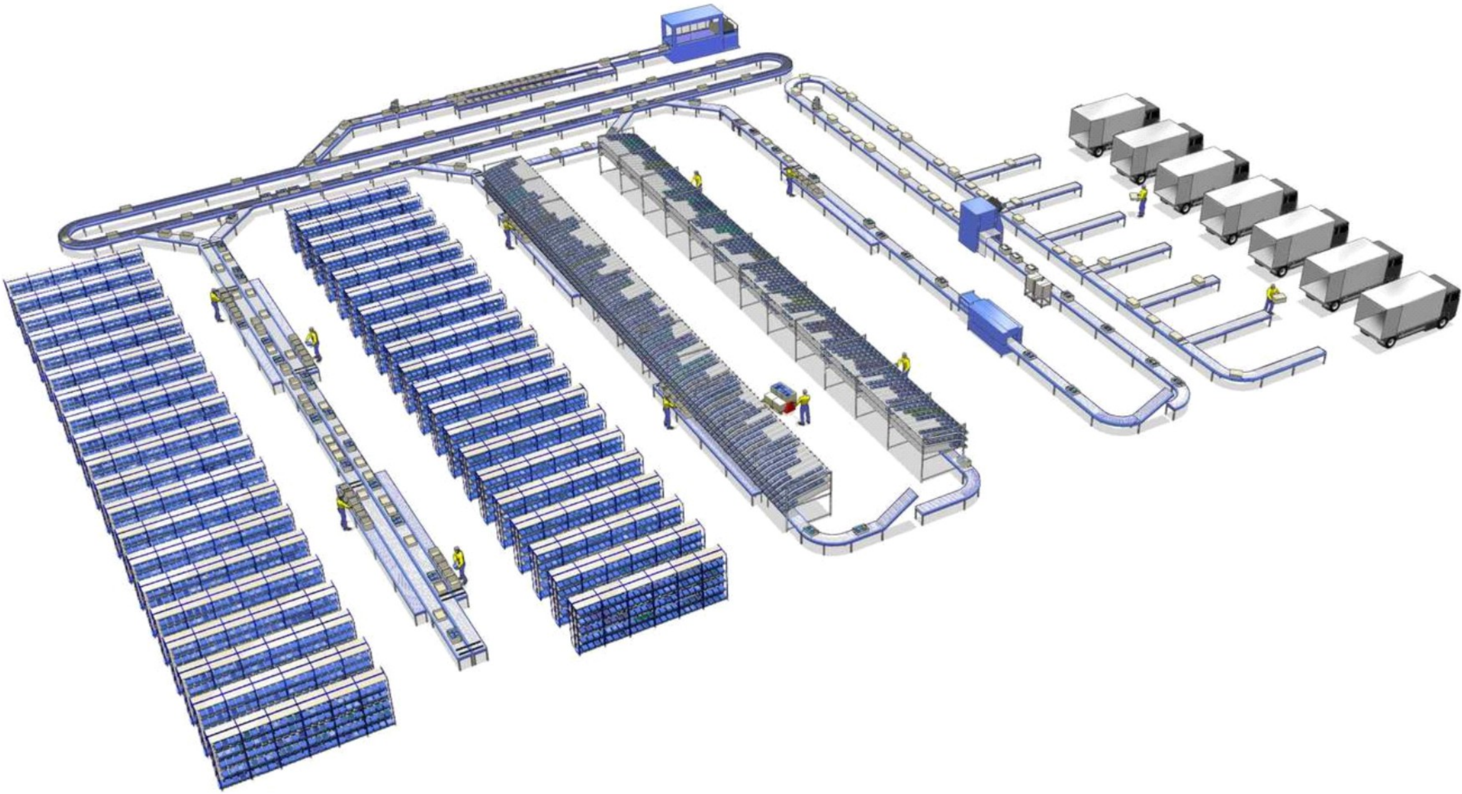
1. Milking robot,
2. Concentrate feeder,
3. Forage lane,
4. Water trough,
5. Cubicle and
6. (artifical one) Walking.

Example: Robotic barn

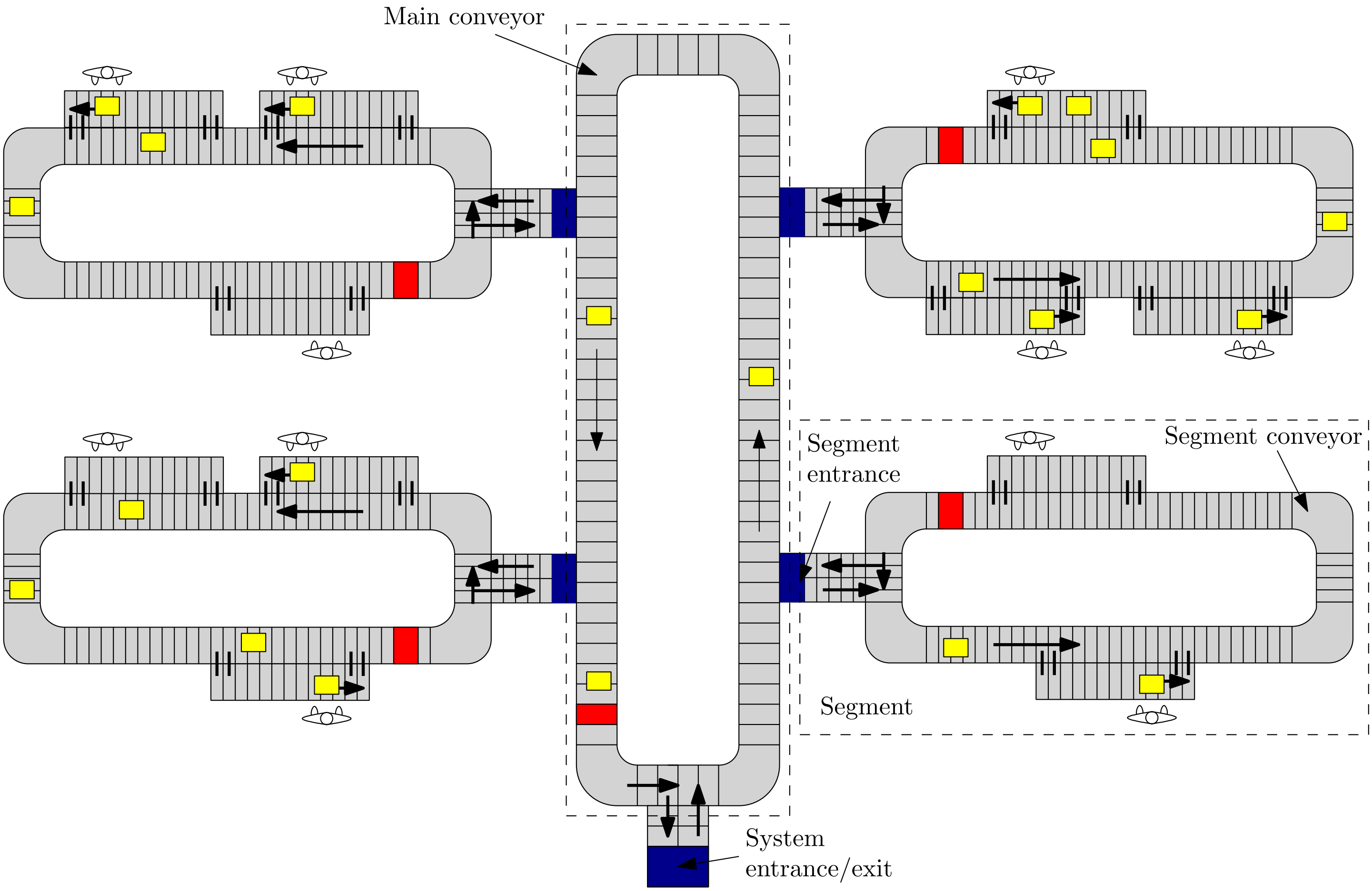
Closed network with K circulating cows and 6 workstations:



Zone-Picking Systems



Example: Zone-Picking

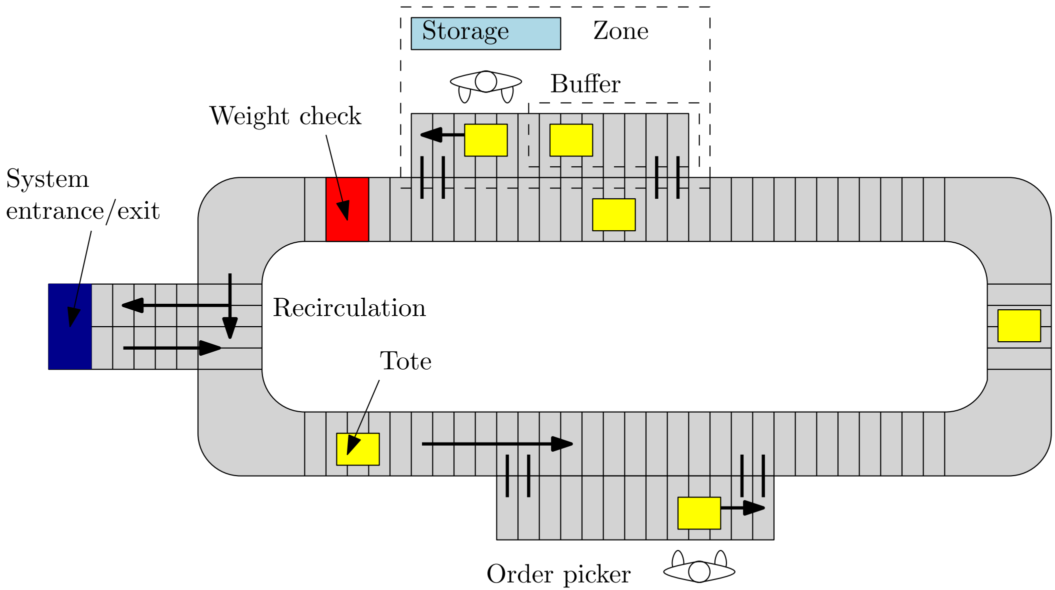


Example: Zone-Picking

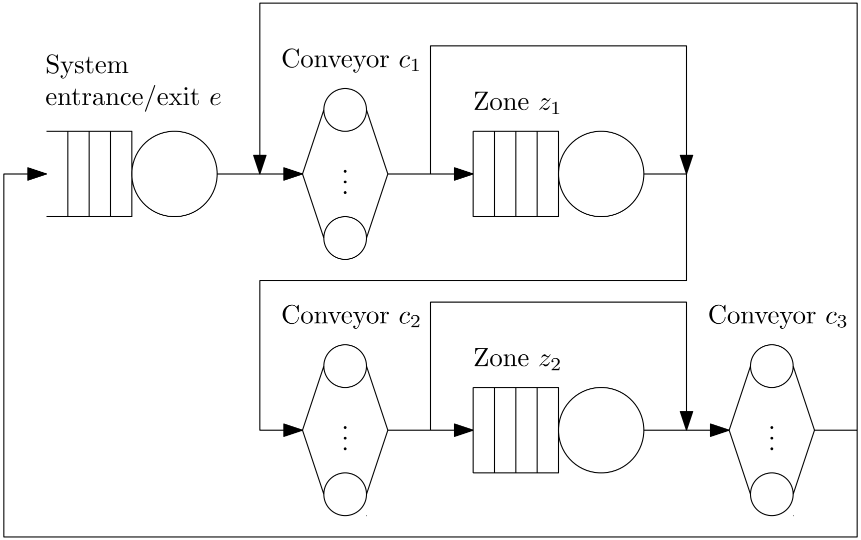
Issues in design:

- 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



Example: Single Zone



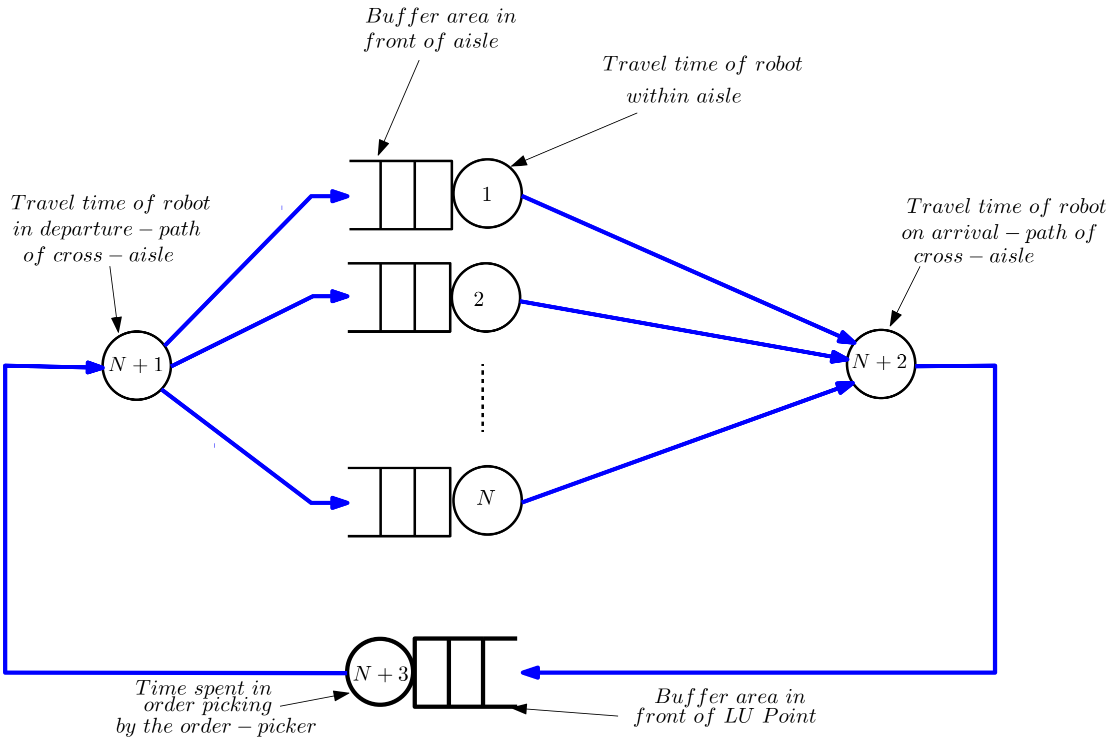
Closed network with K totes and 6 workstations

Example: KIVA robots

Example: KIVA robots

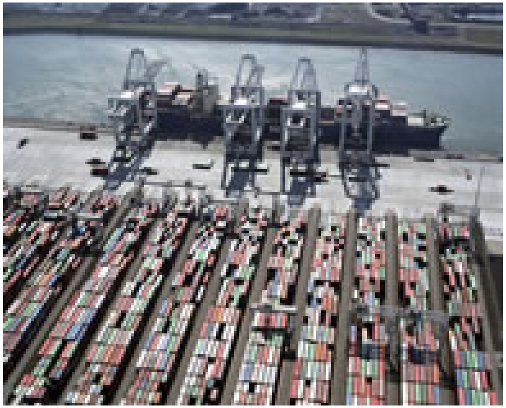
How to design a KIVA system? How many robots?

Example: KIVA robots



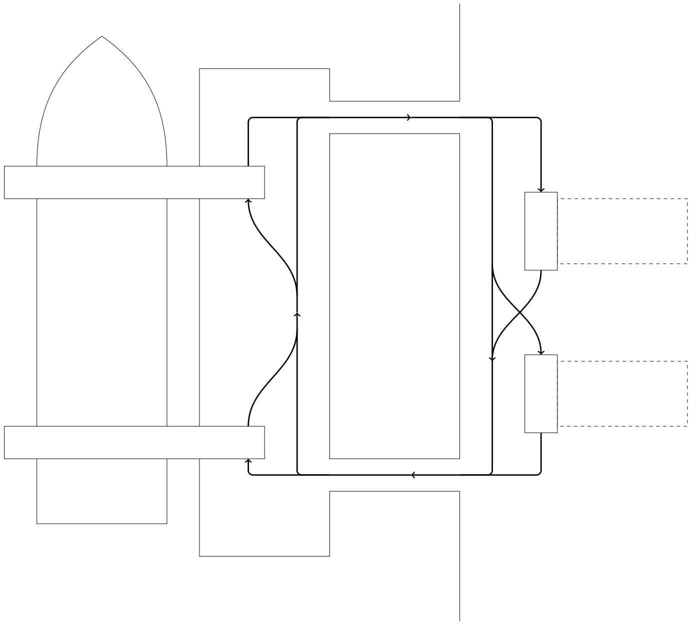
Closed queueing network model with K circulating robots

Example: Container terminal



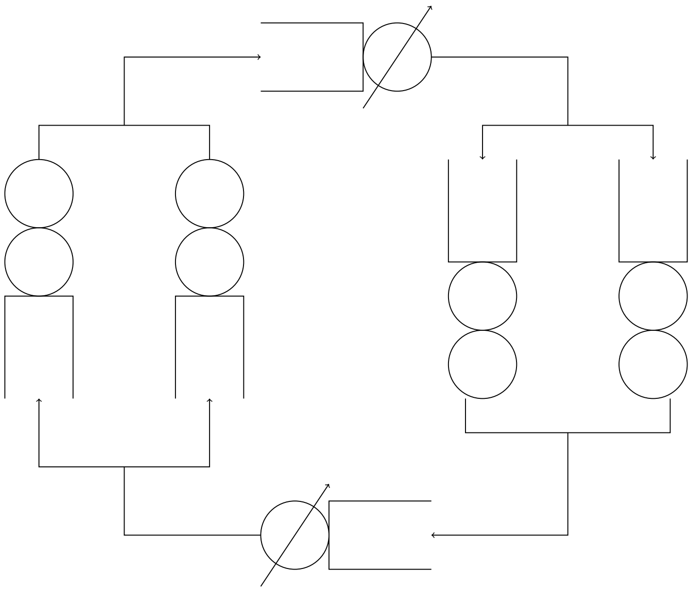
How many AGVs needed for unloading ship?

Example: Container terminal



Abstract view of load/unload process

Example: Container terminal



Closed queueing network model with K circulating AGVs

Exponential single server network

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 ($c_m = 1$)

$$\begin{aligned}
\text{Flow out of } \underline{k} \quad &= \quad \text{Flow into } \underline{k} \\
p(\underline{k}) \sum_{m=1}^{M} \mu_m \epsilon(k_m) \quad &= \quad \sum_{n=1}^{M} \sum_{m=1}^{M} p(\underline{k} + \underline{e}_n - \underline{e}_m) \mu_n p_{nm} \epsilon(k_m)
\end{aligned}$$

where $\underline{e}_m = (0, \ldots, 1, \ldots, 0)$ with 1 at place m and $\epsilon(k) = \begin{cases} 1 & \text{if } k > 0 \\ 0 & \text{else} \end{cases}$

Exponential single server network

Product form solution “Jackson’s miracle”

$$p(\underline{k}) = C p_1(k_1) p_2(k_2) \cdots p_M(k_M),$$

where C is normalizing constant and

$$p_m(k_m) = \left(\frac{v_m}{\mu_m}\right)^{k_m}, \quad k_m = 0, 1, \ldots$$

with v_m the “arrival rate” to workstation m

This is again a product of $M/M/1$ solutions:

Number in station m follows $M/M/1$ with arrival rate v_m and service rate μ_m!

Exponential single server network

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
- Although $p(\underline{k})$ is again a product, the queues at stations are dependent!
- Product form result also valid for fixed routing
- How to compute C?

Normalizing constant

Let

$$C(m,n) = \sum_{\substack{k_1,\ldots,k_m \geq 0 \\ \sum_{i=1}^m k_i = n}} \left(\frac{v_1}{\mu_1}\right)^{k_1} \left(\frac{v_2}{\mu_2}\right)^{k_2} \cdots \left(\frac{v_m}{\mu_m}\right)^{k_m}.$$

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

$$C(0,n) = 0, \quad n = 1,\ldots,N, \quad C(m,0) = 1, \quad m = 1,\ldots,M,$$

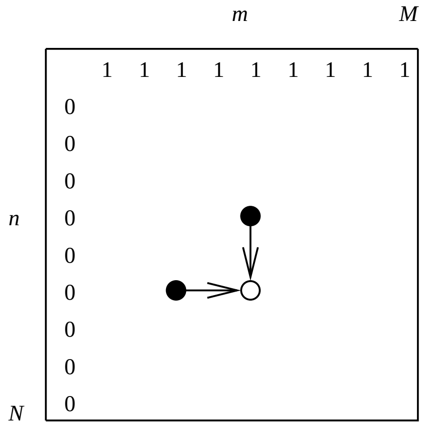
Normalizing constant

Recursion (Buzen's algorithm):

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

with initial conditions

$$C(0,n) = 0, \quad n = 1,\ldots,N, \quad C(m,0) = 1, \quad m = 1,\ldots,M,$$



Mean values

- What is the real arrival rate λ_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

$$p(\underline{k}) = C p_1(k_1) p_2(k_2) \cdots p_M(k_M),$$

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 μ_m!

Normalizing constant C can again be calculated via recursion (verify!)

Arrival theorem

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 $\underline{k} \in S(N-1) = \{\underline{k} \geq 0 | \sum_{i=1}^{M} k_i = N-1\}$

$$\sum_{m=1}^{M} p(\underline{k} + \underline{e}_m)\mu_m = \frac{1}{C(M,N)} p_1(k_1) \cdots p_M(k_M) \sum_{m=1}^{M} v_m,$$

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

$$\sum_{\underline{l} \in S(N-1)} \sum_{m=1}^{M} p(\underline{l} + \underline{e}_m)\mu_m = \frac{1}{C(M,N)} \sum_{\underline{l} \in S(N-1)} p_1(l_1) \cdots p_M(l_M) \sum_{m=1}^{M} v_m,$$

Arrival theorem

Fraction of jumps per time unit that see the network in state $\underline{k} \in S({\color{red}N - 1})$

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

which is probability that network with $N - 1$ circulating jobs is in state $\underline{k}$

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?

Mean value analysis

Define for network with population k

$$\begin{aligned} E(S_m(k)) &= \text{mean production lead time at station } m \\ \Lambda_m(k) &= \text{throughput of station } m \\ E(L_m(k)) &= \text{mean number of jobs in station } m \end{aligned}$$

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

$$\begin{aligned} E(S_m(k)) &= E(L_m(k-1))\frac{1}{\mu_m} + \frac{1}{\mu_m} \quad \text{(Arrival theorem)} \\ \Lambda_m(k) &= \frac{k v_m}{\sum_{n=1}^{M} v_n E(S_n(k))} \quad \text{(Little)} \\ E(L_m(k)) &= \Lambda_m(k) E(S_m(k)) \quad \text{(Little)} \end{aligned}$$

with initially $E(L_m(0)) = 0$ for all m

Remark:

- $\sum_{n=1}^{M} v_n E(S_n(k))$ is mean cycle time of job

Mean value analysis

In multi server network

$$E(S_m(k)) = \Pi_m(k-1)\frac{1}{c_m\mu_m} + \left(E(L_m(k-1)) - \frac{\Lambda_m(k-1)}{\mu_m}\right)\frac{1}{c_m\mu_m} + \frac{1}{\mu_m}$$

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

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

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

If $c_m = \infty$ (no waiting)

$$E(S_m(k)) = \frac{1}{\mu_m}$$

General closed networks

In multi server station

$$
\begin{aligned}
E(S_m(k)) \;&=\; \Pi_m(k-1)\,\frac{E(R_m)}{c_m} + (E(L_m(k-1)) - \Lambda_m(k-1)E(B_m))\,\frac{E(B_m)}{c_m} \\
&\quad + E(B_m)
\end{aligned}
$$

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

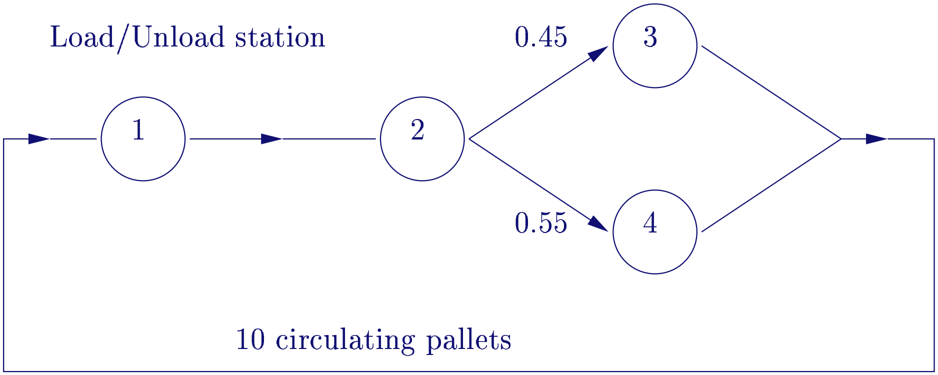
In single server station this reduces to

$$E(S_m(k)) = \rho_m(k-1)E(R_m) + (L_m(k-1) - \rho_m(k-1))\;E(B_m) + E(B_m)$$

where $\rho_m(k-1) = \Lambda_m(k-1)\;E(B_m)$

Example

Closed system with 4 single server stations and 10 circulating pallets:



Processing characteristics:

Station	$E(B_m)$	$c^2_{B_m}$
1	1.25	0.25
2	1.25	0.50
3	2.00	0.33
4	1.60	1.00

Example

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

Station	$E(S_m(10))$	
	amva	sim
1	4.417	4.890 ± 0.106
2	5.050	4.760 ± 0.169
3	4.181	3.860 ± 0.068
4	4.086	3.790 ± 0.118

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

Example

Optimization problem:

$$\max TH(c_1, c_2, \ldots, c_M)$$

subject to

$$\sum_{m=1}^{M} c_m \leq C,$$

$$1 \leq c_m \leq r_m, \quad m = 1, 2, \ldots, M.$$

where c_m is number of tool sets m being used

Example

Optimization problem:

$$\max TH(c_1, c_2, \ldots, c_m)$$

subject to

$$\sum_{m=1}^{M} c_m \leq C,$$

$$1 \leq c_m \leq r_m, \quad m = 1, 2, \ldots, M.$$

where c_m is number of tool sets m being used

Heuristic solution:

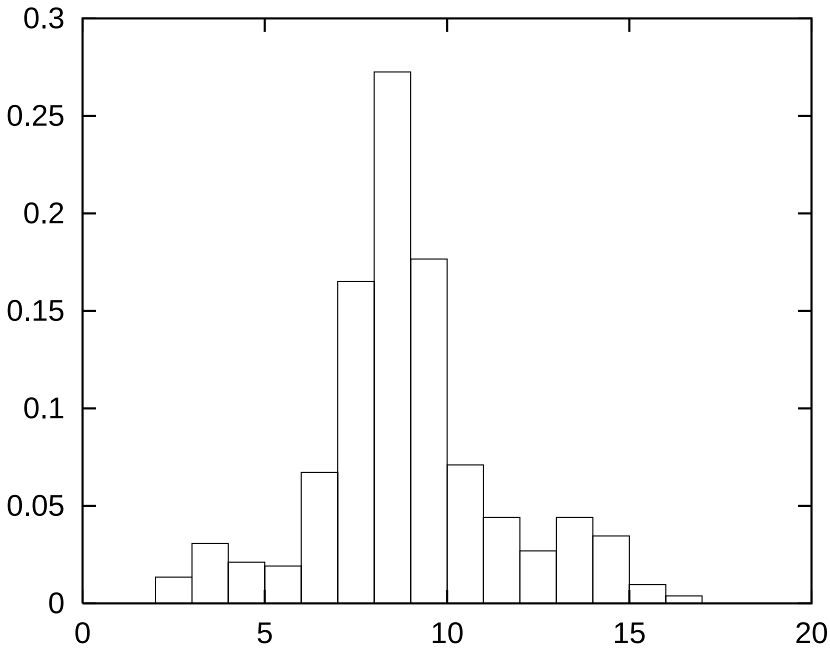
- Subsequently allocate tool sets to machines
- allocate tool set with maximum increase in throughput

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



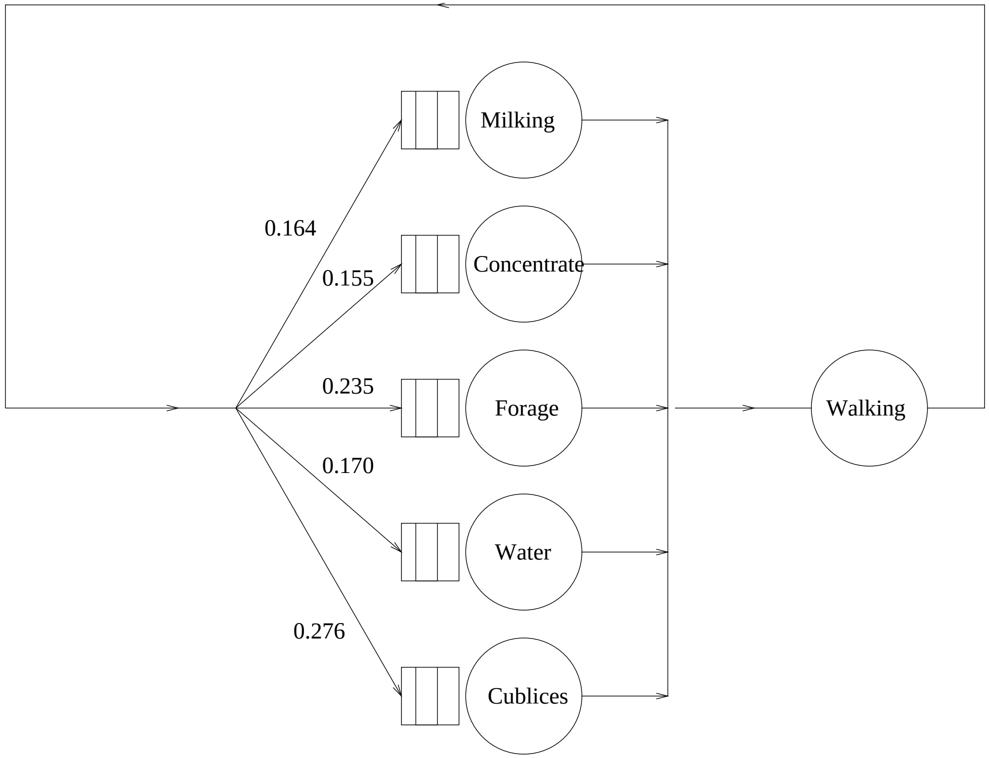
Histogram of the processing time (in min.) in the milking robot:

Example: Robotic barn

Processing times in the facilities of the barn:

		Processing time (in min.)	
Facility	Routing probability	Mean	Standard deviation
Milking robot	0.164	8.41	2.52
Concentrate feeder	0.155	6.38	6.25
Forage lane	0.235	15.0	11.9
Water trough	0.170	3.18	2.30
Cubicle	0.276	38.9	60.3

Example: Robotic barn



General closed network model of robotic dairy barn.

Type Cow


```
type cow = tuple (real arr; int stat);
type cow_walk = tuple(cow x; timer t);
```

Buffer


```
proc B(chan? cow a; chan! cow b):
    list cow xs;
    cow x;

    while true:
        select
            a?x:
                x.arr = time;
                xs = xs + [x]
        alt
            size(xs) > 0, b!xs[0]:
                xs = xs[1:]
        end
    end
end
```

Machine


```
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
```

Workstation


```
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
```

Workstation Walking


```
proc L(chan? cow a; chan! cow b; real walk):
    list cow_walk xst;
    cow x;

    while true:
        select
            a?x:
                xst = xst + [(x, timer(walk))]
        alt
            not empty(xst) and ready(xst[0].t), b!xst[0].x
                xst = xst[1:]
        end
    end
end
```


Technische Universiteit
Eindhoven
University of Technology

Routing


```
proc R(chan? cow a; list chan! cow b):
    cow x;
    list(1000) int dest;

    for i in range(1000):
        if i < 164:
            dest[i] = 0;
        elif i < 319:
            dest[i] = 1;
        ...
    end;

    while true:
        a?x;
        x.stat = dest[sample uniform(0, 1000)];
        b[x.stat]!x
    end
end
```

Model Dairy Barn


```
model DairyBarn():
chan cow a, c, d;
    list(5) chan cow b;

run G(a, 10),
        L(a, d, 5.0),
        R(d, b),
        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
```

Multiple visits to work stations

- 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 $E(R_{mr})$

Multiple visits to work stations

Define

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

Then

$$
\begin{aligned}
E(S_{mr}(k)) &= \sum_{s=1}^{n_m} \rho_{ms}(k-1)E(R_{ms}) + \sum_{s=1}^{n_m}(E(L_{ms}(k-1)) - \rho_{ms}(k-1))E(B_{ms}) \\
&\quad + E(B_{mr})
\end{aligned}
$$

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

$$
\begin{aligned}
\Lambda_{mr}(k) &= \frac{k v_{mr}}{\sum_{n=1}^{M}\sum_{s=1}^{n_m} v_{ns}E(S_{ns}(k))} \\
E(L_{mr}(k)) &= \Lambda_{mr}(k)E(S_{mr}(k))
\end{aligned}
$$

Closed multi-class networks

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

This network is also called Closed multi-class Jackson network

Closed multi-class networks

States of network $(\underline{k}_1, \ldots, \underline{k}_M)$ where

- $\underline{k}_m = (k_{m1}, \ldots, k_{mR})$ 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

$$v_{mr} = \sum_{n=1}^{M} v_{nr} p_{nm}^{r}, \qquad m = 1, 2, \ldots, M.$$

Closed multi-class networks

Jackson's miracle

$$p(\underline{k}) = C p_1(\underline{k}_1) p_2(\underline{k}_2) \cdots p_M(\underline{k}_M),$$

where C is normalizing constant

If $c_m = 1$

$$p_m(\underline{k}_m) = \frac{(k_{m1} + k_{m2} + \cdots + k_{mR})!}{k_{m1}! k_{m2}! \cdots k_{mR}!} \left(\frac{v_{m1}}{\mu_m}\right)^{k_{m1}} \left(\frac{v_{m2}}{\mu_m}\right)^{k_{m2}} \cdots \left(\frac{v_{mR}}{\mu_m}\right)^{k_{mR}}$$

If $c_m > 1$

$$p_m(\underline{k}_m) = \frac{(k_{m1} + k_{m2} + \cdots + k_{mR})!}{k_{m1}! k_{m2}! \cdots k_{mR}!} \frac{v_{m1}^{k_{m1}} v_{m2}^{k_{m2}} \cdots v_{mR}^{k_{mR}}}{\mu_m(1)\mu_m(2) \cdots \mu_m(k_{m1} + k_{m2} + \cdots + k_{mR})}$$

where $\mu_m(k) = \min(k, c_m)\mu_m$

Arrival theorem

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 $\underline{N} - \underline{e}_r$

Mean value analysis

Define for network with population $\underline{N}$

$$\begin{aligned} E(S_{mr}(\underline{N})) &= \text{mean production lead time at work station } m \text{ for type } r \text{ job} \\ \Lambda_{mr}(\underline{N}) &= \text{throughput of type } r \text{ jobs of station } m \\ E(L_{mr}(\underline{N})) &= \text{mean number of type } r \text{ jobs in station } m \end{aligned}$$

In single-server network

$$\begin{aligned} E(S_{mr}(\underline{N})) &= \sum_{s=1}^{r} E(L_{ms}(\underline{N} - \underline{e}_r))\frac{1}{\mu_m} + \frac{1}{\mu_m} \\ \Lambda_{mr}(\underline{N}) &= \frac{N_r v_{mr}}{\sum_{n=1}^{M} v_{nr} E(S_{nr}(\underline{N}))} \\ E(L_{mr}(\underline{N})) &= \Lambda_{mr}(\underline{N}) E(S_{mr}(\underline{N})) \end{aligned}$$

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

Recursion over population vector $\underline{N}$, starting from $\underline{k} = \underline{0}$ to $\underline{k} = \underline{N}$!

Priority stations

Define for network with population $\underline{N}$

$$\begin{aligned}
E(W_{mr}(\underline{N})) &= \text{mean waiting time at work station } m \text{ for type } r \text{ job} \\
\Lambda_{mr}(\underline{N}) &= \text{throughput of type } r \text{ jobs of station } m \\
E(Q_{mr}(\underline{N})) &= \text{mean number of type } r \text{ jobs waiting in station } m
\end{aligned}$$

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

$$\begin{aligned}
E(W_{mr}(\underline{N})) &= \sum_{s=1}^{R} \rho_{ms}(\underline{N} - \underline{e}_r)\frac{1}{\mu_m} + \sum_{s=1}^{r} E(Q_{ms}(\underline{N} - \underline{e}_r))\frac{1}{\mu_m} \\
&\quad + \sum_{s=1}^{r-1} \Lambda_{ms}(\underline{N} - \underline{e}_r) E(W_{mr}(\underline{N}))\frac{1}{\mu_m}
\end{aligned}$$

where $\rho_{mr}(\underline{N}) = \Lambda_{mr}(\underline{N})\frac{1}{\mu_m}$

Fixed-point modeling

Breaking the recursion:

Assume jumping type r job sees the system in equilibrium with population $\underline{N}$ (instead of $\underline{N} - \underline{e}_r$)

In FCFS single server station m

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

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

Fixed-point modeling

Breaking the recursion:

Assume jumping type r job sees the system in equilibrium with population $\underline{N}$ (instead of $\underline{N} - \underline{e}_r$)

In FCFS single server station m

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

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

To avoid self queueing

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

Fixed-point modeling

$3MR$ equations for $3MR$ unknowns $E(S_{mr}(\underline{N}))$, $\Lambda_{mr}(\underline{N})$ and $E(L_{mr}(\underline{N}))$

$$
\begin{aligned}
E(S_{mr}(\underline{N})) &= \sum_{s \neq r} E(L_{ms}(\underline{N}))\frac{1}{\mu_m} + \frac{N_r - 1}{N_r} E(L_{mr}(\underline{N}))\frac{1}{\mu_m} + \frac{1}{\mu_m} \\
\Lambda_{mr}(\underline{N}) &= \frac{N_r v_{mr}}{\sum_{n=1}^{M} v_{nr} E(S_{nr}(\underline{N}))} \\
E(L_{mr}(\underline{N})) &= \Lambda_{mr}(\underline{N}) E(S_{mr}(\underline{N}))
\end{aligned}
$$

Solution by successive substitutions

Wrapping up: Examination

- Weekly (8) take home (individual) assignments
- Best 7 out of 8 take home assignments count (40%)
- Final assignment, done in groups of two (60%)
- Send email to iadan@tue.nl to inform about group composition
- Final assignment will be returned by mail
- Due date for final assignment is September 1, send report as pdf
- Report: Present your work clearly (assumptions, analysis, results, etc.) and try to keep size of report limited, to say 3-4 pages
- Appointments will be scheduled: individual evaluation, which is mix of report evaluation and oral exam