Analysis and Control of Large Flow/Storage Systems Using Decomposition

Stanley B. Gershwin

Duke University
Fuqua School of Business

November 11, 2005

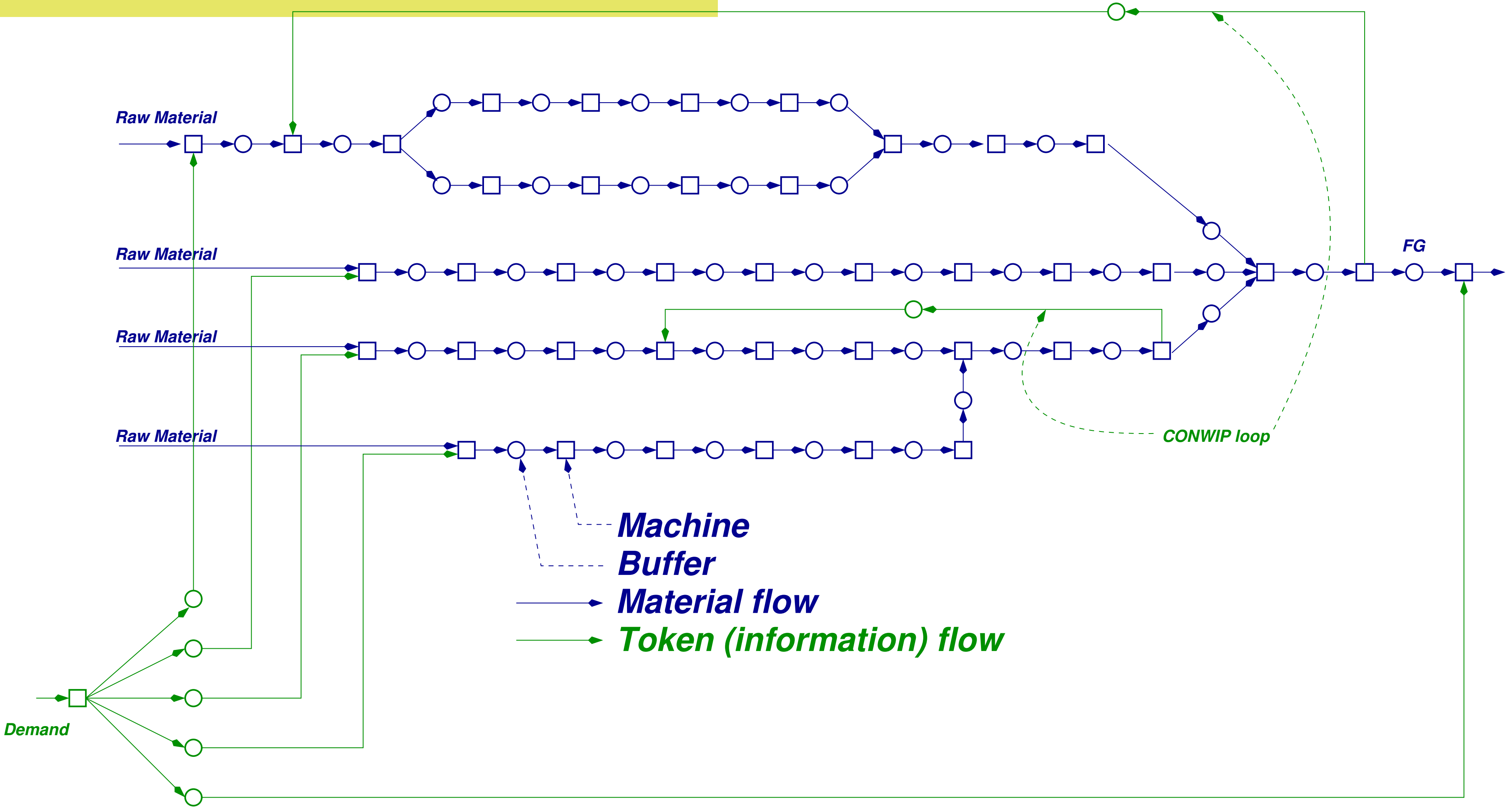
Introduction

- Survey of past and current research in the analysis of a class of large systems by decomposition.

- Mainly focused on my own work.

- Many other people have contributed.

Description of systems



Description of systems

Characteristics

- Material or tokens travel from machine to buffer to machine.
- Machines may assemble or disassemble or both or neither.
- Machine behavior is stochastic:
  - ⋆ random operation times or machine failures.
- Buffers are finite.
- Material flow control is accomplished by token flow.

- Motivated by manufacturing systems.
- Finiteness of buffers especially important. *("Lean Manufacturing.")*
  - ⋆ *Infinite buffers are* <u>not</u> *always a good approximation to large buffers.*
- Used by or for GM, HP, Peugeot, Johnson & Johnson, Hitachi, and a small Athens frozen pizza company.
  - ⋆ GM project was an INFORMS Edelman winner; HP and Peugeot were Edelman finalists; economic impact is *at least* hundreds of millions of dollars.

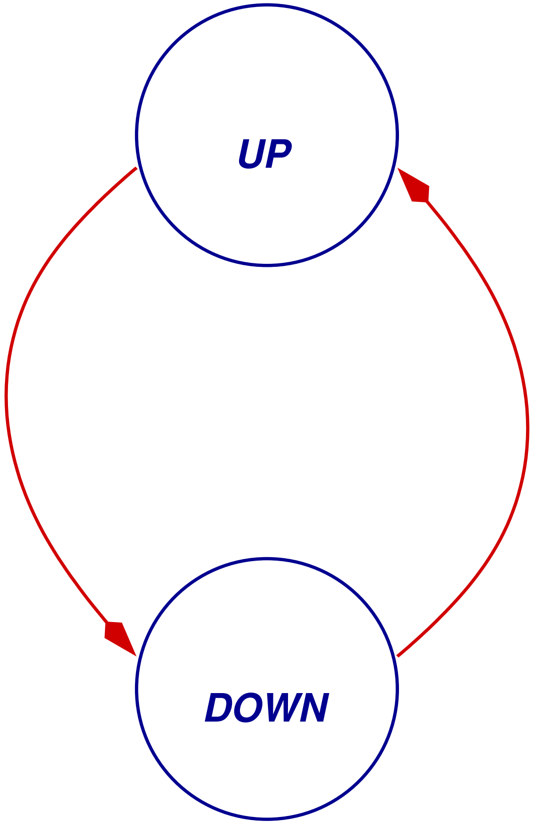
Difficulty

- Two-machine-one-buffer systems can be solved analytically.
- No known analytic solution for more than two machines.
- Size of the state space is given by

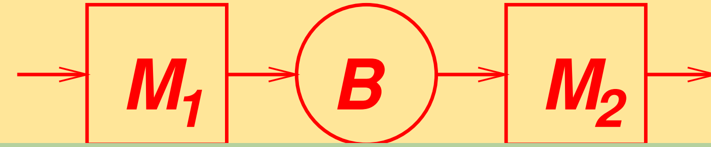
$$2^k \prod_{i=1}^{k} (N_i + 1)$$

assuming k two-state machines and $k - 1$ buffers of size $N_i, i = 1, ..., k - 1$.

- Machines have two states: *up* and *down*.
- Cases:
  - ⋆ Discrete time, discrete material.
  - ⋆ Continuous time, discrete material.
  - ⋆ Continuous time, continuous material.



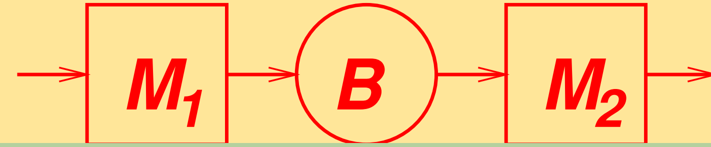
Description of systems



Discrete time, discrete material

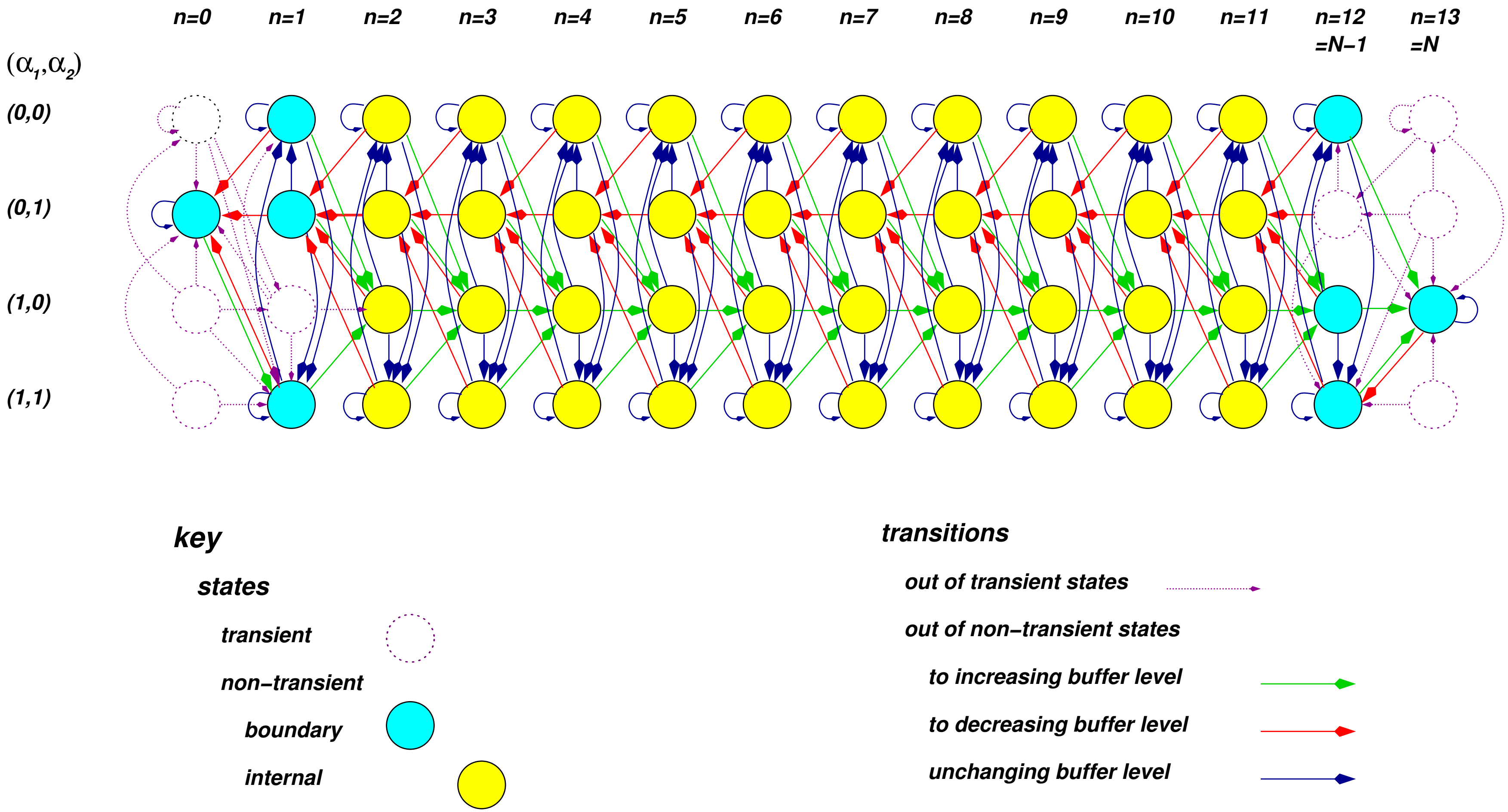
- Operation time = 1.
- Two parameters per machine:
  - ⋆ p_i = probability of failure during an operation time when machine is up;
  - ⋆ r_i = probability of repair during an operation time when machine is down.
- One parameter for the buffer: N, the maximum number of parts it can hold.
- We construct a Markov chain, obtain its steady-state probability distribution, and evaluate performance measures (production rate, probability of starvation and blockage, average inventory).

Description of systems

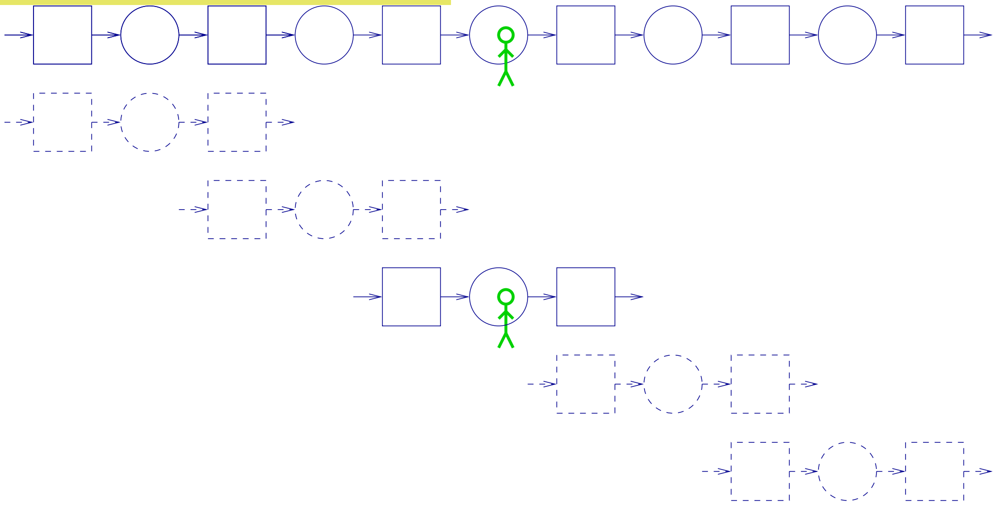


Discrete time, discrete material

State Transition Graph for Deterministic Processing Time, Two−Machine Line



Decomposition



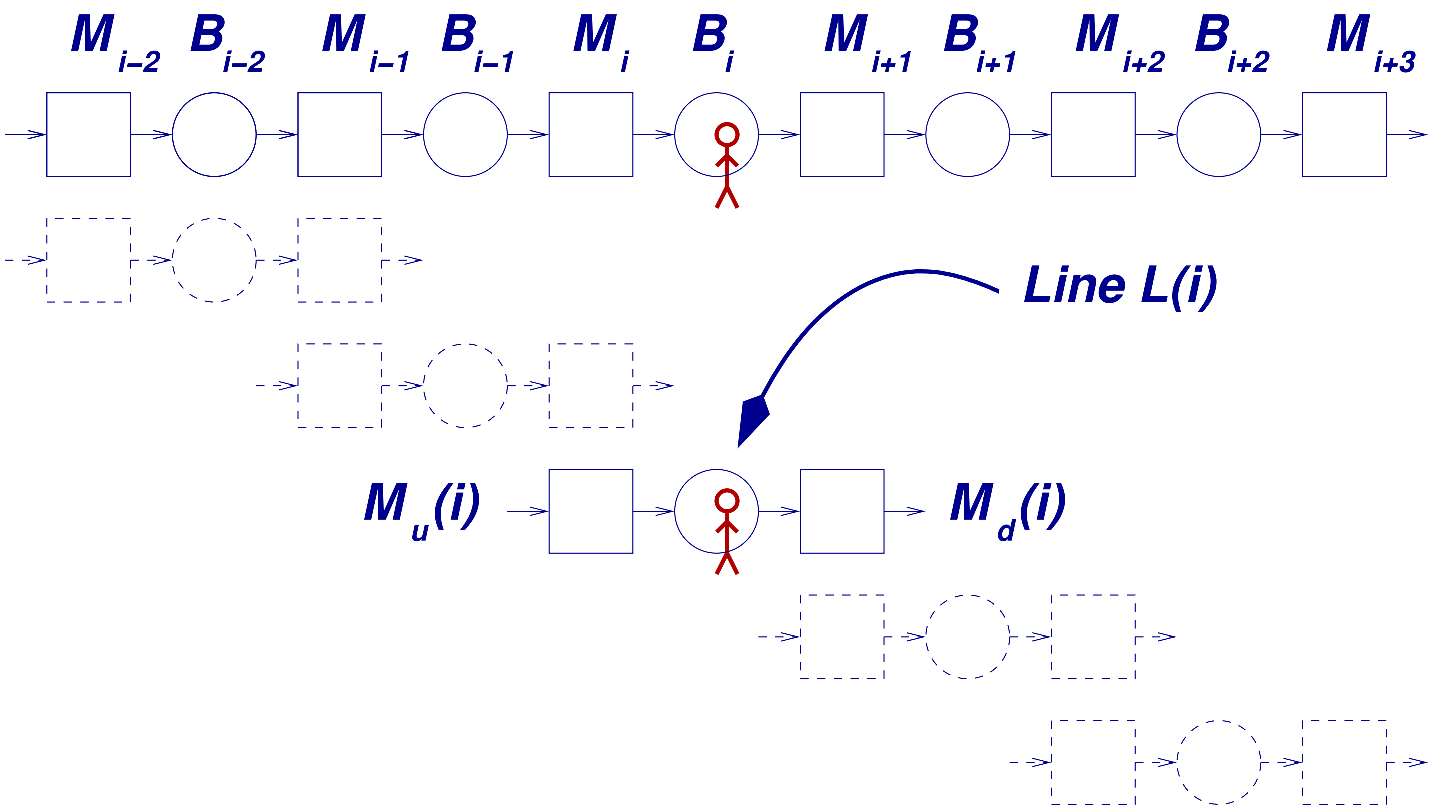
- Decomposition breaks up systems and then reunites them.
- Conceptually: put an observer in a buffer, and tell him that he is in the buffer of a two-machine line.
- Question: *What would the observer see, and how can he be convinced he is in a two-machine line? Construct the two-machine line. Construct all the two-machine lines.*

Decomposition

- 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



Decomposition

There are $4(k-1)$ unknowns. Therefore, we need

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

Decomposition

Equations

- *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 $M_d(k-1)$.

- All the quantities in 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.

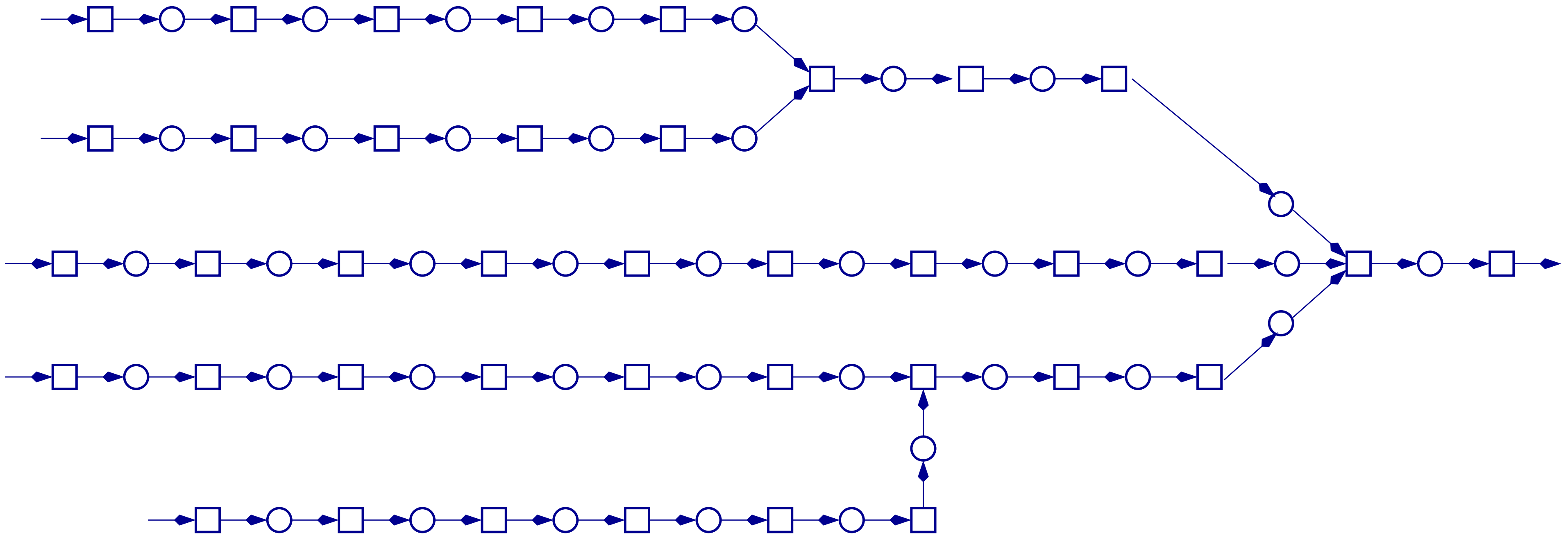
DDX algorithm : due to Dallery, David, and Xie (1988).

1. Guess the downstream parameters of $L(1)$ ($r_d(1), p_d(1)$). Set $i = 2$.
2. Use the 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 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.

Extensions

- Extended to the continuous-time models (discrete and continuous material).
- *Optimization:* evaluation embedded in gradient search.
  - ⋆ *Primal* Minimize buffer space subject to production rate constraint.
  - ⋆ *Dual* Maximize production rate subject to buffer space constraint.
- Acyclic A/D (tree-structured) systems
  - ⋆ Straightforward extension of line equations and algorithm

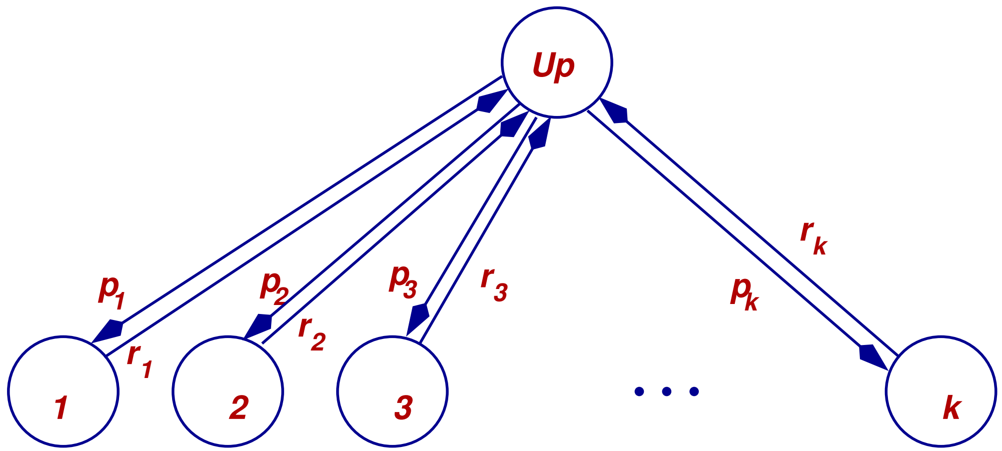
Extensions



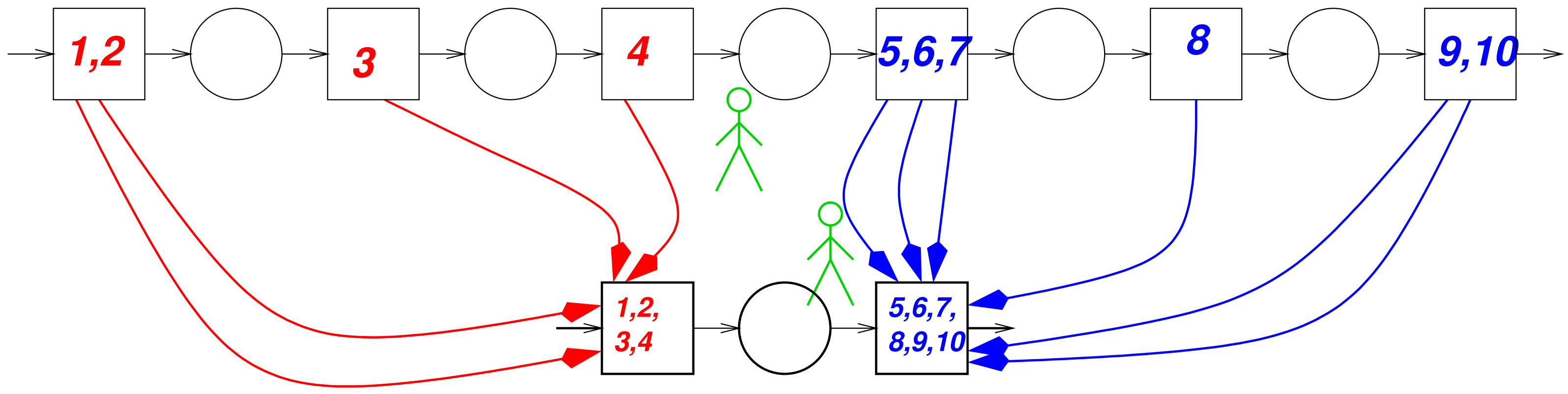
Extensions

Multiple-failure-mode models

- Machines can fail in modes with different MTTF and MTTR.
- Used in decomposition (Tolio et al, 1998).
- Improves accuracy of lines, but more useful for loops.

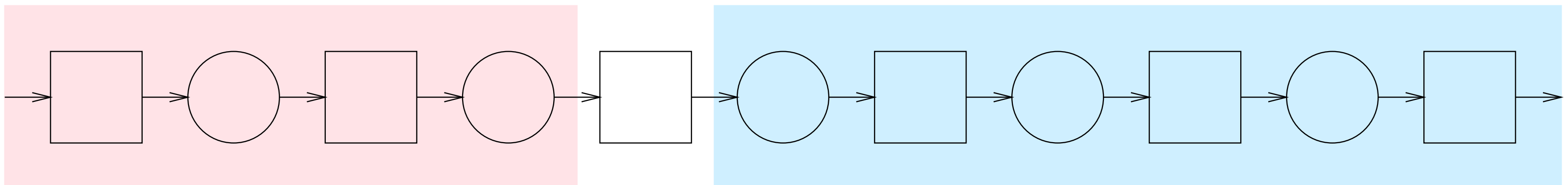


Extensions



- For each failure mode downstream of a given buffer, there is a corresponding mode in the downstream machine of its two-machine line. Similarly for upstream modes.
- The downstream failure modes appear to the observer after propagation through *blockage* .
- The upstream failure modes appear to the observer after propagation through *starvation* .

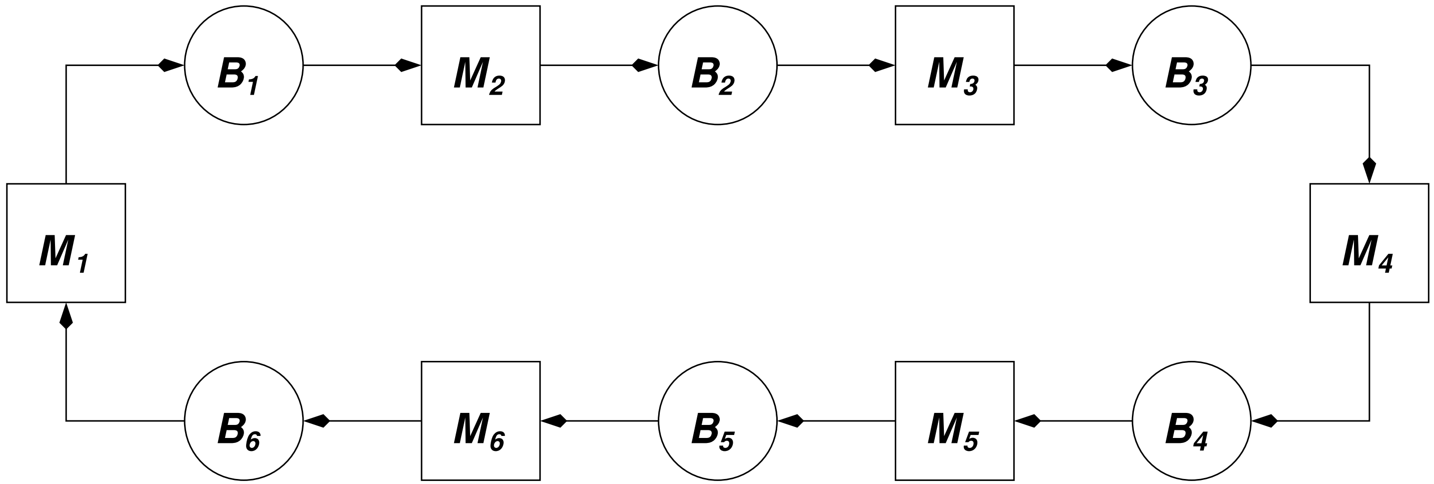
Extensions



- The *range of blocking of a machine* is the set of all machines that could block it if they stayed down long enough.
  - ⋆ The range of blocking of a machine in a line is the entire downstream part of the line.
- The *range of starvation of a machine* is the set of all machines that could starve it if they stayed down long enough.
  - ⋆ The range of starvation of a machine in a line is the entire upstream part of the line.

Extensions

Single-loop systems



- Finite buffers $(0 \leq n_i(t) \leq N_i)$.
- Closed loop – fixed population $(\sum_i n_i(t) = N)$.
- Motivation:
  - ⋆ Limited pallets/fixtures.
  - ⋆ CONWIP (or hybrid).

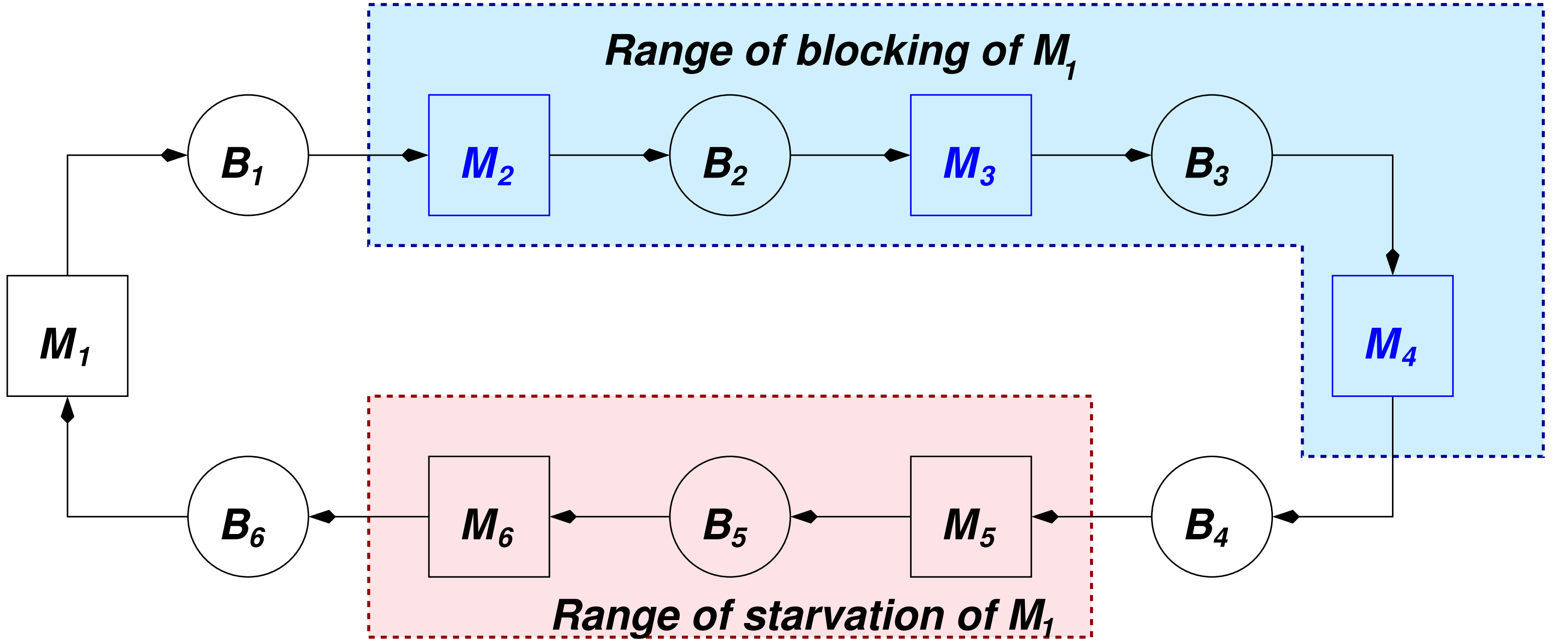
Extensions

- Frein, Commault, Dallery (1996):
  - ⋆ Treat the loop as a line in which the first machine and the last are the same.
  - ⋆ In the resulting decomposition, one equation is missing.
  - ⋆ The missing equation is replaced by *the expectation of* the population constraint $(\sum_i \bar{n}_i(t) = N)$.
  - ⋆ Accuracy good for large systems, not so good for small systems.
  - ⋆ Accuracy good for intermediate-size populations; not so good for very small or very large populations.

Extensions

Single-loop systems

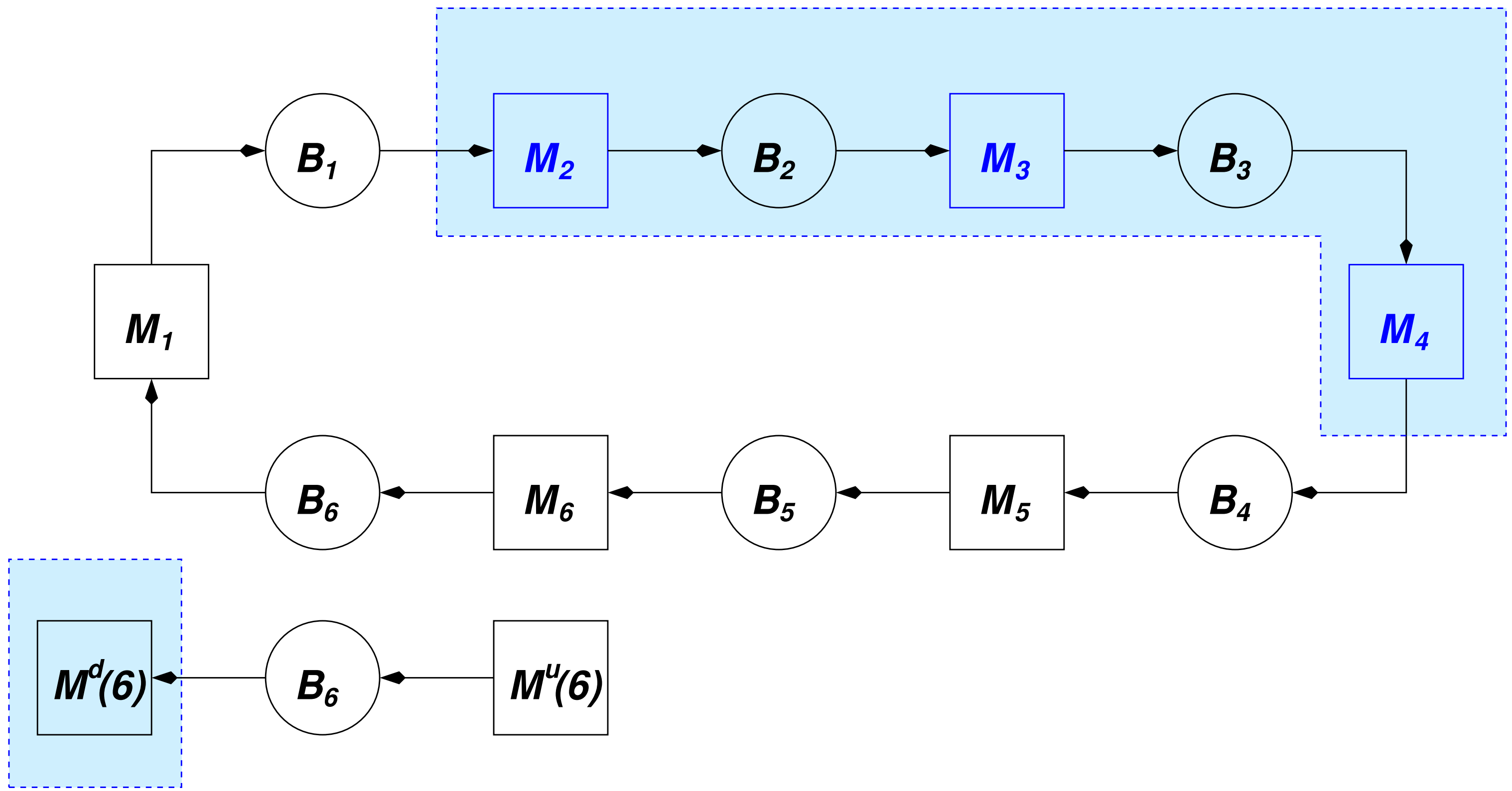
Multiple-failure-mode method



Extensions

Single-loop systems

Multiple-failure-mode method

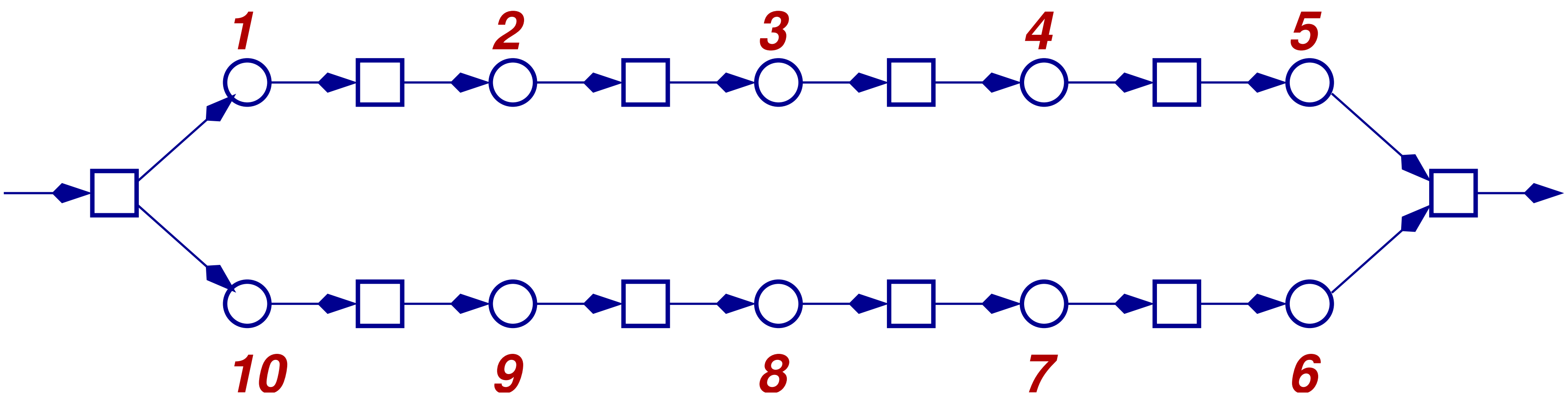


- *Analysis method:* Use the multi-failure-mode decomposition, but adjust the ranges of blocking and starvation accordingly.

Extensions

Single-loop systems

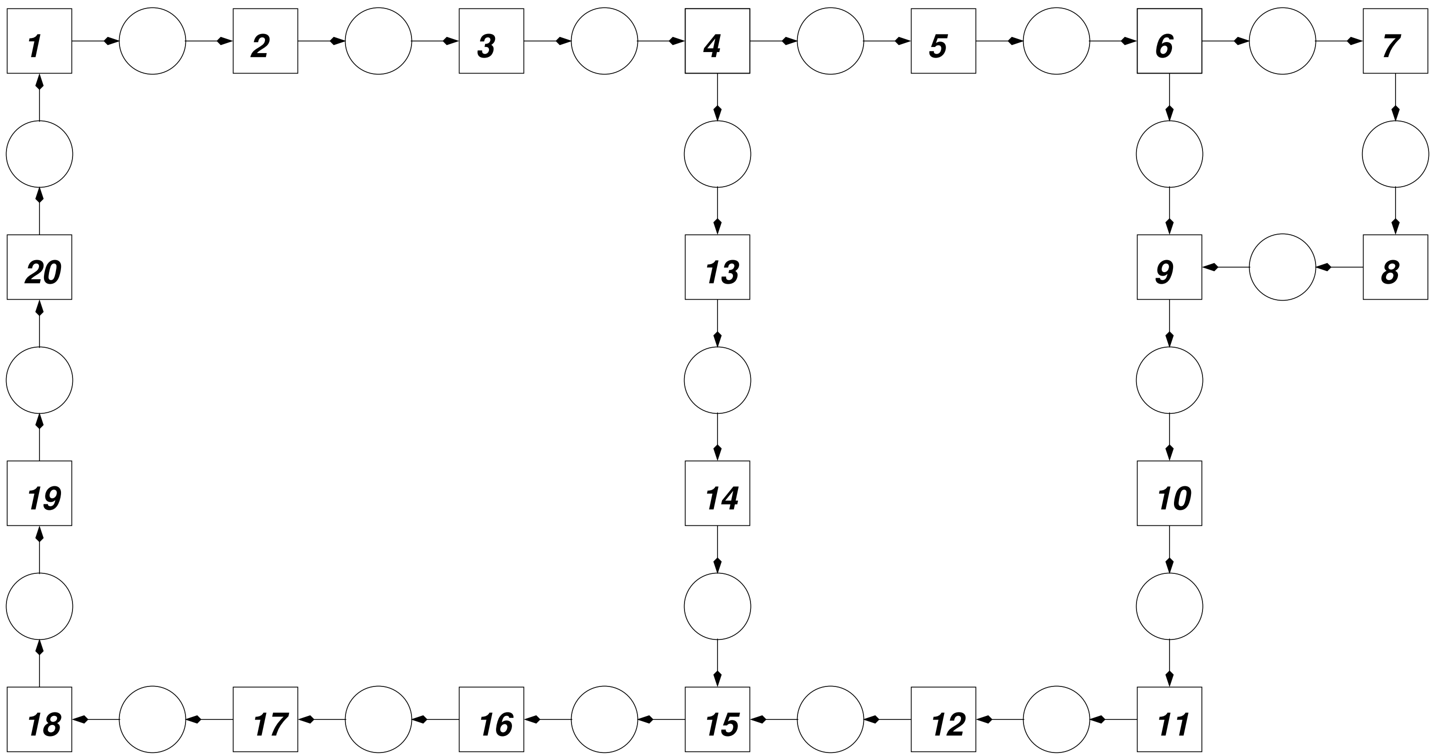
Disassembly–Assembly



Invariant:

$$n_1 + n_2 + n_3 + n_4 + n_5 - n_6 - n_7 - n_8 - n_9 - n_{10} = C$$

Extensions



- One invariant for each independent loop.
- This complicates the ranges of blocking and starvation.

Extensions

Multiple-loop systems

Decomposition and algorithm

- Levantesi PhD thesis, 2001
- Decomposition equations are identical to those of Tolio and Matta (1998).
- Algorithm:
  - ⋆ Phase 1: Determine ranges of blocking and starvation.
  - ⋆ Phase 2: Tolio and Matta DDX-type algorithm

Extensions

Multiple-loop systems

Range of Blocking

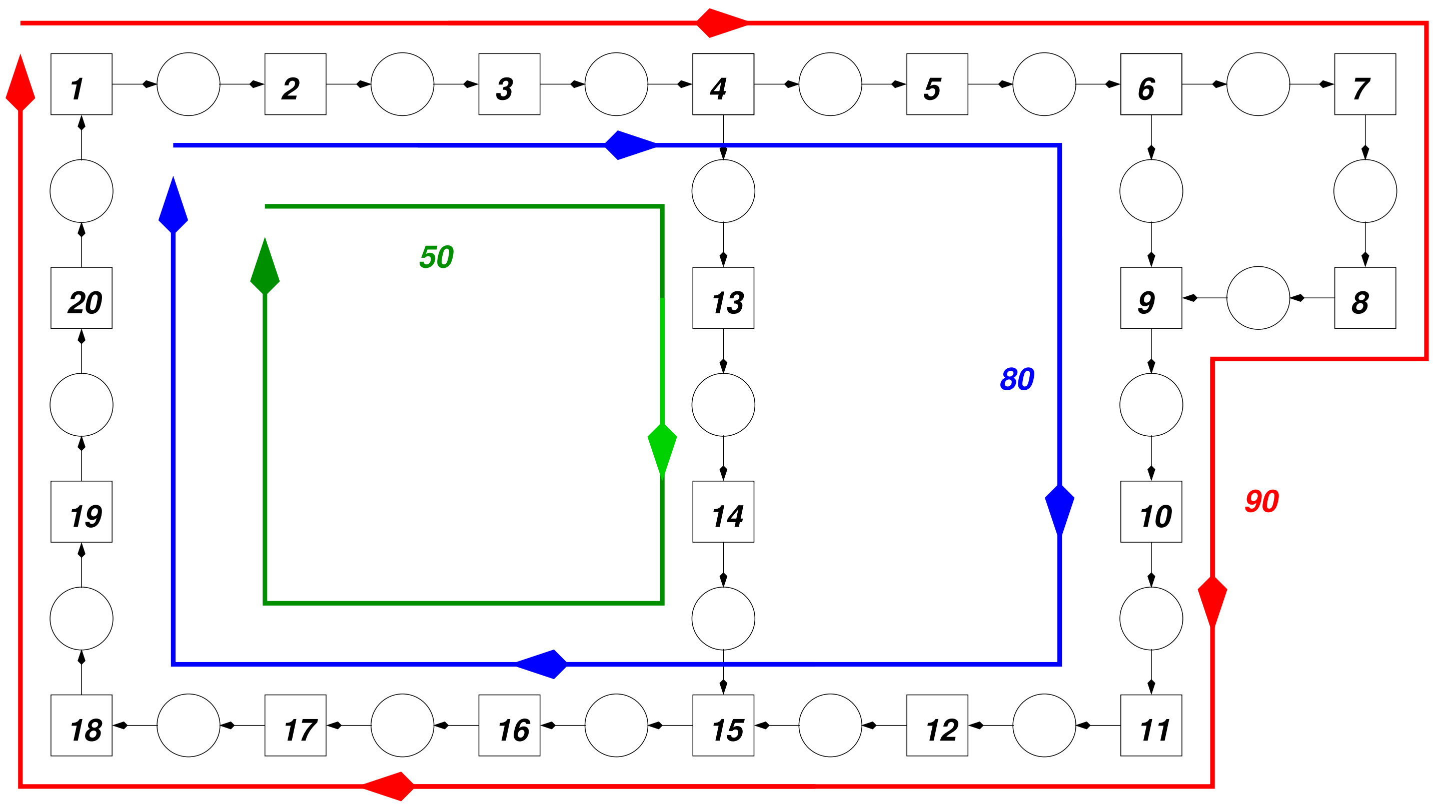
- To calculate the *range* of blocking for each machine M_i ...
  - ⋆ *(the set of machines M_j that could block M_i if M_j were down for a very long time)*
- we first determine the *domain* of blocking for each M_j
  - ⋆ *(the set of machines that M_j could block if it were down for a very long time)*
- and then transpose the table.

Similarly for the range of starvation.

Extensions

Multiple-loop systems

Example

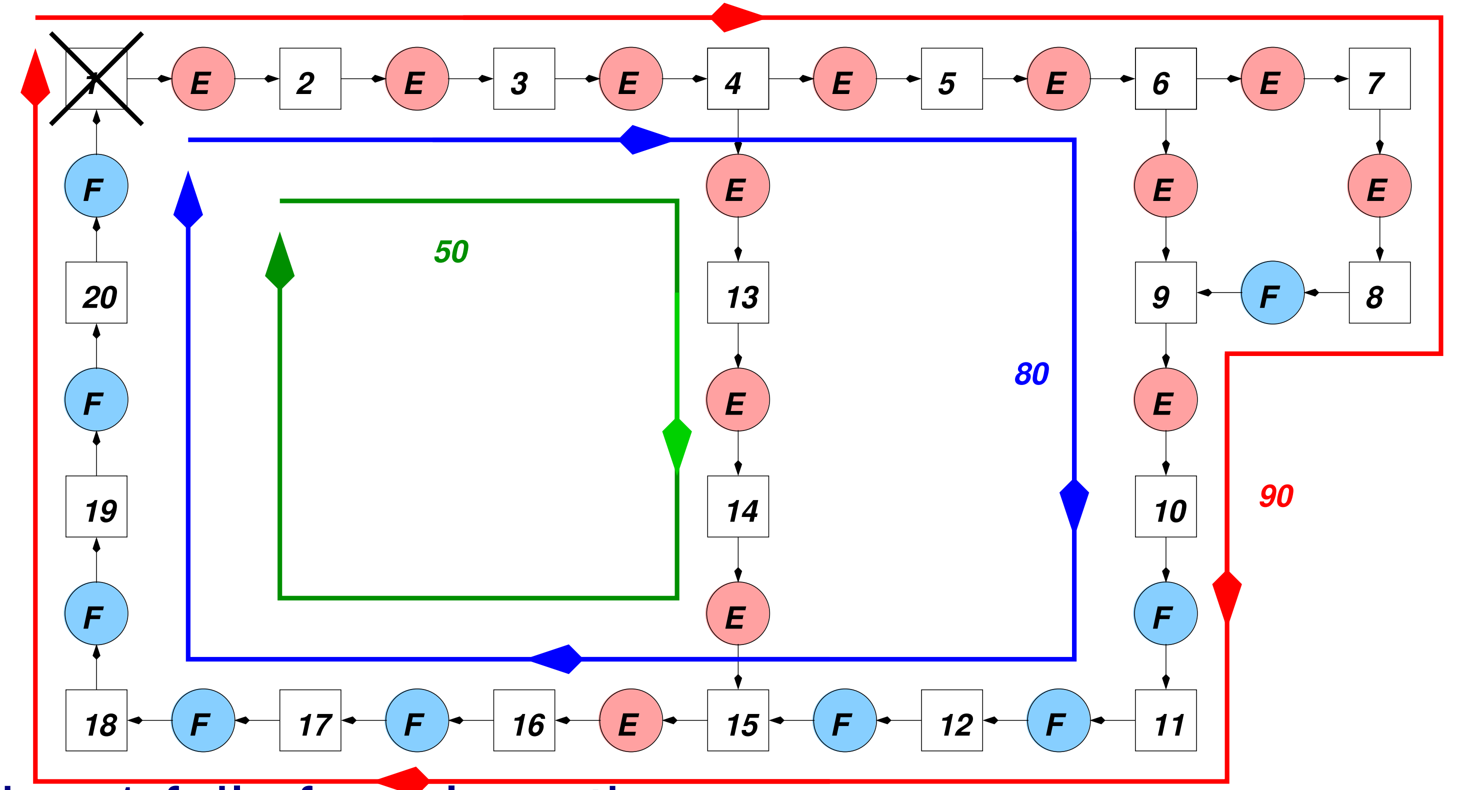


- All buffer sizes = 10.

Extensions

Multiple-loop systems

Example

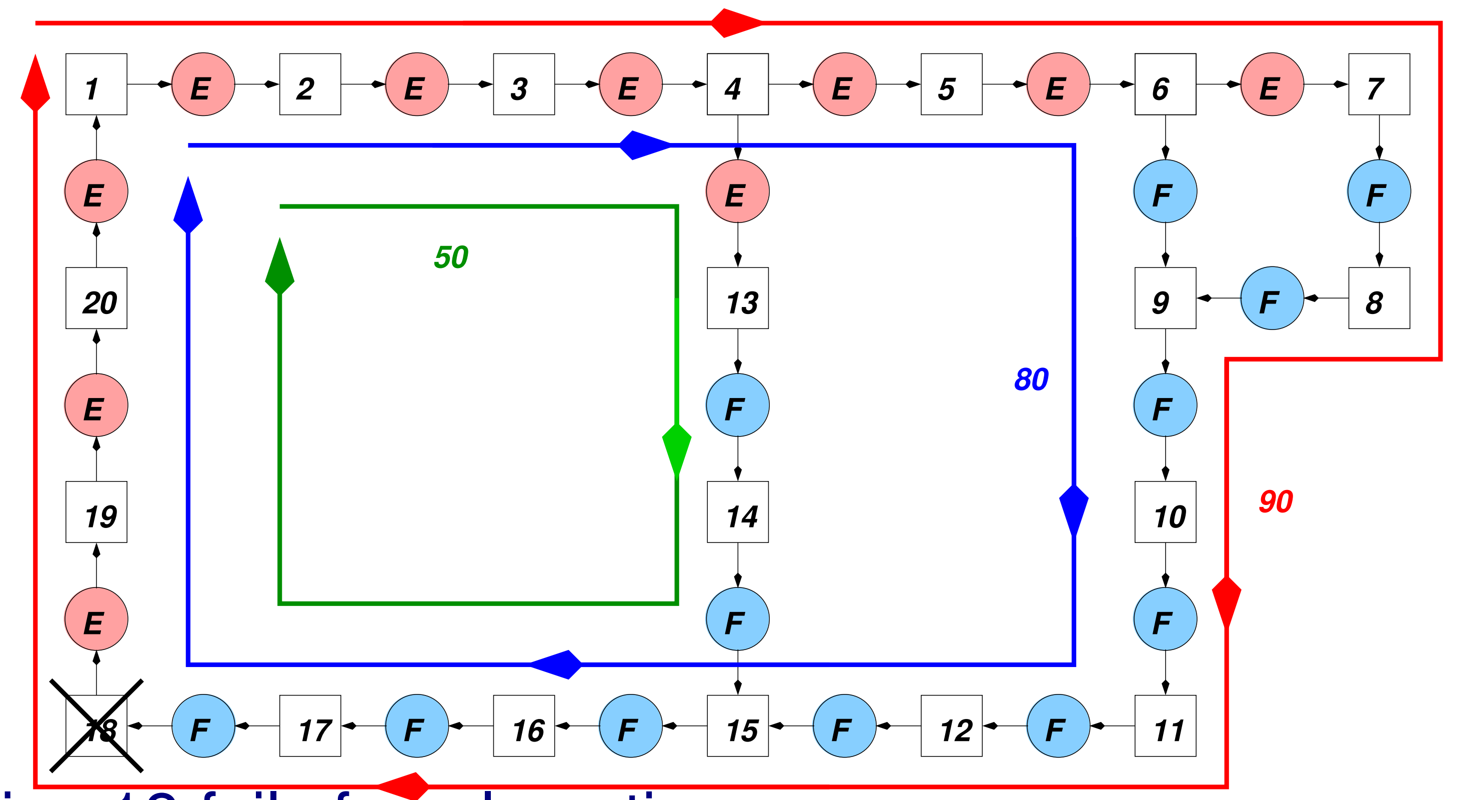


- Machine 1 fails for a long time.
- Buffers in the *domains* of blocking and starvation indicated.

Extensions

Multiple-loop systems

Example

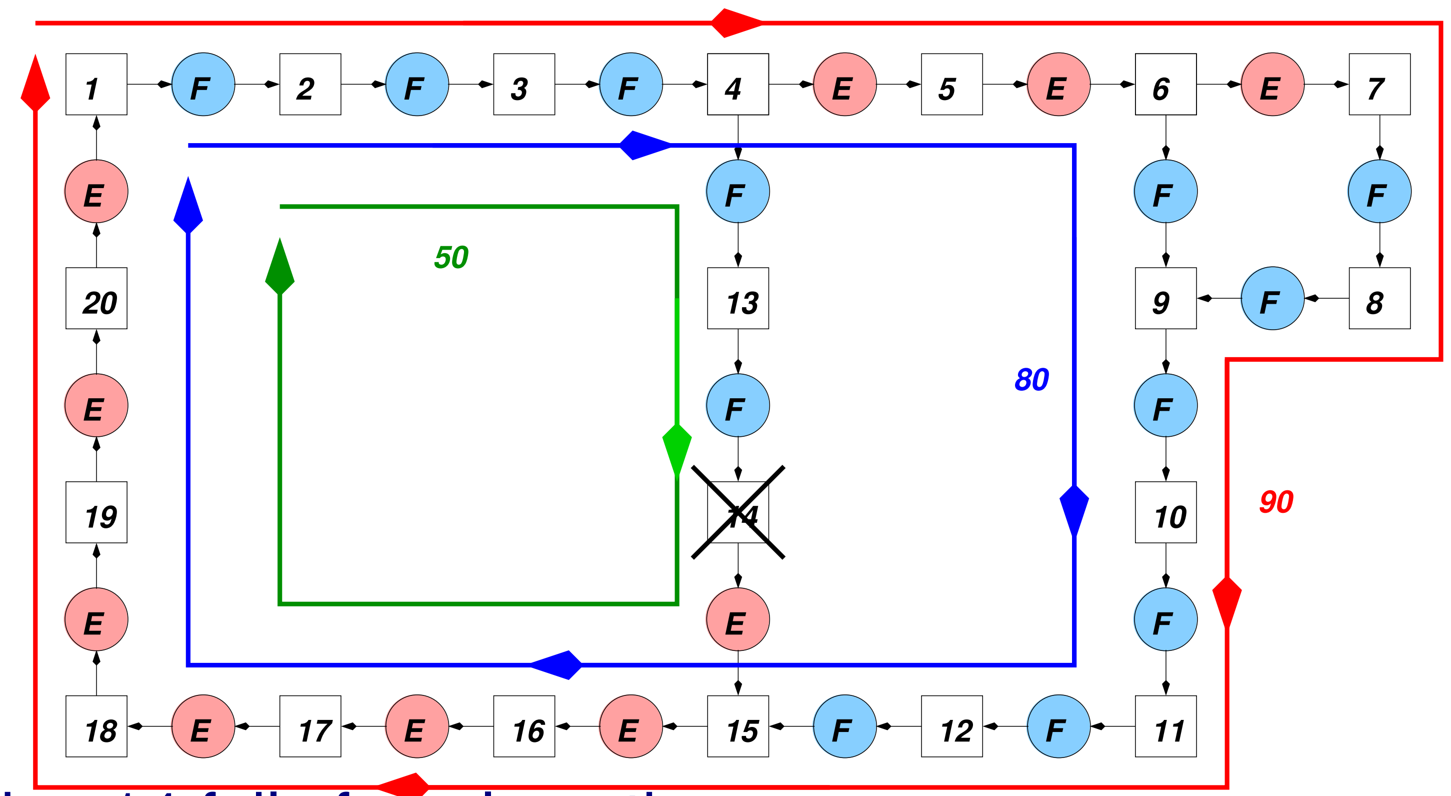


- Machine 18 fails for a long time.
- Buffers in the *domains* of blocking and starvation indicated.

Extensions

Multiple-loop systems

Example



- Machine 14 fails for a long time.
- Buffers in the *domains* of blocking and starvation indicated.

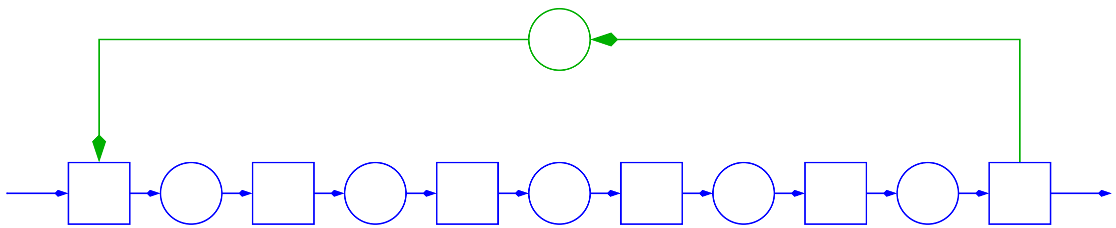
Extensions

- Ranges of blocking and starvation are more complex — they need not be contiguous.
- James Zhang's PhD thesis — in process — describes an efficient way of determining the ranges of blocking and starvation.

Extensions

Control with tokens

CONWIP, kanban, and hybrid

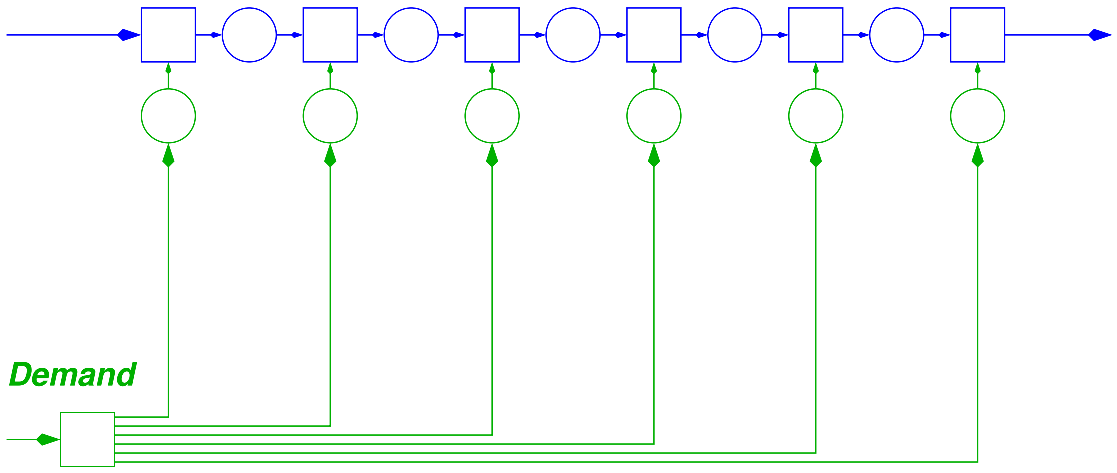


- *CONWIP:* finite population, infinite buffers
- *kanban:* infinite population, finite buffers
- *hybrid:* finite population, finite buffers

Extensions

Control with tokens

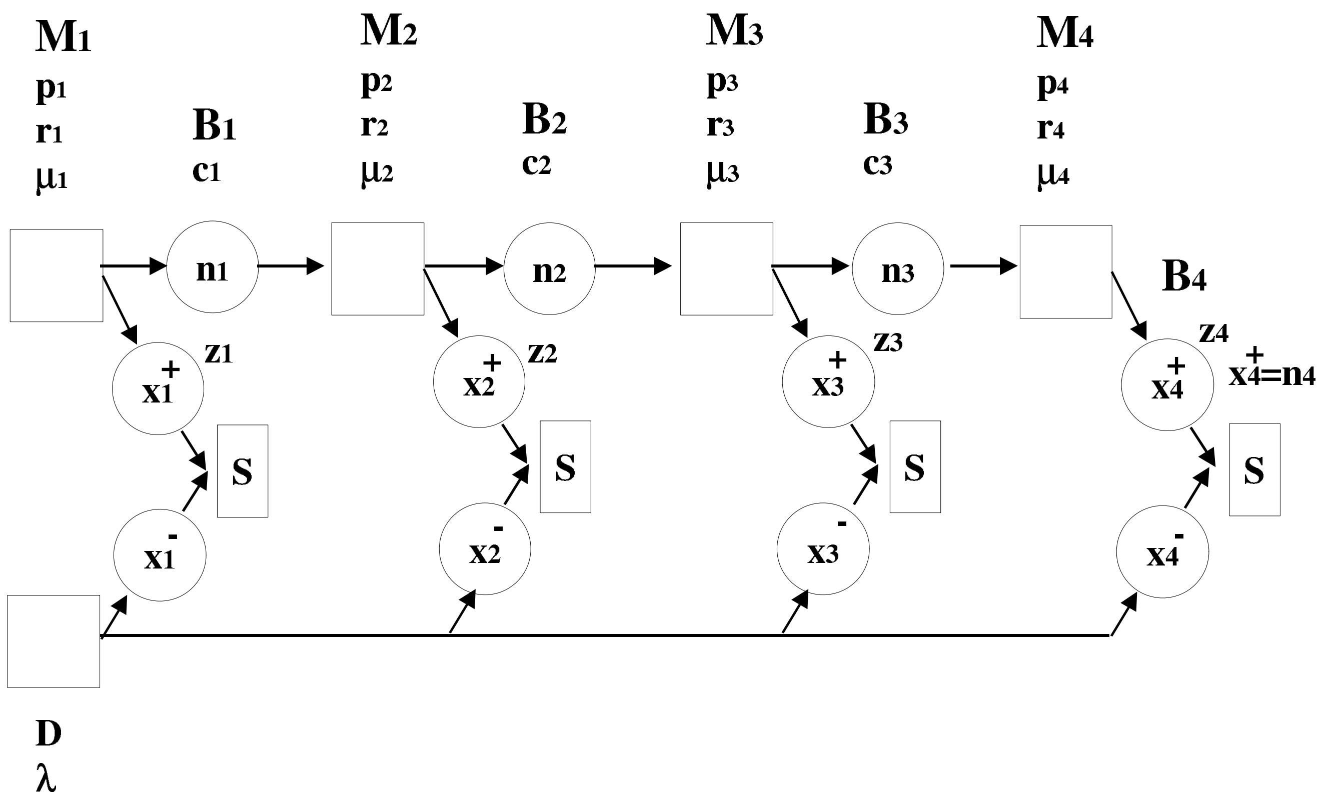
Basestock



Extensions

Control with tokens

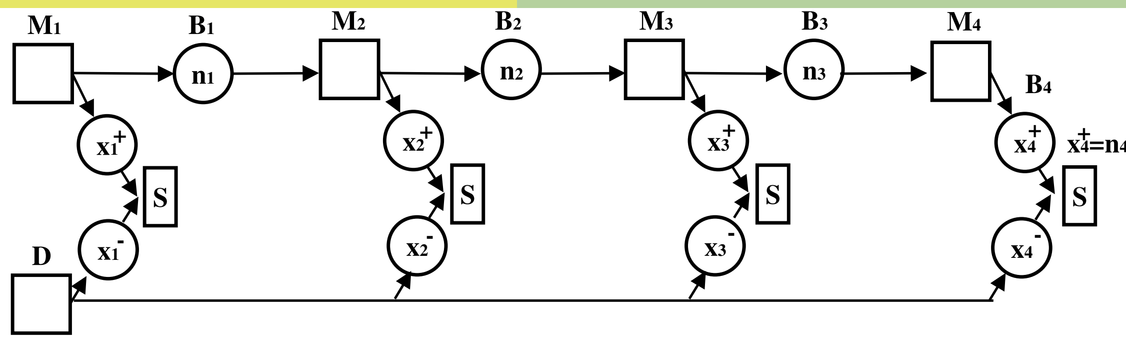
Control Point Policy

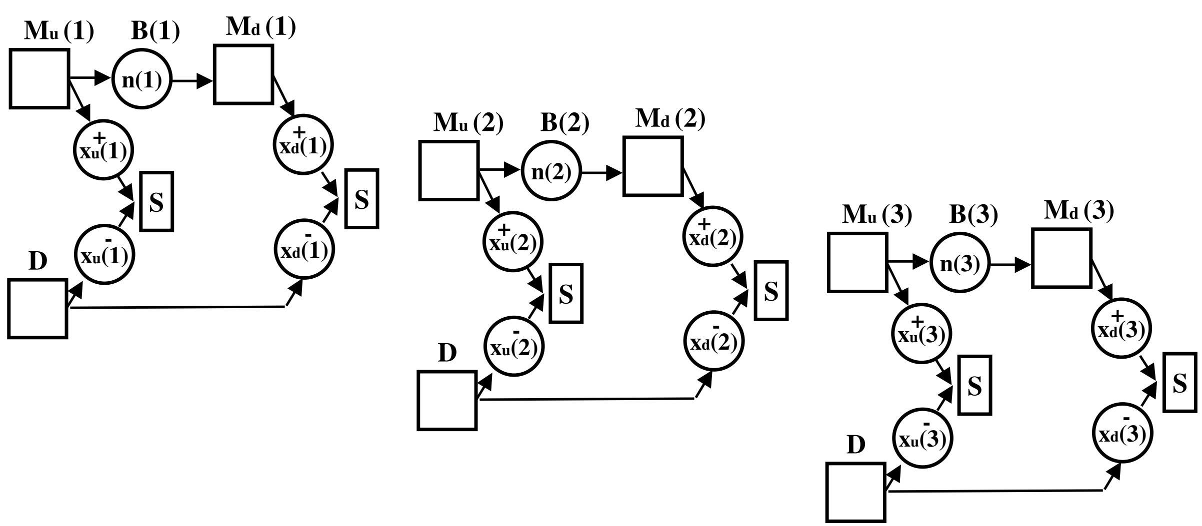


Extensions

Control with tokens

Alternate decomposition approach



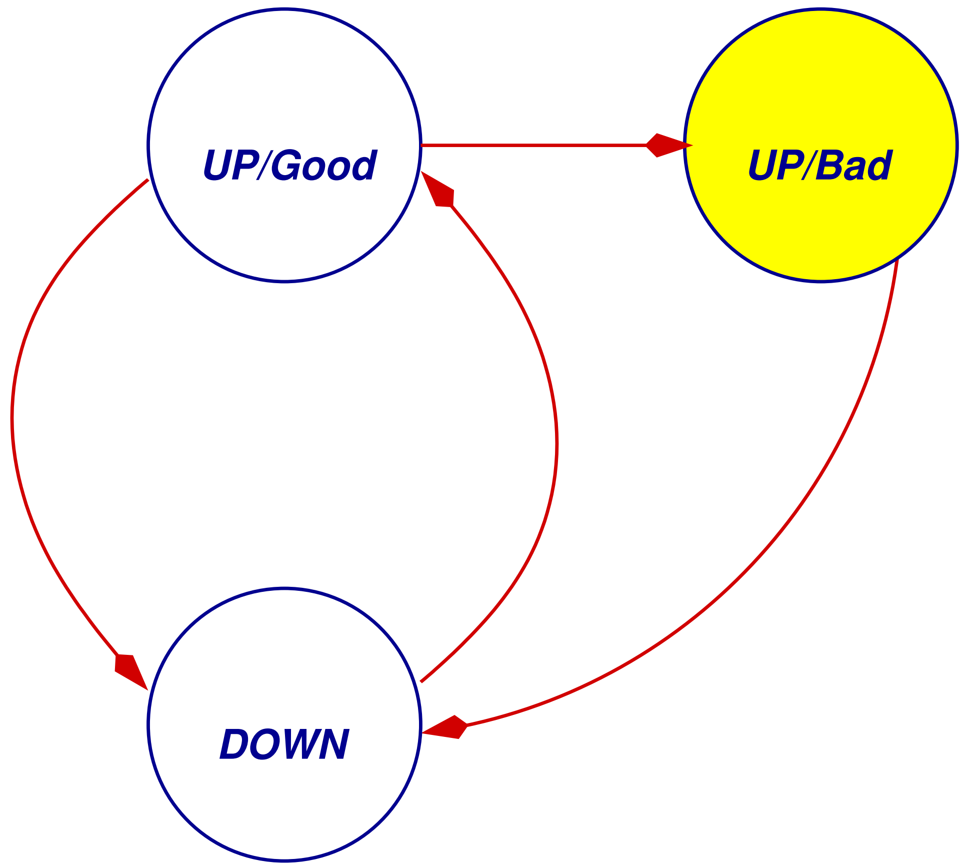


Vericourt and Gershwin (2004)

Extensions

Quality/Quantity Modeling

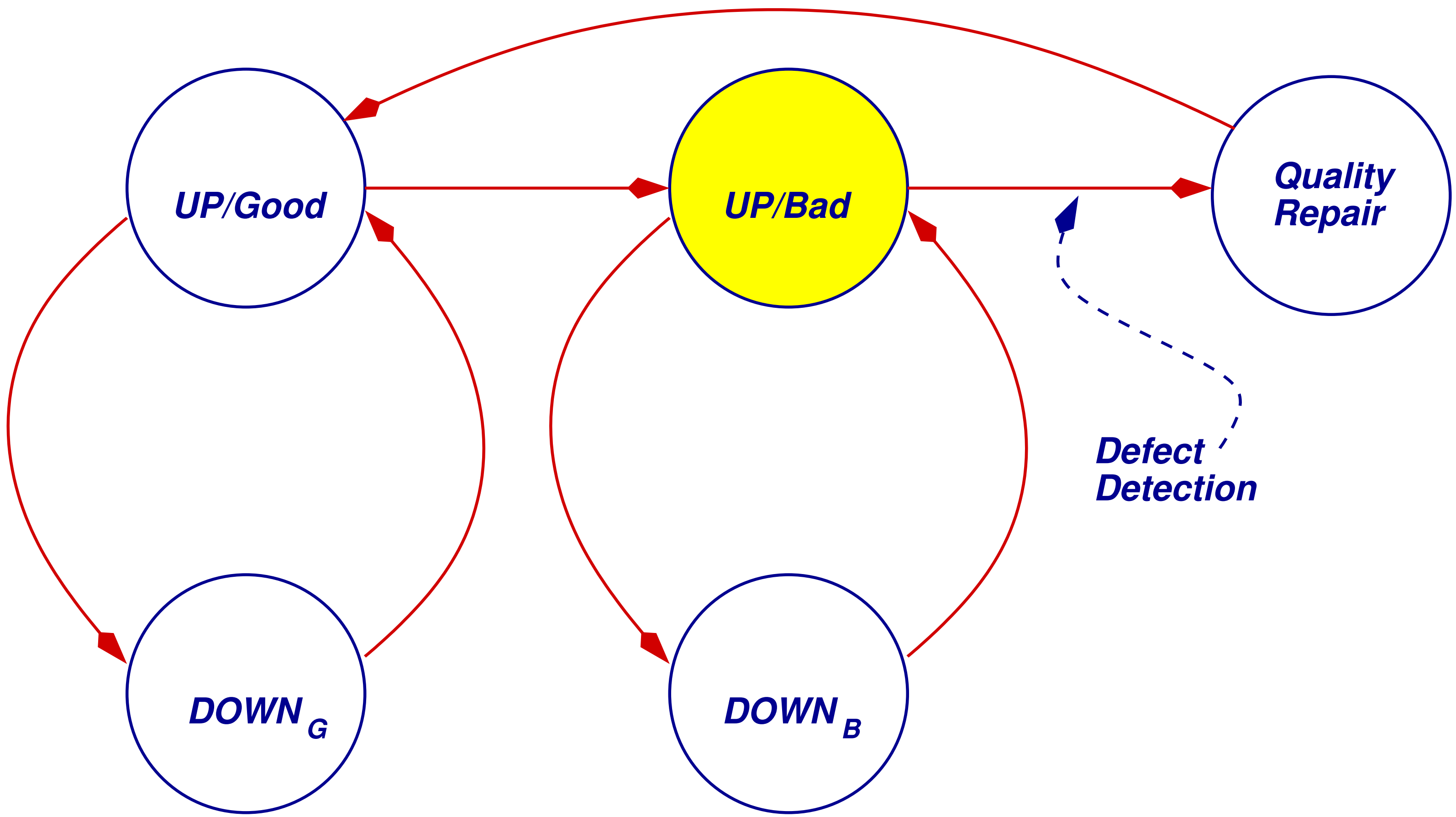
Simple machine model



Extensions

Quality/Quantity Modeling

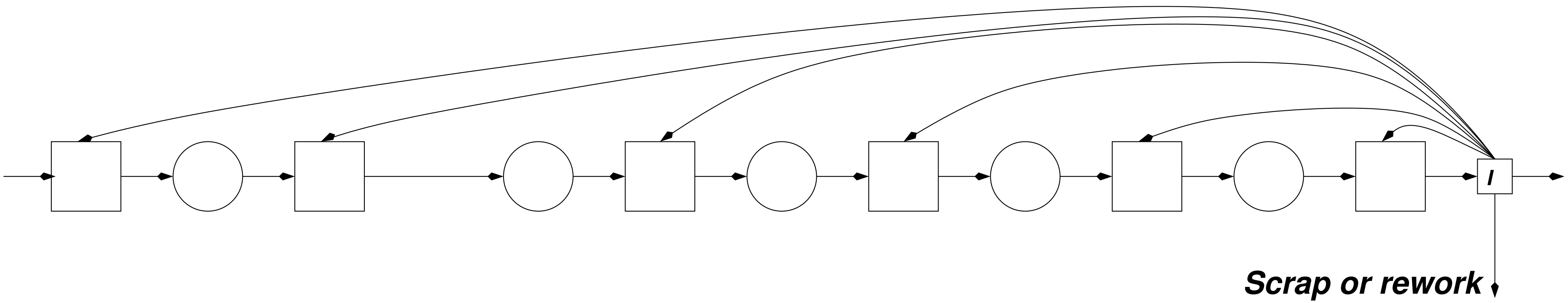
Slightly more realistic model



Extensions

Quality/Quantity Modeling

Inspection in long lines



- The transition from UP/BAD to QUALITY/REPAIR is signaled from a downstream inspection.
- The detection of the failure can only occur when the first bad part reaches the inspection station.
- Thus the production rate of *good* parts depends on how much inventory there is between the machine and the inspection.

Extensions

Inspection in long lines

To analyze this system by decomposition, we must

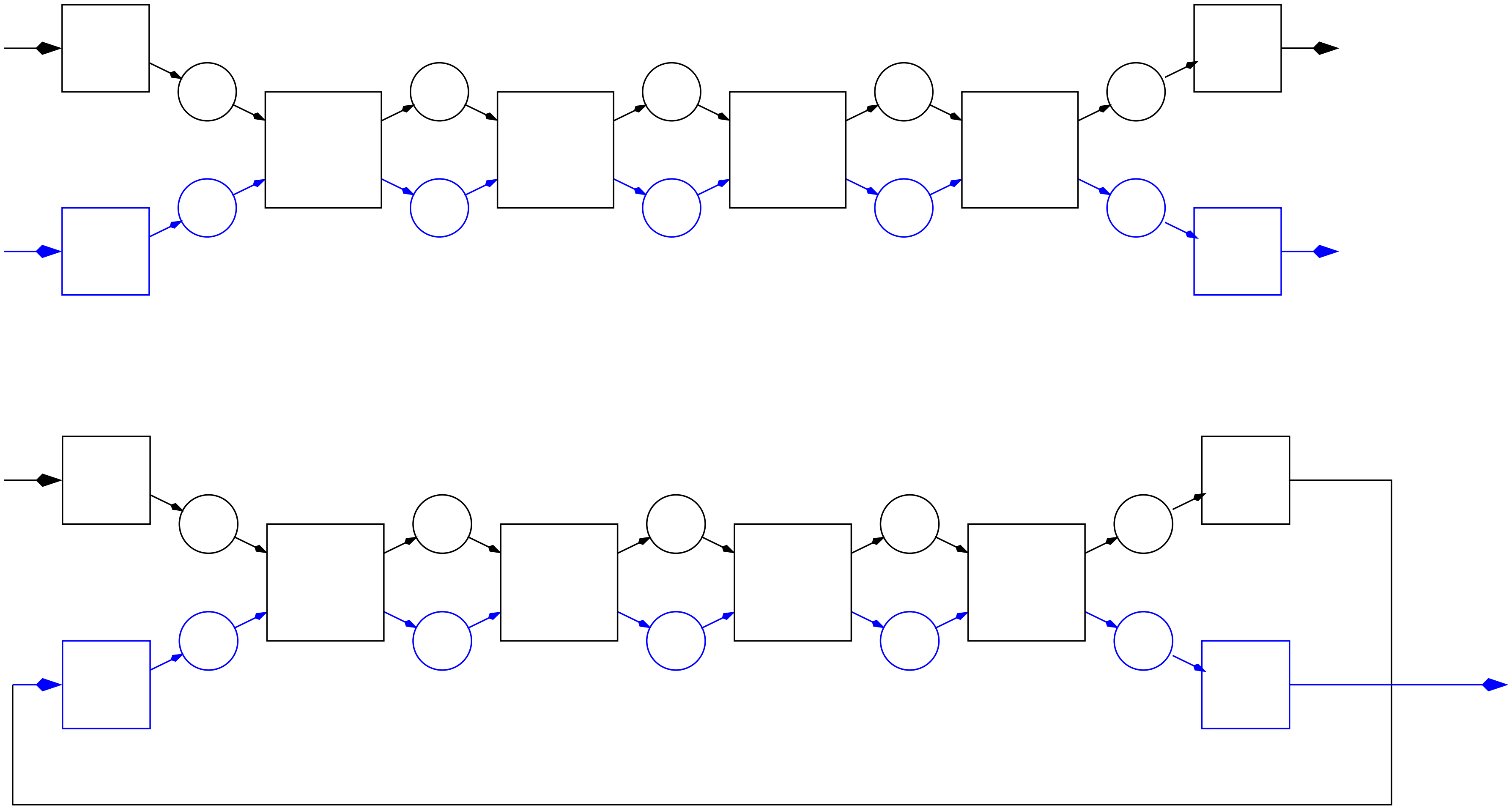
- analyze two-machine lines with multiple up- and down-states, and
- relate the transition rate from UP/BAD to QUALITY/REPAIR to the amount of inventory between the operation and the inspection.

Jongyoon Kim PhD (2004)

Extensions

Multiple part types and reentrant flow

Current work



Future Research

- Bounds on errors in decomposition.
- Improvement of errors in average inventory.
- Estimate of lead time.
- Optimization of token routing.
- Many topics in quality/quantity.
- ...

Closing Comment

Decomposition is what we do — formally or informally — whenever we deal with a complex system.

- We divide the world into two parts;
- we model the inside part in detail;
- and we simplify the outside world by summarizing it in the boundary.