MIT 2.852
Manufacturing Systems Analysis
Lecture 10–12

Transfer Lines – Long Lines

Stanley B. Gershwin

http://web.mit.edu/manuf-sys

Massachusetts Institute of Technology

Spring, 2010

Long Lines

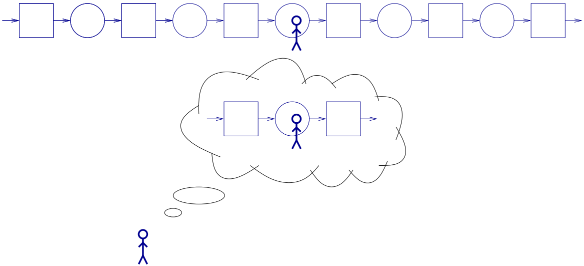
- Difficulty:
  - No simple formula for calculating production rate or inventory levels.
  - State space is too large for exact numerical solution.
    - If all buffer sizes are N and the length of the line is k, the number of states is $S = 2^k(N+1)^{k-1}$.
    - if $N = 10$ and $k = 20$, $S = 6.41 \times 10^{25}$.
  - *Decomposition* seems to work successfully.

Decomposition — Concept

Decomposition works for many kinds of systems, and extending it is an active research area.

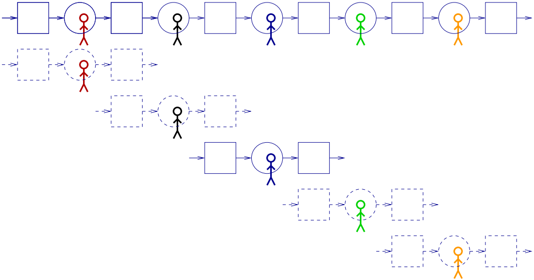
- We start with deterministic processing time lines.
- Then we extend decomposition to other lines.
- Then we extend it to assembly/disassembly systems without loops.
- Then we look at systems with loops.
- Etc., etc. if there is time.

Decomposition — Concept



- Conceptually: put an observer in a buffer, and tell him that he is in the buffer of a two-machine line.
- Question: *What would the observer see, and how can he be convinced he is in a two-machine line?*

Decomposition — Concept



- Decomposition breaks up systems and then reunites them.
- *Construct all the two-machine lines.*

Decomposition — Concept

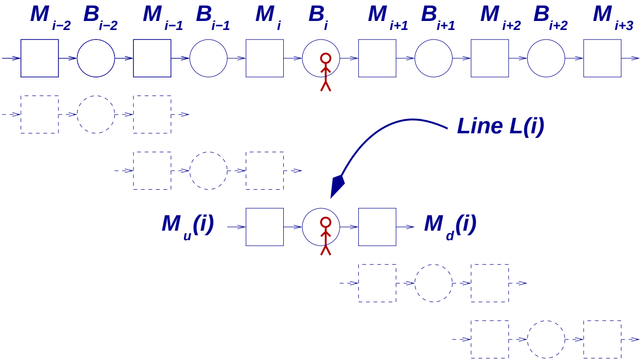
- Evaluate the performance measures (production rate, average buffer level) of each two-machine line, and use them for the real line.
- This is an *approximation;* the behavior of the flow in the buffer of a two-machine line is not exactly the same as the behavior of the flow in a buffer of a long line.
- The two-machine lines are sometimes called *building blocks.*

Decomposition — Concept

- Consider an observer in Buffer B_i.
  - Imagine the material flow process that the observer sees *entering* and the material flow process that the observer sees *leaving* the buffer.
- We construct a two-machine line $L(i)$
  - (ie, we find machines $M_u(i)$ and $M_d(i)$ with parameters $r_u(i)$, $p_u(i)$, $r_d(i)$, $p_d(i)$, and $N(i) = N_i$)

  such that an observer in its buffer will see almost the same processes.
- The parameters are chosen as functions of the behaviors of the *other* two-machine lines.

Decomposition — Concept



Decomposition — Concept

There are $4(k-1)$ unknowns for the deterministic processing time line:

$$r_u(1), p_u(1), r_d(1), p_d(1),$$
$$r_u(2), p_u(2), r_d(2), p_d(2),$$
$$...,$$
$$r_u(k-1), p_u(k-1), r_d(k-1), p_d(k-1)$$

Therefore, we need

- $4(k-1)$ equations, and
- an algorithm for solving those equations.

Decomposition Equations
Overview

The decomposition equations relate $r_u(i)$, $p_u(i)$, $r_d(i)$, and $p_d(i)$ to behavior in the real line and in other two-machine lines.

- *Conservation of flow,* equating all production rates.
- *Flow rate/idle time,* relating production rate to probabilities of starvation and blockage.
- *Resumption of flow,* relating $r_u(i)$ to upstream events and $r_d(i)$ to downstream events.
- *Boundary conditions,* for parameters of $M_u(1)$ and $M_d(k-1)$.

Decomposition Equations
Overview

- All the quantities in all these equations are
  - specified parameters, or
  - unknowns, or
  - functions of parameters or unknowns derived from the two-machine line analysis.
- This is a set of $4(k-1)$ equations.

Decomposition Equations
Overview

Notation convention:

- Items that pertain to two-machine line $L(i)$ will have i in parentheses. Example: $r_u(i)$.
- Items that pertain to the real line L will have i in the subscript. Example: r_i.

Decomposition Equations
Conservation of Flow

$$E(i) = E(1), i = 2, \ldots, k-1.$$

- Recall that $E(i)$ is a function of the unknowns $r_u(i)$, $p_u(i)$, $r_d(i)$, and $p_d(i)$.
- (It is also a function of $N(i)$, but $N(i)$ is known.)
- We know how to evaluate it easily, but we don't have a simple expression for it.

This is a set of $k-2$ equations.

Decomposition Equations
Flow Rate-Idle Time

$$E_i = e_i \text{ prob } [n_{i-1} > 0 \text{ and } n_i < N_i]$$

where

$$e_i = \frac{r_i}{r_i + p_i}$$

Problem:

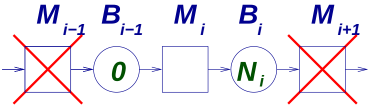
- This expression involves a joint probability of *two* buffers taking certain values at the same time.
- But we only know how to evaluate two-machine, one-buffer lines, so we only know how to calculate the probability of *one* buffer taking on a certain value at a time.

Decomposition Equations
Flow Rate-Idle Time

Observation:

$$\text{prob } (n_{i-1} = 0 \text{ and } n_i = N_i) \approx 0.$$

Reason:



The only way to have $n_{i-1} = 0$ and $n_i = N_i$ is if

- M_{i-1} is down or starved for a long time
- *and* M_i is up
- *and* M_{i+1} is down or blocked for a long time
- *and* to have exactly N_i parts in the two buffers.

Decomposition Equations
Flow Rate-Idle Time

Then

$$\text{prob } [n_{i-1} > 0 \text{ and } n_i < N_i]$$

$$= \text{ prob } [\text{NOT } \{n_{i-1} = 0 \text{ or } n_i = N_i\}]$$

$$= 1 - \text{ prob } [n_{i-1} = 0 \text{ or } n_i = N_i]$$

$$= 1 - \{ \text{ prob } (n_{i-1} = 0) + \text{ prob } (n_i = N_i) - \text{ prob } (n_{i-1} = 0 \text{ and } n_i = N_i)\}$$

$$\approx 1 - \{ \text{ prob } (n_{i-1} = 0) + \text{ prob } (n_i = N_i)\}$$

Therefore

$$E_i \approx e_i \left[1 - \text{ prob } (n_{i-1} = 0) - \text{ prob } (n_i = N_i)\right]$$

Note that

$$\text{prob } (n_{i-1} = 0) = p_s(i-1); \quad \text{prob } (n_i = N_i) = p_b(i)$$

Two of the FRIT relationships in lines $L(i-1)$ and $L(i)$ are

$$E(i) = e_u(i) \left[1 - p_b(i)\right]; \quad E(i-1) = e_d(i-1) \left[1 - p_s(i-1)\right]$$

Decomposition Equations
Flow Rate-Idle Time

or,

$$p_s(i-1) = 1 - \frac{E(i-1)}{e_d(i-1)}; \quad p_b(i) = 1 - \frac{E(i)}{e_u(i)}$$

so (replacing $\approx$ with $=$),

$$E_i = e_i \left[1 - \left\{ 1 - \frac{E(i-1)}{e_d(i-1)} \right\} - \left\{ 1 - \frac{E(i)}{e_u(i)} \right\} \right]$$

The goal is to have $E = E_i = E(i-1) = E(i)$, so

$$E(i) = e_i \left[1 - \left\{ 1 - \frac{E(i)}{e_d(i-1)} \right\} - \left\{ 1 - \frac{E(i)}{e_u(i)} \right\} \right]$$

Decomposition Equations
Flow Rate-Idle Time

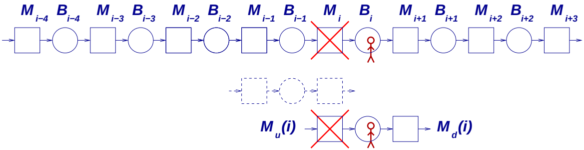
Since

$$e_d(i-1) = \frac{r_d(i-1)}{p_d(i-1) + r_d(i-1)}; \qquad e_u(i) = \frac{r_u(i)}{p_u(i) + r_u(i)},$$

we can write

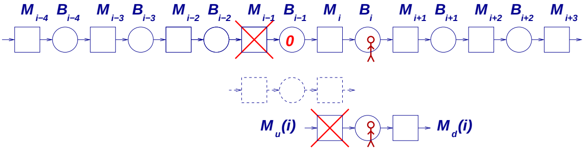
$$\boxed{\frac{p_d(i-1)}{r_d(i-1)} + \frac{p_u(i)}{r_u(i)} = \frac{1}{E(i)} + \frac{1}{e_i} - 2, i = 2, \ldots, k-1}$$

This is a set of $k-2$ equations.



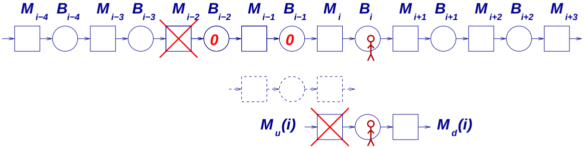
When the observer sees $M_u(i)$ down, M_i may actually be down...

Decomposition Equations
Resumption of Flow



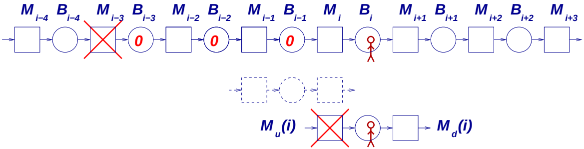
... or, M_{i-1} may be down and B_{i-1} may be empty, ...

Decomposition Equations
Resumption of Flow



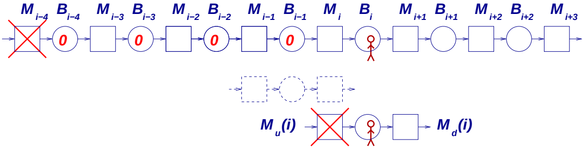
... or M_{i-2} may be down and B_{i-1} and B_{i-2} may be empty, ...

Decomposition Equations
Resumption of Flow



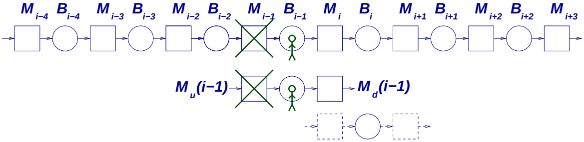
... or M_{i-3} may be down and B_{i-1} and B_{i-2} and B_{i-3} may be empty, ...

Decomposition Equations
Resumption of Flow



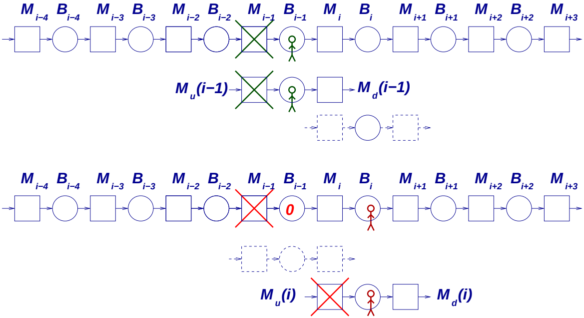
... etc.

Decomposition Equations
Resumption of Flow



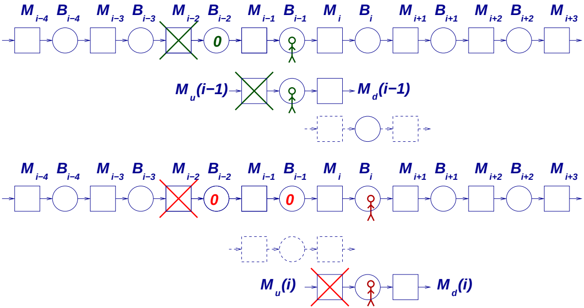
Similarly for the observer in B_{i-1}.

Decomposition Equations
Resumption of Flow

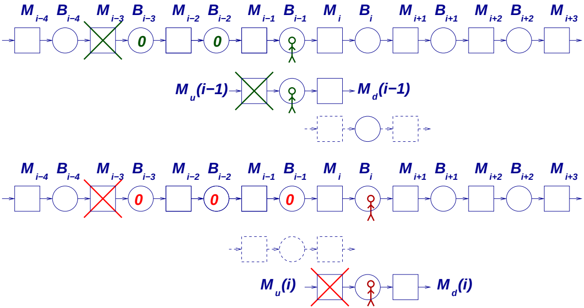


Comparison

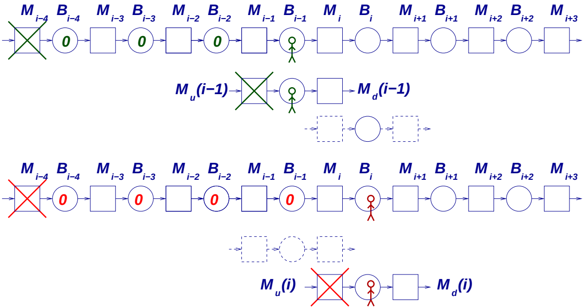
Decomposition Equations
Resumption of Flow



Decomposition Equations
Resumption of Flow

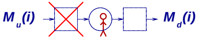


Decomposition Equations
Resumption of Flow

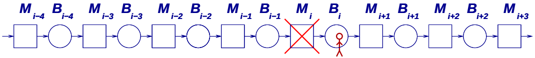


Decomposition Equations
Resumption of Flow

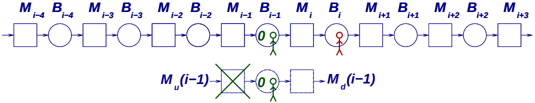
That is, when the Line $L(i)$ observer sees a failure in $M_u(i)$,



- *<u>either</u> real machine M_i is down,*



- *<u>or</u> Buffer B_{i-1} is empty and the Line $L(i-1)$ observer sees a failure in $M_u(i-1)$.*



Note that these two events are mutually exclusive. *Why?*

Decomposition Equations
Resumption of Flow

Also, for the Line $L(j)$ observer to see $M_u(j)$ up, M_j must be up and B_{j-1} must be non-empty. Therefore,

$$\{\alpha_u(j,\tau)=1\} \Longleftrightarrow \{\alpha_j(\tau)=1\} \text{ and } \{n_{j-1}(\tau-1)>0\}$$

$$\{\alpha_u(j,\tau)=0\} \Longleftrightarrow \{\alpha_j(\tau)=0\} \text{ or } \{n_{j-1}(\tau-1)=0\}$$

Decomposition Equations
Resumption of Flow

Then

$$r_u(i) = \text{prob}\left[\alpha_u(i, t+1) = 1 \mid \alpha_u(i, t) = 0\right]$$

$$= \text{prob}\left[\{\alpha_i(t+1) = 1\} \text{ and } \{n_{i-1}(t) > 0\} \middle| \{\alpha_i(t) = 0\} \text{ or } \{n_{i-1}(t-1) = 0\}\right]$$

Decomposition Equations
Resumption of Flow

To express $r_u(i)$ in terms of quantities we know or can find, we have to simplify prob $(U|V \text{ or } W)$, where

$$U = \{\alpha_i(t+1) = 1\} \text{ and } \{n_{i-1}(t) > 0\}$$

$$V = \{\alpha_i(t) = 0\}$$

$$W = \{n_{i-1}(t-1) = 0\}$$

Important: V and W are disjoint.

$$\text{prob } (V \text{ and } W) = 0.$$

$$\text{prob}\ (U|V \text{ or } W) = \frac{\text{prob}\ (U \text{ and } (V \text{ or } W))}{\text{prob}\ (V \text{ or } W)}$$

$$= \frac{\text{prob}\ ((U \text{ and } V) \text{ or } (U \text{ and } W))}{\text{prob}\ (V \text{ or } W)}$$

$$= \frac{\text{prob}\ (U \text{ and } V)}{\text{prob}\ (V \text{ or } W)} + \frac{\text{prob}\ (U \text{ and } W)}{\text{prob}\ (V \text{ or } W)}$$

$$= \frac{\text{prob}\ (U|V)\text{prob}\ (V)}{\text{prob}\ (V \text{ or } W)} + \frac{\text{prob}\ (U|W)\text{prob}\ (W)}{\text{prob}\ (V \text{ or } W)}$$

$$= \text{prob } (U|V)\frac{\text{prob } (V)}{\text{prob } (V \text{ or } W)} + \text{prob } (U|W)\frac{\text{prob } (W)}{\text{prob } (V \text{ or } W)}$$

Note that

$$\text{prob } (V|V \text{ or } W) = \frac{\text{prob } (V \text{ and } (V \text{ or } W))}{\text{prob } (V \text{ or } W)} = \frac{\text{prob } (V)}{\text{prob } (V \text{ or } W)}$$

so

$$\text{prob } (U|V \text{ or } W) = \text{prob } (U|V)\text{prob } (V|V \text{ or } W) + \text{prob } (U|W)\text{prob } (W|V \text{ or } W).$$

Then, if we plug U, V, and W from Slide 33 into this, we get

$$r_u(i) = A(i-1)X(i) + B(i)X'(i), i = 2, \ldots, k-1$$

where

$$\begin{aligned} A(i-1) = & \quad \text{prob } (U|W) \\ = & \quad \text{prob } \left[n_{i-1}(t) > 0 \text{ and } \alpha_i(t+1) = 1 \middle| \right. \\ & \qquad \left. n_{i-1}(t-1) = 0 \right], \end{aligned}$$

Decomposition Equations
Resumption of Flow

$$X(i) = \text{prob}\,(W|V \text{ or } W)$$
$$= \text{prob}\,\left[n_{i-1}(t-1) = 0 \middle| n_{i-1}(t-1) = 0 \text{ or } \alpha_i(t) = 0\right],$$

$$B(i) = \text{prob}\,(U|V)$$
$$= \text{prob}\,\left[n_{i-1}(t) > 0 \text{ and } \alpha_i(t+1) = 1 \mid \alpha_i(t) = 0\right],$$

$$X'(i) = \text{prob}\,(V|V \text{ or } W)$$
$$= \text{prob}\,\left[\alpha_i(t) = 0 \mid \{n_{i-1}(t-1) = 0 \text{ or } \alpha_i(t) = 0\}\right].$$

Decomposition Equations
Resumption of Flow

To evaluate

$$A(i-1) = \text{prob}\ \left[n_{i-1}(t) > 0 \text{ and } \alpha_i(t+1) = 1 \middle| n_{i-1}(t-1) = 0 \right] :$$

Note that

- For Buffer $i-1$ to be empty at time $t-1$, Machine M_i must be up at time $t-1$ and also at time t. It must have been up in order to empty the buffer, and it must stay up because it cannot fail. Therefore $\alpha_i(t) = 1$.
- For Buffer $i-1$ to be non-empty at time t after being empty at time $t-1$, it must have gained 1 part. For it to gain a part when $\alpha_i(t) = 1$, M_i must not have been working (because it was previously starved). Therefore, M_i could not have failed and $A(i-1)$ can therefore be written

$$A(i-1) = \text{prob}\ \left[n_{i-1}(t) > 0 \middle| n_{i-1}(t-1) = 0 \right]$$

Decomposition Equations
Resumption of Flow

$$A(i-1) = \text{prob}\left[n_{i-1}(t) > 0 \middle| n_{i-1}(t-1) = 0\right]$$

- For Buffer $i-1$ to be empty, M_{i-1} must be down or starved. For M_{i-1} to be starved, M_{i-2} must be down or starved, etc. Therefore, saying M_{i-1} is down or starved is equivalent to saying $M_u(i-1)$ is down. That is, if $n_{i-1}(t-1) = 0$ then $\alpha_u(i-1, t-1) = 0$.
- Conversely, for Buffer $i-1$ to be non-empty, M_{i-1} must not be down or starved. That is, if $n_{i-1}(t) > 0$, then $\alpha_u(i-1, t) = 1$.

Therefore,

$$A(i-1) = \text{prob}\left[\alpha_u(i-1,t) = 1 \middle| \alpha_u(i-1, t-1) = 0\right] = r_u(i-1)$$

Decomposition Equations
Resumption of Flow

Similarly,

$$B(i) = \text{prob } [n_{i-1}(t) > 0 \text{ and } \alpha_i(t+1) = 1 \mid \alpha_i(t) = 0]$$

Note that if $\alpha_i(t) = 0$, we must have $n_{i-1}(t) > 0$. Therefore

$$B(i) = \text{prob } [\alpha_i(t+1) = 1 \mid \alpha_i(t) = 0] \,,$$

or,

$$B(i) = r_i$$

so

$$r_u(i) = r_u(i-1)X(i) + r_i X'(i),$$

Decomposition Equations
Resumption of Flow

Interpretation so far:

- $r_u(i)$, the probability that $M_u(i)$ goes from down to up, is
  - r_i times the probability that $M_u(i)$ is down because M_i is down
  - plus $r_u(i-1)$ times the probability that $M_u(i)$ is down because $M_u(i-1)$ is down and B_{i-1} is empty.

Decomposition Equations
Resumption of Flow

$X(i)$= the probability that $M_u(i)$ is down because $M_u(i-1)$ is down and B_{i-1} is empty;

$X'(i)$ = the probability that $M_u(i)$ is down because M_i is down.

Since these are the only two ways that $M_u(i)$ can be down,

$$X'(i) = 1 - X(i)$$

Decomposition Equations
Resumption of Flow

$$X(i) = \text{prob}\ \left[n_{i-1}(t-1) = 0 \middle| n_{i-1}(t-1) = 0 \text{ or } \alpha_i(t) = 0 \right]$$

$$= \frac{\text{prob}\ [n_{i-1}(t-1) = 0 \text{ and } \{n_{i-1}(t-1) = 0 \text{ or } \alpha_i(t) = 0\}]}{\text{prob}\ [n_{i-1}(t-1) = 0 \text{ or } \alpha_i(t) = 0]}$$

$$= \frac{\text{prob}\ [n_{i-1}(t-1) = 0]}{\text{prob}\ [n_{i-1}(t-1) = 0 \text{ or } \alpha_i(t) = 0]}$$

$$= \frac{p_s(i-1)}{\text{prob}\ [n_{i-1}(t-1) = 0 \text{ or } \alpha_i(t) = 0]}$$

Decomposition Equations
Resumption of Flow

To analyze the denominator, note

- $\{n_{i-1}(t-1) = 0 \text{ or } \alpha_i(t) = 0\} = \{\alpha_u(i) = 0\}$ by definition;
- prob $[n_{i-1}(t-1) = 0 \text{ or } \alpha_i(t) = 0] \approx$
  prob $[\{n_{i-1}(t-1) = 0 \text{ or } \alpha_i(t) = 0\}$ and $n_i(t-1) < N_i]$ because
  prob $[n_{i-1}(t-1) = 0$ and $n_i(t-1) = N_i] \approx 0$

so the denominator is, approximately,

$$\text{prob } [\alpha_u(i) = 0 \text{ and } n_i(t-1) < N_i]$$

Recall that this is equal to

$$\frac{p_u(i)}{r_u(i)} \text{prob } [\alpha_u(i) = 1 \text{ and } n_i(t-1) < N_i] = \frac{p_u(i)}{r_u(i)} E(i)$$

Therefore,

$$X(i) = \frac{p_s(i-1)r_u(i)}{p_u(i)E(i)}$$

and

$$\boxed{r_u(i) = r_u(i-1)X(i) + r_i(1 - X(i)), i = 2, \ldots, k-1}$$

This is a set of $k - 2$ equations.

Decomposition Equations
Resumption of Flow

By the same logic,

$$r_d(i-1) = r_d(i)Y(i) + r_i(1 - Y(i)), i = 2, \ldots, k-1$$

where

$$Y(i) = \frac{p_b(i) r_d(i-1)}{p_d(i-1)E(i-1)}.$$

This is a set of $k - 2$ equations.

We now have $4(k-2) = 4k - 8$ equations.

Decomposition Equations
Boundary Conditions

$M_d(1)$ is the same as M_1 and $M_d(k-1)$ is the same as M_k. Therefore

$$
\begin{aligned}
r_u(1) &= r_1 \\
p_u(1) &= p_1 \\
r_d(k-1) &= r_k \\
p_d(k-1) &= p_k
\end{aligned}
$$

This is a set of 4 equations.

We now have $4(k-1)$ equations in $4(k-1)$ unknowns $r_u(i)$, $p_u(i)$, $r_d(i)$, $p_d(i)$, $i = 1, ..., k-1$.

Decomposition Equations

Algorithm

FRIT: $$\frac{p_d(i-1)}{r_d(i-1)} + \frac{p_u(i)}{r_u(i)} = \frac{1}{E(i)} + \frac{1}{e_i} - 2$$

Upstream equations:

$$r_u(i) = r_u(i-1)X(i) + r_i(1-X(i)); \qquad X(i) = \frac{p_s(i-1)r_u(i)}{p_u(i)E(i)}$$

$$p_u(i) = r_u(i)\left(\frac{1}{E(i)} + \frac{1}{e_i} - 2 - \frac{p_d(i-1)}{r_d(i-1)}\right)$$

Downstream equations:

$$r_d(i) = r_d(i+1)Y(i+1) + r_{i+1}(1-Y(i+1)); \; Y(i+1) = \frac{p_b(i+1)r_d(i)}{p_d(i)E(i)}.$$

$$p_d(i) = r_d(i)\left(\frac{1}{E(i+1)} + \frac{1}{e_{i+1}} - 2 - \frac{p_u(i+1)}{r_u(i+1)}\right)$$

Decomposition Equations
Algorithm

We use the conservation of flow conditions by modifying these equations.

Modified upstream equations:

$$r_u(i) = r_u(i-1)X(i) + r_i(1 - X(i)); \qquad X(i) = \frac{p_s(i-1)r_u(i)}{p_u(i)E(i-1)}$$

$$p_u(i) = r_u(i)\left(\frac{1}{E(i-1)} + \frac{1}{e_i} - 2 - \frac{p_d(i-1)}{r_d(i-1)}\right)$$

Modified downstream equations:

$$r_d(i) = r_d(i+1)Y(i+1) + r_{i+1}(1 - Y(i+1)); \; Y(i+1) = \frac{p_b(i+1)r_d(i)}{p_d(i)E(i+1)}.$$

$$p_d(i) = r_d(i)\left(\frac{1}{E(i+1)} + \frac{1}{e_{i+1}} - 2 - \frac{p_u(i+1)}{r_u(i+1)}\right)$$

Possible Termination Conditions:

- $|E(i) - E(1)| < \epsilon$ for $i = 2, ..., k-1$, or
- The change in *each* $r_u(i)$, $p_u(i)$, $r_d(i)$, $p_d(i)$ parameter, $i = 1, ..., k-1$ is less than ϵ, or
- etc.

Decomposition Equations

Algorithm

DDX algorithm : due to Dallery, David, and Xie (1988).

1. Guess the downstream parameters of $L(1)$ $(r_d(1), p_d(1))$. Set $i = 2$.
2. Use the modified upstream equations to obtain the upstream parameters of $L(i)$ $(r_u(i), p_u(i))$. Increment i.
3. Continue in this way until $L(k-1)$. Set $i = k - 2$.
4. Use the modified downstream equations to obtain the downstream parameters of $L(i)$. Decrement i.
5. Continue in this way until $L(1)$.
6. Go to Step 2 or terminate.

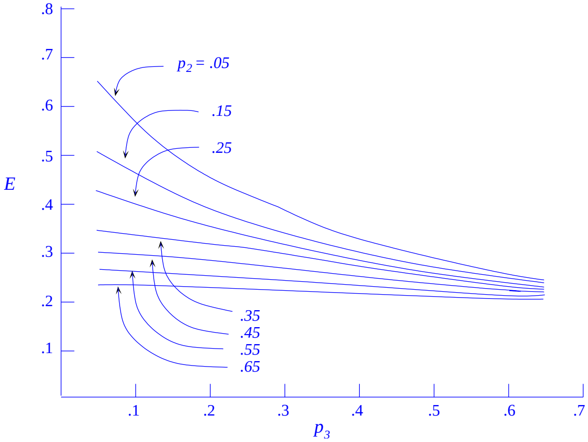
Is the decomposition exact? *NO,* because

1. The behavior of the flow in the buffer of a two-machine line is not exactly the same as the behavior of the flow in a buffer of a long line.

2. prob $[n_{i-1}(t-1) = 0 \text{ and } n_i(t-1) = N_i] \approx 0$

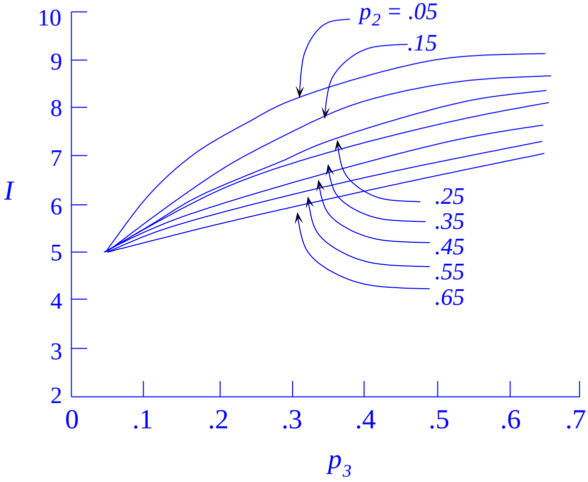
Question: When will this work well, and when will it work badly?

Three-machine line – production rate.



$r_1 = r_2 = r_3 = .2$
$p_1 = .05$
$N_1 = N_2 = 5$

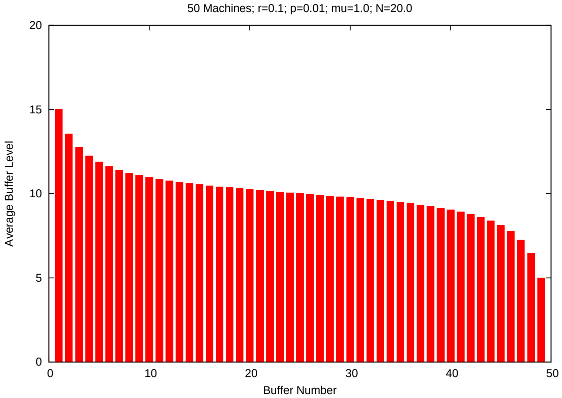
Three-machine line – total average inventory



$r_1 = r_2 = r_3 = .2$

$p_1 = .05$

$N_1 = N_2 = 5$

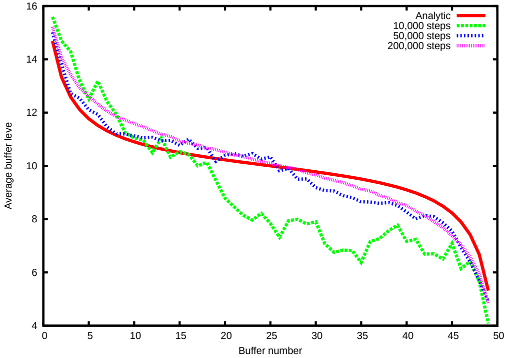


Distribution of material in a line with identical machines and buffers.

Explain the shape.

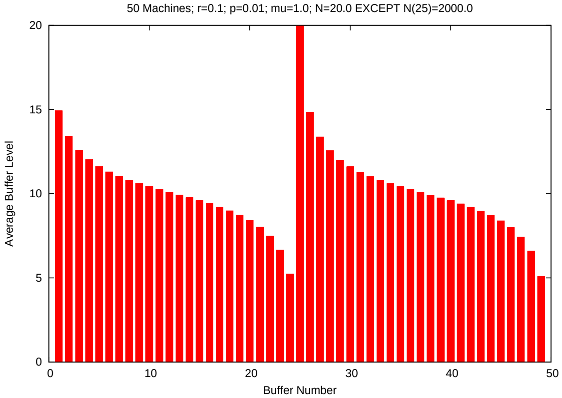
Analytical vs simulation

Time steps	*Decomp*	10,000	50,000	200,000
Production rate	0.786	0.740	0.751	0.750



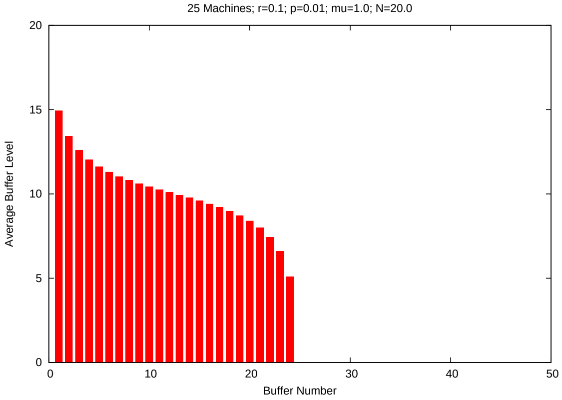
(Not the same line as in Slide 55.)

Examples
Long lines



Same as Slide 55 except that Buffer 25 is now huge.

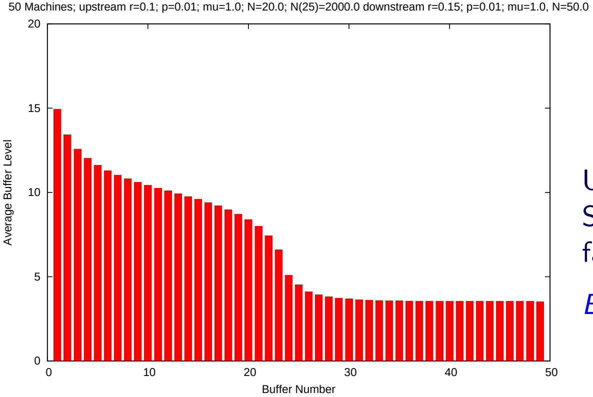
Explain the shape.



Upstream half of Slide 57.

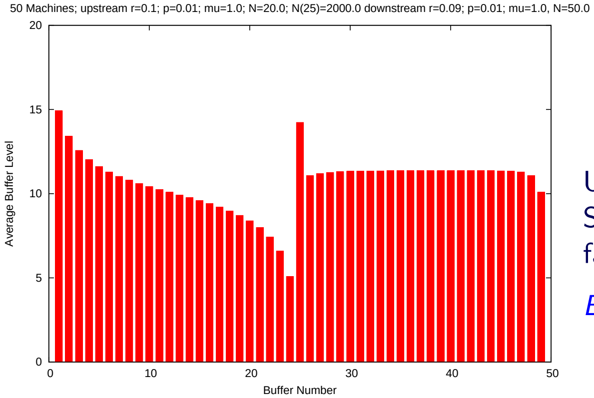
Explain the shape.

Examples
Long lines



Upstream same as Slide 58; downstream faster.

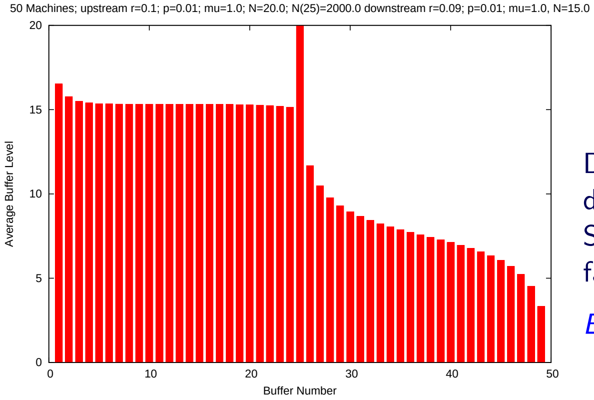
Explain the shape.



Upstream same as Slide 58; downstream faster.

Explain the shape.

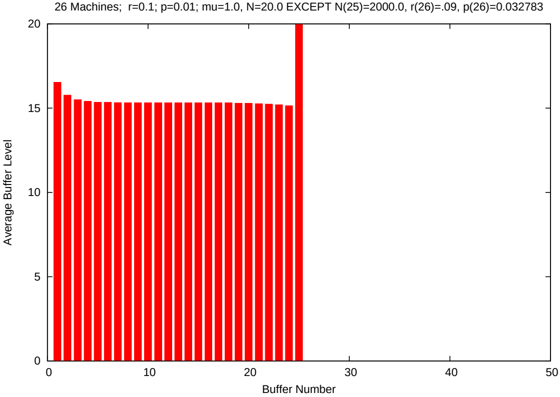
Examples
Long lines



Downstream same as downstream half of Slide 57; upstream faster.

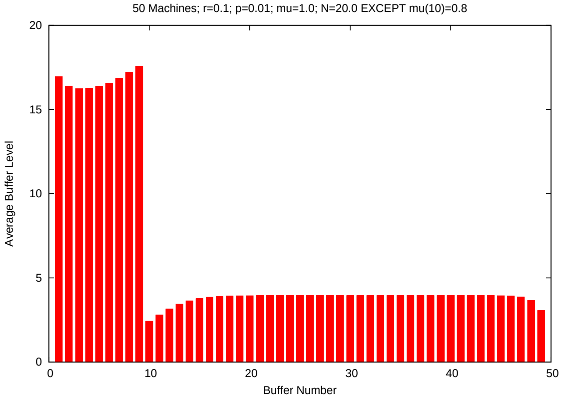
Explain the shape.

Examples
Long lines



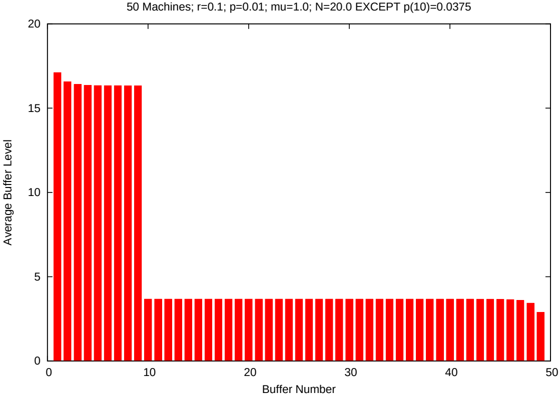
Same as upstream half of Slide 61 *except for Machine 26.*

Explain the shape. How was Machine 26 chosen?



Operation time bottleneck. Identical machines and buffers, except for M_{10}.

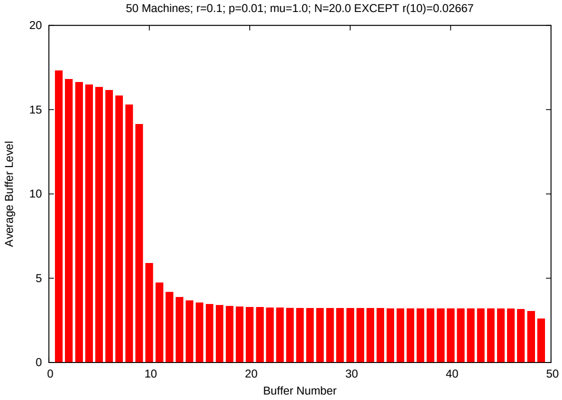
Explain the shape.



Failure time bottleneck.

Explain the shape.

Examples
Long lines — Bottlenecks



Repair time bottleneck.

Explain the shape.

Infinitely long lines with identical machines and buffers

$$\left.\begin{array}{l} r_i = r \\ p_i = p \\ N_i = N \end{array}\right\} \text{ for each } i, -\infty < i < \infty.$$

The observer in each buffer sees exactly the same behavior. Consequently, the decomposed pseudo-machines are all identical and symmetric. For each i,

$$\begin{array}{l} r_u(i) = r_u(i-1) = r_d(i) = r_d(i-1) \\ p_u(i) = p_u(i-1) = p_d(i) = p_d(i-1). \end{array}$$

Resumption of flow says

$$r_u(i) = r_u(i-1)X(i) + r_i(1 - X(i))$$
$$r_u = r_u X + r(1 - X)$$

so $r_u(i) = r_d(i) = r$.

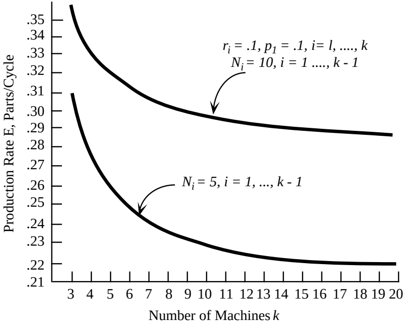
FRIT says

$$\frac{p_d(i-1)}{r_d(i-1)} + \frac{p_u(i)}{r_u(i)} = \frac{1}{E(i)} + \frac{1}{e_i} - 2$$

$$\frac{2p_u}{r} = \frac{1}{E} + \frac{1}{e} - 2$$

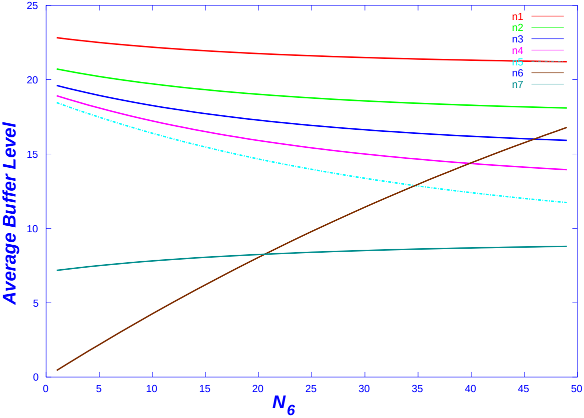
Examples
Infinitely long lines

In the last equation, p_u is unknown and E is a function of p_u. This is one equation in one unknown.



Examples

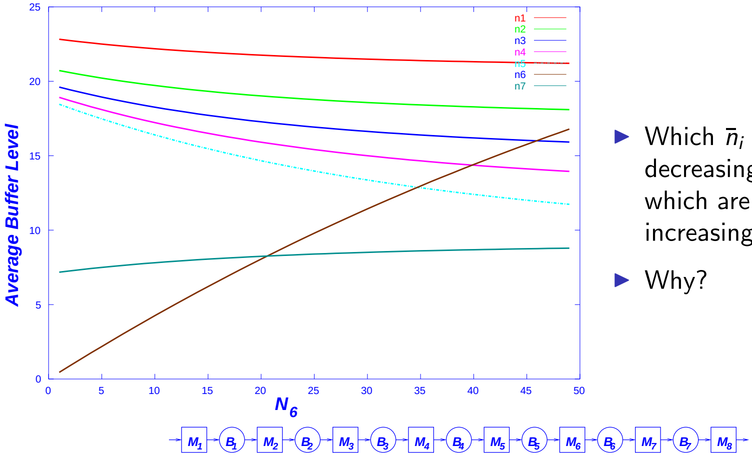
Effect of *one* buffer size on *all* buffer levels



Continuous material model.

- Eight-machine, seven-buffer line.
- For each machine, $r = .075$, $p = .009$, $\mu = 1.2$.
- For each buffer *(except Buffer 6)*, $N = 30$.

Examples
Effect of *one* buffer size on *all* buffer levels

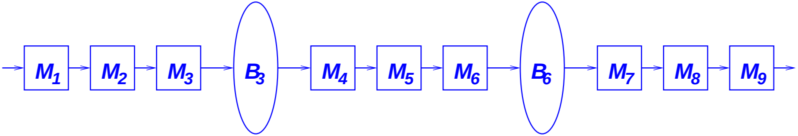


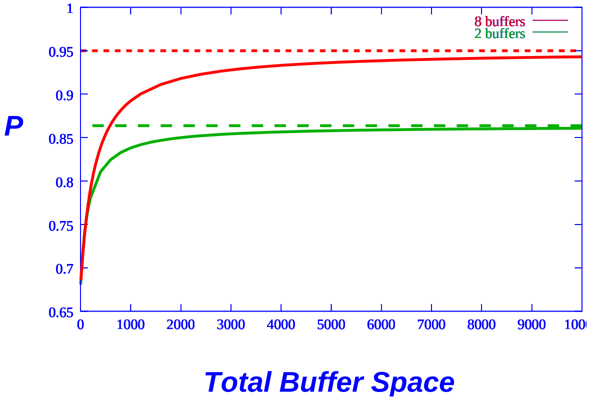
- Which $\bar{n}_i$ are decreasing and which are increasing?
- Why?

Which has a higher production rate?

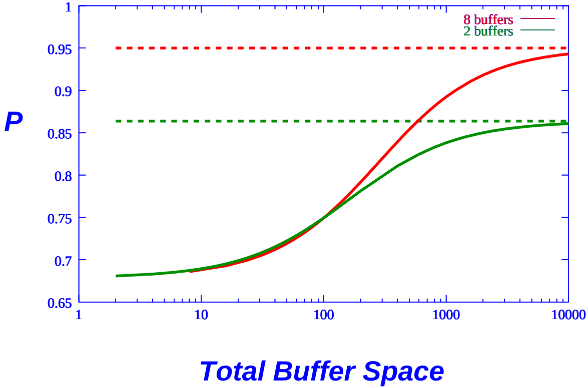
- 9-Machine line with two buffering options:
  - 8 buffers equally sized; and

  - 2 buffers equally sized.





- Continuous model; all machines have $r = .019$, $p = .001$, $\mu = 1$.
- What are the asymptotes?
- Is 8 buffers *always* faster?



- *Is 8 buffers always faster?*
- Perhaps not, but difference is not significant in systems with very small buffers.

Long Lines — More Models

Discrete Material Exponential Processing Time *and* Continuous Material Models

- *New issue:* machines may operate at different speeds.
- Blockage and starvation may be caused by differences in machine speeds, not only failures.
- Decomposition of these classes of systems is similar to that of discrete-material, deterministic-processing time lines *except*
  - The two-machine lines have machines with 3 parameters ($r_u(i)$, $p_u(i)$, $\mu_u(i)$; $r_d(i)$, $p_d(i)$, $\mu_d(i)$). More equations — $6(k-1)$ — are therefore needed.
- Exponential decomposition is described in the book in detail; continuous material decomposition was not developed until after book was written.

Long Lines — Exponential Processing Time Model

The observer thinks he is in a two-machine exponential processing time line with parameters

$r_u(i)\delta t =$ probability that $M_u(i)$ goes from down to up in $(t, t+\delta t)$, for small δt;

$p_u(i)\delta t =$ probability that $M_u(i)$ goes from up to down in $(t, t+\delta t)$ if it is not blocked, for small δt;

$\mu_u(i)\delta t =$ probability that a piece flows into B_i in $(t, t+\delta t)$ when $M_u(i)$ is up and not blocked, for small δt;

$r_d(i)\delta t =$ probability that $M_d(i)$ goes from down to up in $(t, t+\delta t)$, for small δt;

$p_d(i)\delta t =$ probability that $M_d(i)$ goes from up to down in $(t, t+\delta t)$ if it is not starved, for small δt;

$\mu_d(i)\delta t =$ probability that a piece flows out of B_i in $(t, t+\delta t)$ when $M_d(i)$ is up and not starved, for small δt.

We have $6(k-1)$ unknowns, so we need $6(k-1)$ equations. They are

- *Interruption of flow* , relating $p_u(i)$ to upstream events and $p_d(i)$ to downstream events,
- *Resumption of flow,*
- *Conservation of flow,*
- *Flow rate/idle time,*
- *Boundary conditions.*

All of these, except for the Interruption of Flow equations, are similar to those of the deterministic processing time case.

Long Lines — Exponential Processing Time Model

Interruption of Flow

The first two sets of equations describe the interruptions of flow caused by machine failures. By definition,

$$p_u(i)\delta t = \text{prob } \left[\alpha_u(i; t+\delta t) = 0 \middle| \alpha_u(i;t) = 1 \text{ and } n_i(t) < N_i \right],$$

or,

$$p_u(i)\delta t = \text{prob } \left[M_u(i) \text{ down at } t+\delta t \middle| M_u(i) \text{ up and } n_i < N_i \text{ at } t \right].$$

Long Lines — Exponential Processing Time Model
Interruption of Flow

We define the events that a pseudo-machine is up or down as follows:

$M_u(i)$ is down if

1. M_i *is down*, or
2. $n_{i-1} = 0$ *and* $M_u(i-1)$ *is down*.

$M_u(i)$ is up for all other states of the transfer line upstream of Buffer B_i. Therefore, $M_u(i)$ is up if

1. M_i *is operational and* $n_{i-1} > 0$, or
2. M_i *is operational*, $n_{i-1} = 0$ *and* $M_u(i-1)$ *is up*.

After a lot of equation manipulation, we get:

$$p_u(i) = p_i + \frac{r_u(i-1)\mathbf{p}(i-1;001)}{E_u(i)}.$$

and similarly,

$$p_d(i) = p_{i+1} + \frac{r_d(i+1)\mathbf{p}(i+1;N10)}{E_d(i)}.$$

in which $\mathbf{p}(i-1;001)$ is the steady state probability that line $L(i-1)$ is in state $(0,0,1)$ and $\mathbf{p}(i+1;N10)$ is the steady state probability that line $L(i+1)$ is in state $(N_{i+1},1,0)$.

Long Lines — Exponential Processing Time Model
Resumption of Flow

$$
\begin{aligned}
r_u(i) &= r_u(i-1)\frac{\mathbf{p}_{i-1}(0,0,1)r_u(i)\mu_u(i)}{p_u(i)P(i)} \\
&\quad + r_i\left(1 - \frac{\mathbf{p}_{i-1}(0,0,1)r_u(i)\mu_u(i)}{p_u(i)P(i)}\right), \\
&\quad i = 2,\cdots,k-1
\end{aligned}
$$

$$
\begin{aligned}
r_d(i) &= r_d(i+1)\frac{\mathbf{p}_{i+1}(N_{i+1},1,0)r_d(i)\mu_d(i)}{p_d(i)P(i)} \\
&\quad + r_{i+1}\left(1 - \frac{\mathbf{p}_{i+1}(N_{i+1},1,0)r_d(i)\mu_d(i)}{p_d(i)P(i)}\right), \\
&\quad i = 1,\cdots,k-2
\end{aligned}
$$

$$P(i) = P(1), i = 2, \ldots, k-1.$$

The flow rate-idle time relationship is, *approximately,*

$$P_i = e_i \mu_i \left(1 - \text{prob}\,[n_{i-1} = 0] - \text{prob}\,[n_i = N_i]\right).$$

which can be transformed into

$$\frac{1}{e_i \mu_i} + \frac{1}{P} = \frac{1}{e_d(i-1)\mu_d(i-1)} + \frac{1}{e_u(i)\mu_u(i)}; \quad i = 2, \ldots, k-1.$$

Long Lines — Exponential Processing Time Model
Flow Rate/Idle Time

For the algorithm, we express it as

$$\mu_u(i) = \frac{1}{e_u(i)} \left\{ \frac{1}{\frac{1}{P(i)} + \frac{1}{e_i \mu_i} - \frac{1}{e_d(i-1)\mu_d(i-1)}} \right\},$$

$$i = 2, \cdots, k-1,$$

$$\mu_d(i) = \frac{1}{e_d(i)} \left\{ \frac{1}{\frac{1}{P(i)} + \frac{1}{e_{i+1} \mu_{i+1}} - \frac{1}{e_u(i+1)\mu_u(i+1)}} \right\},$$

$$i = 1, \cdots, k-2.$$

Long Lines — Exponential Processing Time Model
Boundary Conditions

$M_d(1)$ is the same as M_1 and $M_d(k-1)$ is the same as M_k. Therefore

$$r_u(1) = r_1$$

$$p_u(1) = p_1$$

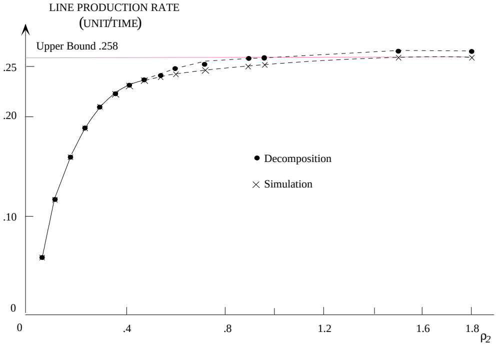
$$\mu_u(1) = \mu_1$$

$$r_d(k-1) = r_k$$

$$p_d(k-1) = p_k$$

$$\mu_d(k-1) = \mu_k$$

Long Lines — Exponential Processing Time
Example



i	Parameters			
	r_i	p_i	μ_i	N_i
1	.05	.03	.5	8
2	.06	.04	—	8
3	.05	.03	.5	

- Exponential processing time line — 3 machines
- Upper bound determined by smallest ρ_i.
- Simulation satisfies upper bound; decomposition does not. *Why?*

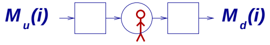
Long Lines — Continuous Material

Conceptually very similar to exponential processing time model. One difference:

- prob $(x_{i-1} = 0 \text{ and } x_i = N_i) = 0$ *exactly* .

Long Lines — Continuous Material Model
New approximation

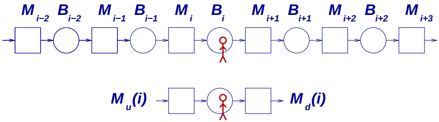
- *New approximation:* The observer sees both pseudo-machines operating at multiple rates, but the two-machine lines assume single rates.



If this were *really* a two-machine continuous material line,

- material would enter the buffer at rate $\mu_u(i)$ (if $M_u(i)$ is up and the buffer is not full) or $\mu_d(i)$ (if $M_u(i)$ and $M_d(i)$ are up and the buffer is full and $\mu_d(i) < \mu_u(i)$) or 0;
- material would exit the buffer at rate $\mu_d(i)$ (if $M_d(i)$ is up and the buffer is not empty) or $\mu_u(i)$ (if $M_u(i)$ and $M_d(i)$ are up and the buffer is empty and $\mu_u(i) < \mu_d(i)$) or 0;

Long Lines — Continuous Material
New approximation



Assume that ... $< \mu_{i-2} < \mu_{i-1} < \mu_i < \mu_{i+1} <$ .... Assume all the machines are up and B_i is not full. Then the observer in B_i *actually* sees material entering B_i ...

- at rate μ_i if B_{i-1} is not empty;
- at rate μ_{i-1} if B_{i-2} is not empty and B_{i-1} is empty;
- at rate μ_{i-2} if B_{i-3} is not empty and B_{i-2} is empty and B_{i-1} is empty;
- etc.

Therefore, this approximation may break down if the μ_i are very different.

Long Lines — Continuous Material
Equations

We have the same $6(k-1)$ unknowns, so we need $6(k-1)$ equations. They are, as before,

- *Interruption of flow* ,
- *Resumption of flow,*
- *Conservation of flow,*
- *Flow rate/idle time,*
- *Boundary conditions.*

They are the same as in the exponential processing time case except for the Interruption of Flow equations.

Long Lines — Continuous Material

Interruption of Flow

Considerable manipulation leads to

$$p_u(i) \quad = \quad p_i \left(1 + \frac{\mathbf{p}_{i-1}(0,1,1)\mu_u(i)}{P(i) - \mathbf{p_i}(N_i,1,1)\mu_d(i)} \left(\frac{\mu_u(i-1)}{\mu_i} - 1\right)\right) + \left(\frac{\mathbf{p}_{i-1}(0,0,1)\mu_u(i)}{P(i) - \mathbf{p_i}(N_i,1,1)\mu_d(i)}\right) r_u(i-1), i = 2, \cdots, k-1$$

and, similarly,

$$p_d(i) \quad = \quad p_{i+1} \left(1 + \frac{\mathbf{p}_{i+1}(N_{i+1},1,1)\mu_d(i)}{P(i) - \mathbf{p_i}(0,1,1)\mu_u(i)} \left(\frac{\mu_d(i+1)}{\mu_{i+1}} - 1)\right)\right) + \left(\frac{\mathbf{p}_{i+1}(N_{i+1},1,0)\mu_d(i+1)}{P(i) - \mathbf{p_i}(0,1,1)\mu_u(i)}\right) r_d(i+1), i = 1, \cdots, k-2$$

To come

- Assembly/Disassembly Systems
- Buffer Optimization
- Effect of Buffers on Quality
- Loops
- Real-Time Control
- ????

