Flow Control

☞ *and* ☜

Multiple-Loop, Finite Buffer Net

Analysis

Stanley B. Gershwin

Massachusetts Institute of Technology

May 2, 2002

Ecole Polytechnique Montre

Outline

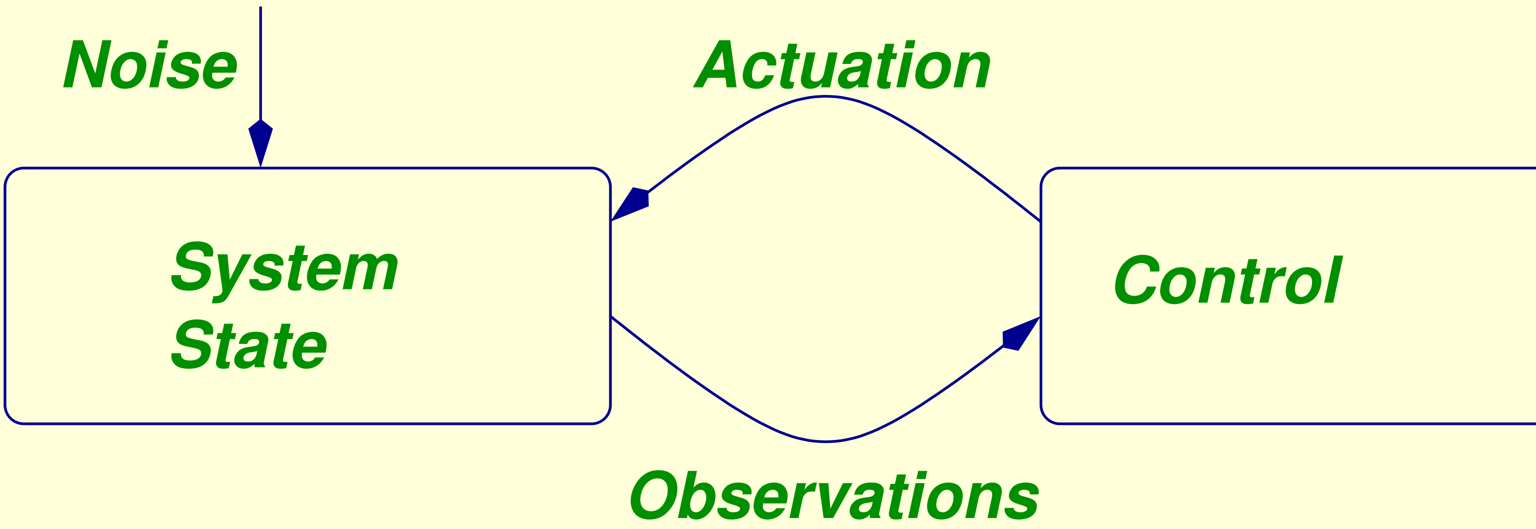
- Introduction
- Scheduling
  - ⋆ Triality
  - ⋆ Control point policy and other policies
- Decomposition
  - ⋆ Old decomposition
  - ⋆ Tolio decomposition
  - ⋆ Single loop systems
  - ⋆ Multiple loop systems
- More general policies
- Conclusions and future research

Introduction

- The goal is to develop good real-time sche policies for factories from first principles *a* develop analytical techniques for evaluatin
- Earlier scheduling work: solutions for single-machine, single part-type systems.
- Earlier evaluation work: tandem systems a tree-structured assembly/disassembly sys buffer policy. *Do operations whenever a m operational and is not blocked or starved.*

Real-time scheduling

Control Paradig



This is the general paradigm for control theo
engineering.

Real-time scheduling

- Many policies and approaches.
- No simple statement telling which is better.
- Policies are not all well-defined in the literature or i
- My opinion:
  - ⋆ This is because policies are not *derived* from firs
  - ⋆ Instead, they are invented and then tested and c
- One goal of the research is to find first principles a to guide policy development and choice.

Real-time scheduling

Control Paradig

In a factory,

- *State:* distribution of inventory, repair/failu of machines, etc.
- *Control:* move a part to a machine and st operation; begin preventive maintenance,
- *Noise:* machine failures, change in demar

Scheduling systems or methods should ...

- deliver good factory performance.
- compute decisions quickly, in response to conditions.

Real-time scheduling

Requirements

Performance G

- To minimize inventory and backlog.
- To maximize the probability that customers satisfied (ie, the probability of on-time deliv
- To maximize predictability (ie, minimize pe variability).

Real-time scheduling

Triality

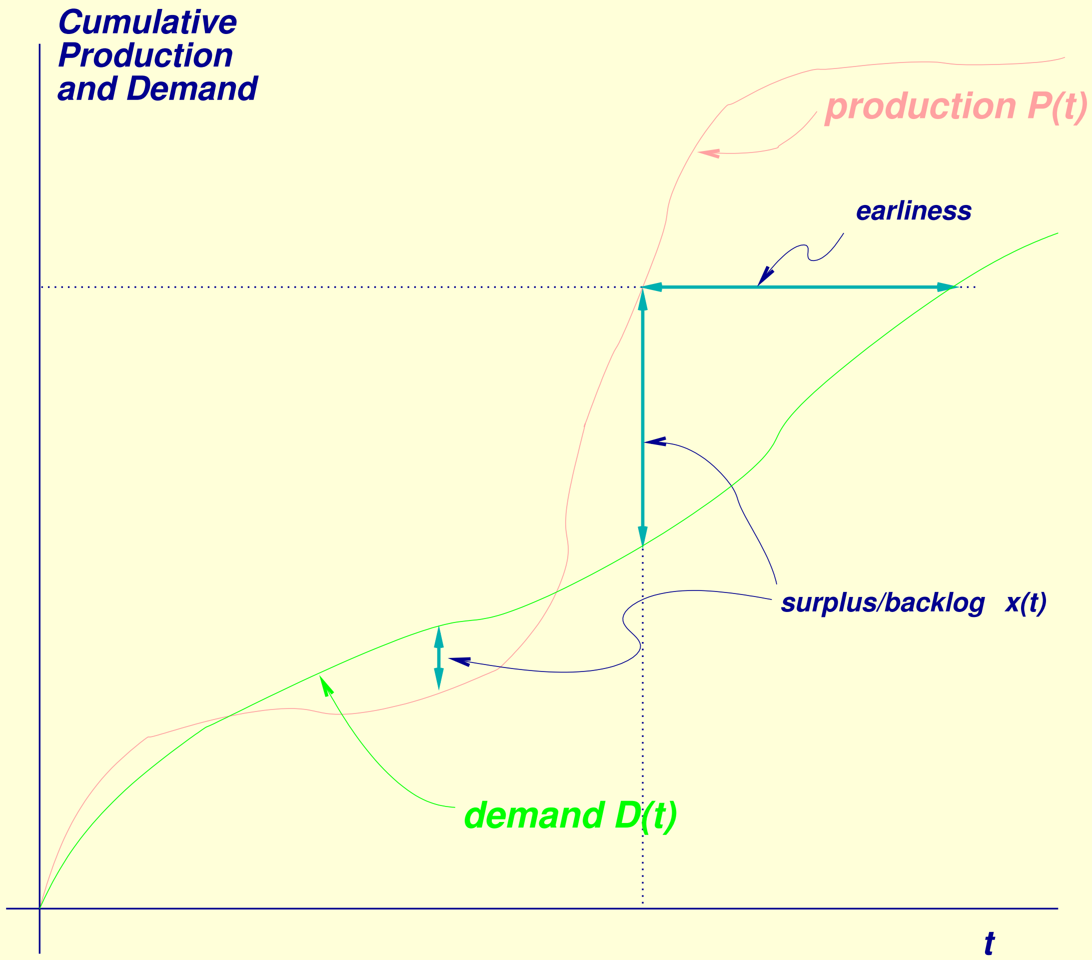
Three kinds of scheduling policies, which ar
sometimes exactly the same.

- *Surplus-based:* make decisions based on
  much production exceed demand.
- *Time-based:* make decisions based on ho
  late a product is.
- *Token-based:* make decisions based on p
  or absence of tokens.

Real-time scheduling

Objective of Sched

Surplus and t



- Objective is cumulative p close to cum demand.
- Surplus-bas look at vertic differences b graphs.
- Time-based at the horizo differences.

Real-time scheduling

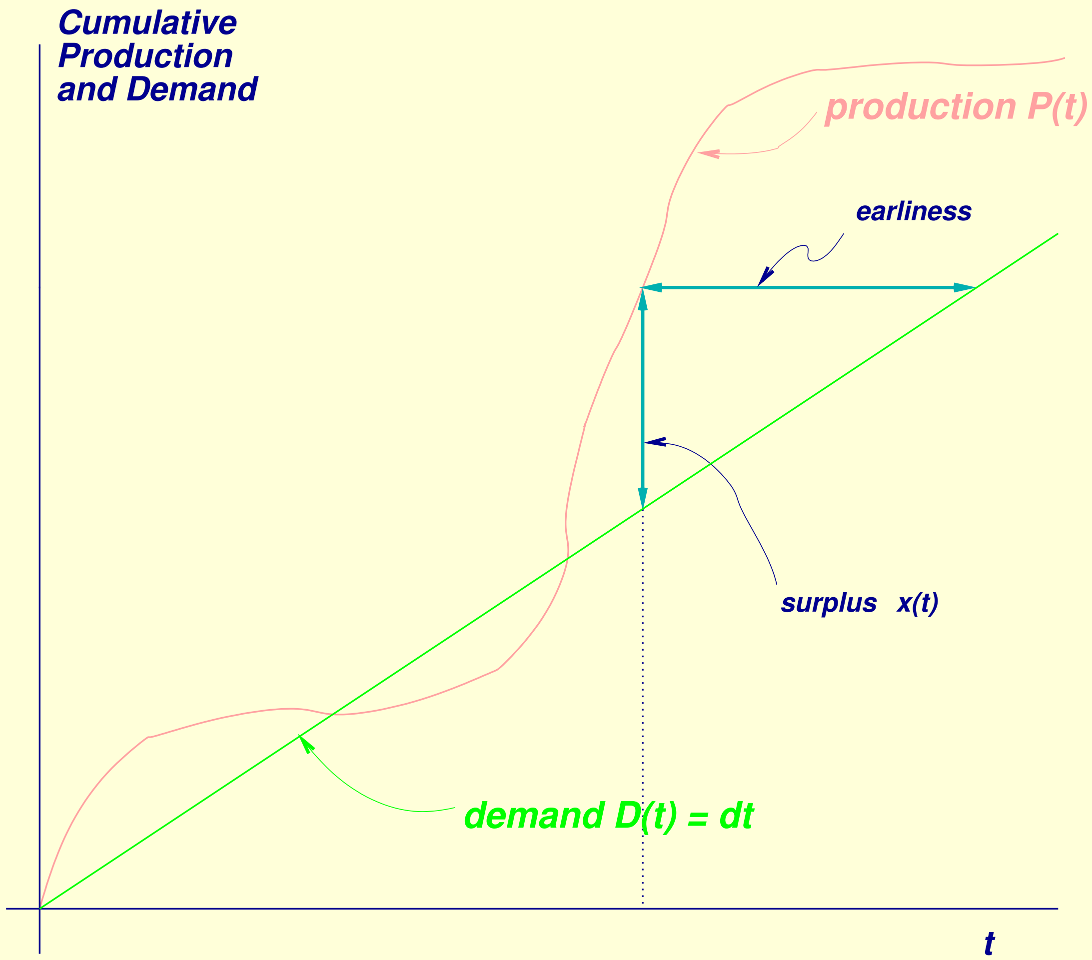
Special case

Bielecki and Kum

- Single machine.
- Single part type.
- Constant demand.
- Exponentially distributed time to fail and ti repair.
- *Problem:* choose the production rate at ev instant to minimize inventory and backlog

Real-time scheduling

Special case



- *State:* $(x$
- x = surplu
between c
productio
- α = mach
$\alpha = 1$ me
is up; $\alpha =$
machine i

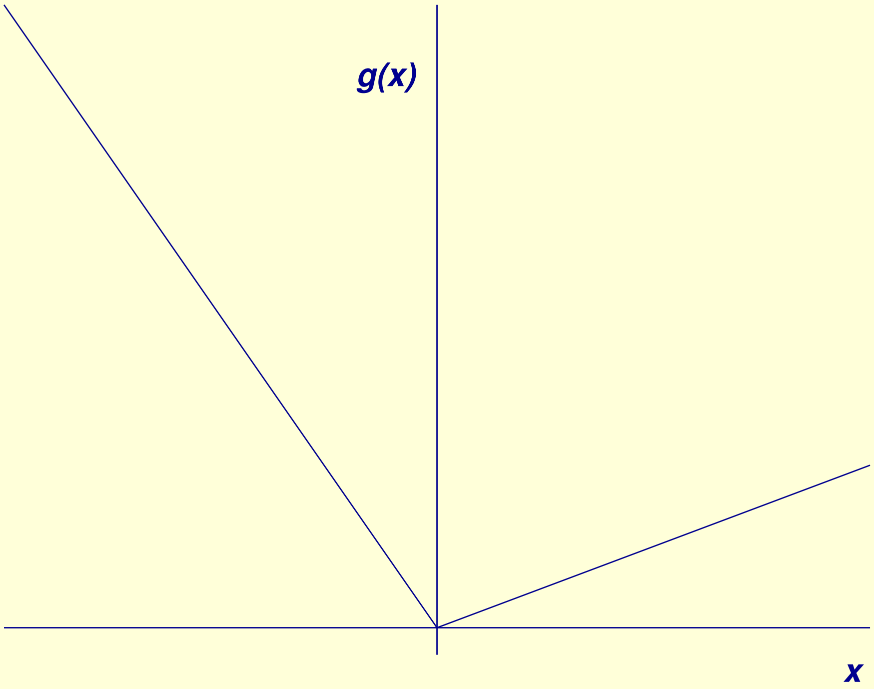
Real-time scheduling

Special case

- *Control:* u
- u = short term production rate. $P(t) = \int_0^t$
- μ is the maximum production rate of the m
  - ⋆ if $\alpha = 1$, $0 \leq u \leq \mu$;
  - ⋆ if $\alpha = 0$, $u = 0$.

Real-time scheduling

Special case



- Objective functio

$$\min E \int_0^T g($$

- where

$$g(x) = \begin{cases} g_+ x, \\ -g_- x \end{cases}$$

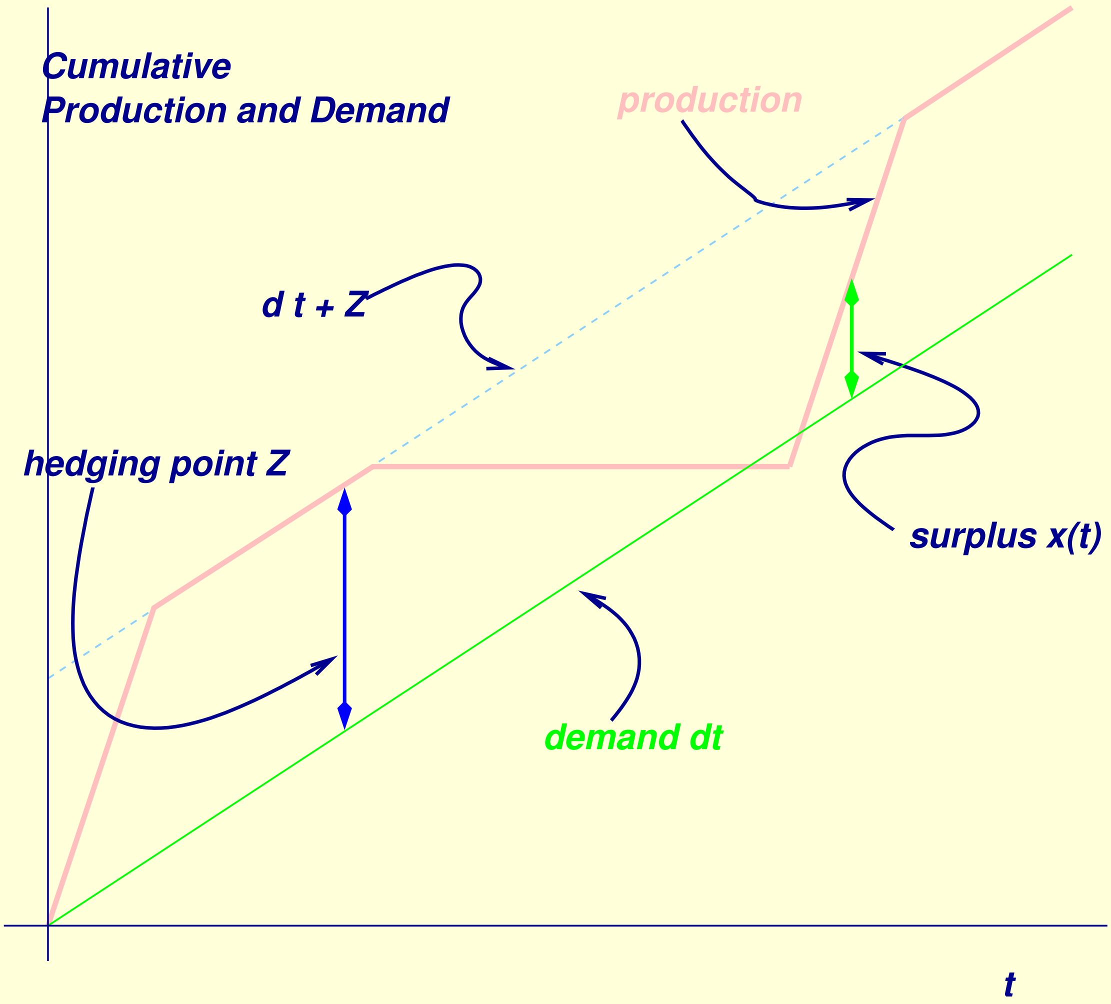
Real-time scheduling

Special case

- Dynamics:
  - $\star$ $\dfrac{dx}{dt} = u - d$
  - $\star$ α goes from 0 to 1 according to an expo distribution with parameter r.
  - $\star$ α goes from 1 to 0 according to an expo distribution with parameter p.

Real-time scheduling

Special case



Solution:

- if $x(t) > Z$
- if $x(t) = Z$
demand ra
- if $x(t) < Z$
maximum

- The hedging point Z is the single control p
- Z represents a trade-off between costs of and risk of disappointing customers.
- Z is a function of d, μ, r, p, g_+, g_-. A form
- *There are no complete solutions for any la systems.*

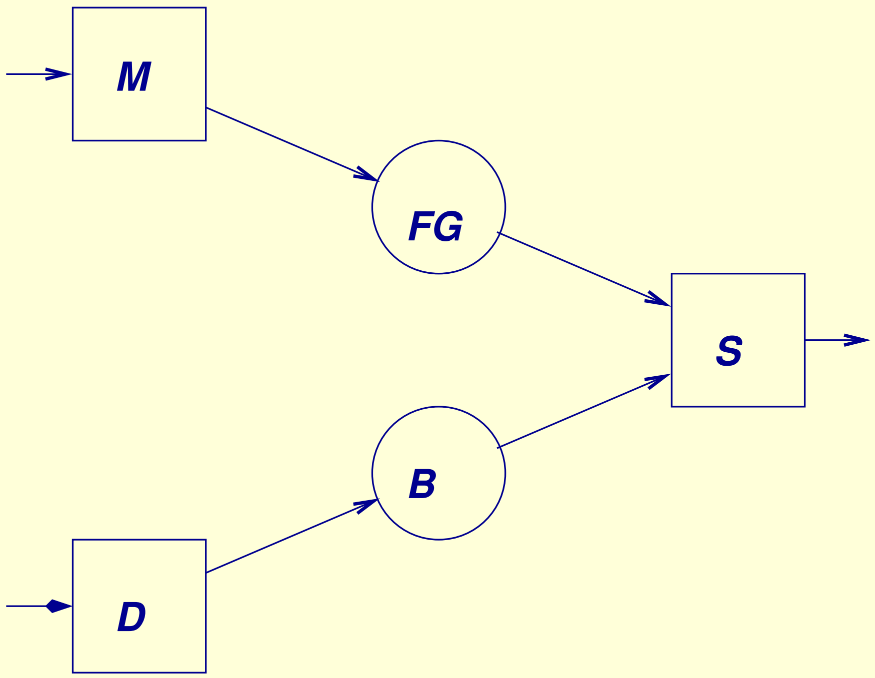
Real-time scheduling

- Proposed approximate solution for multiple
  single machine system:
  - ⋆ *Rank order the part types, and bring the*
    hedging points in that order.
- Proposed approximate solution for multiple
  multiple machine systems:
  - ⋆ *Treat each machine as a multiple-part, s*
    machine system.

Real-time scheduling

Surplus and tok

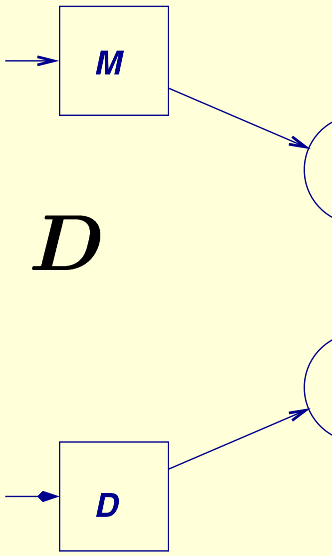
(Gershwin, 20



- *Operating Machi according to the point policy is eq operating this as system accordin buffer policy.*

Surplus and tok

(Gershwin, 20



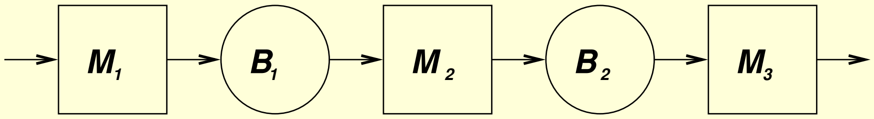
- D is a *demand generator* .
  - ⋆ Whenever a demand arrives, D sends a token to B.
- S is a synchronization machine.
  - ⋆ S is perfectly reliable and infinitely fast.
- FG is a finite finished goods buffer.
- B is an infinite backlog buffer.

Real-time scheduling

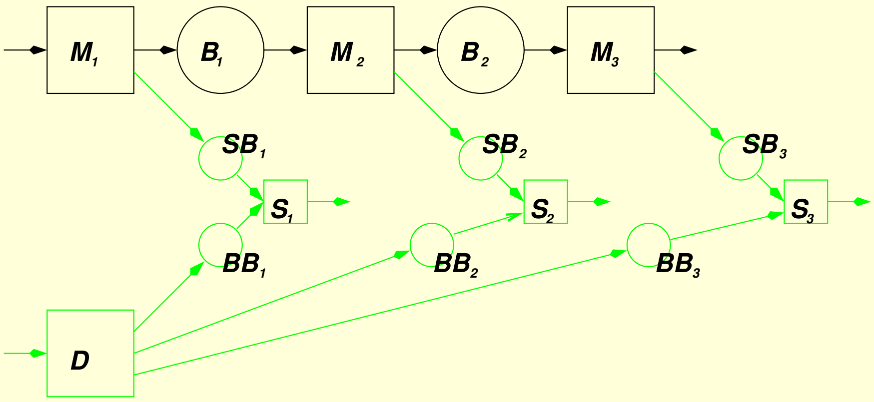
Proposed control p

(Gershwin, 20

To control



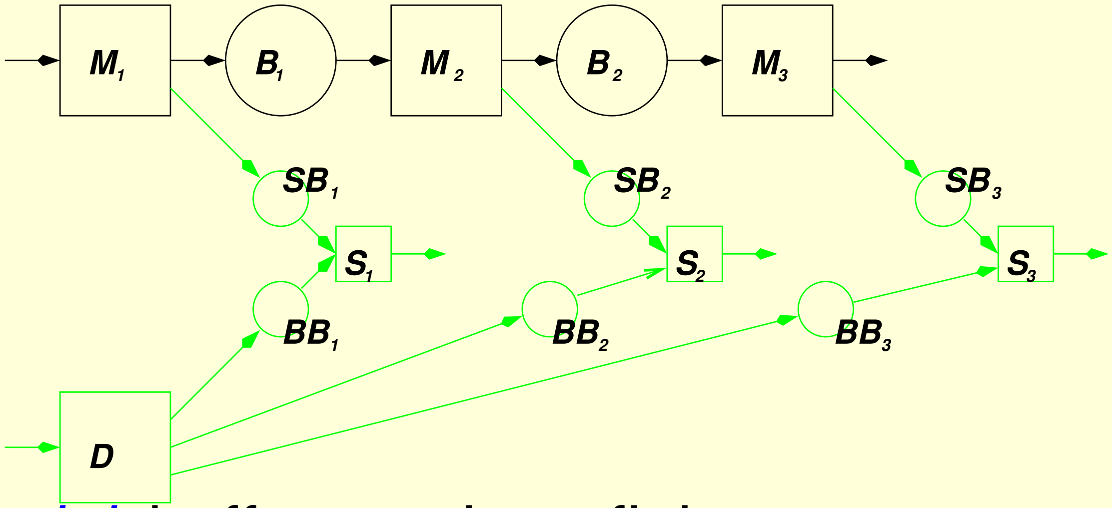
add an information flow system:



Real-time scheduling

Proposed control p

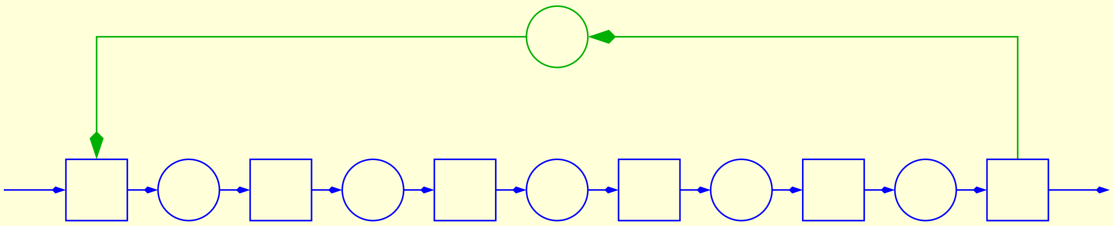
(Gershwin, 20



- B_i are *material* buffers and are finite.
- SB_i are *surplus* buffers and are finite.
- BB_i are *backlog* buffers and are infinite.
- The sizes of B_i and SB_i are control parameters.
- *Problem:* predicting the performance of this syste

Other policies

CONWIP,kanban, a

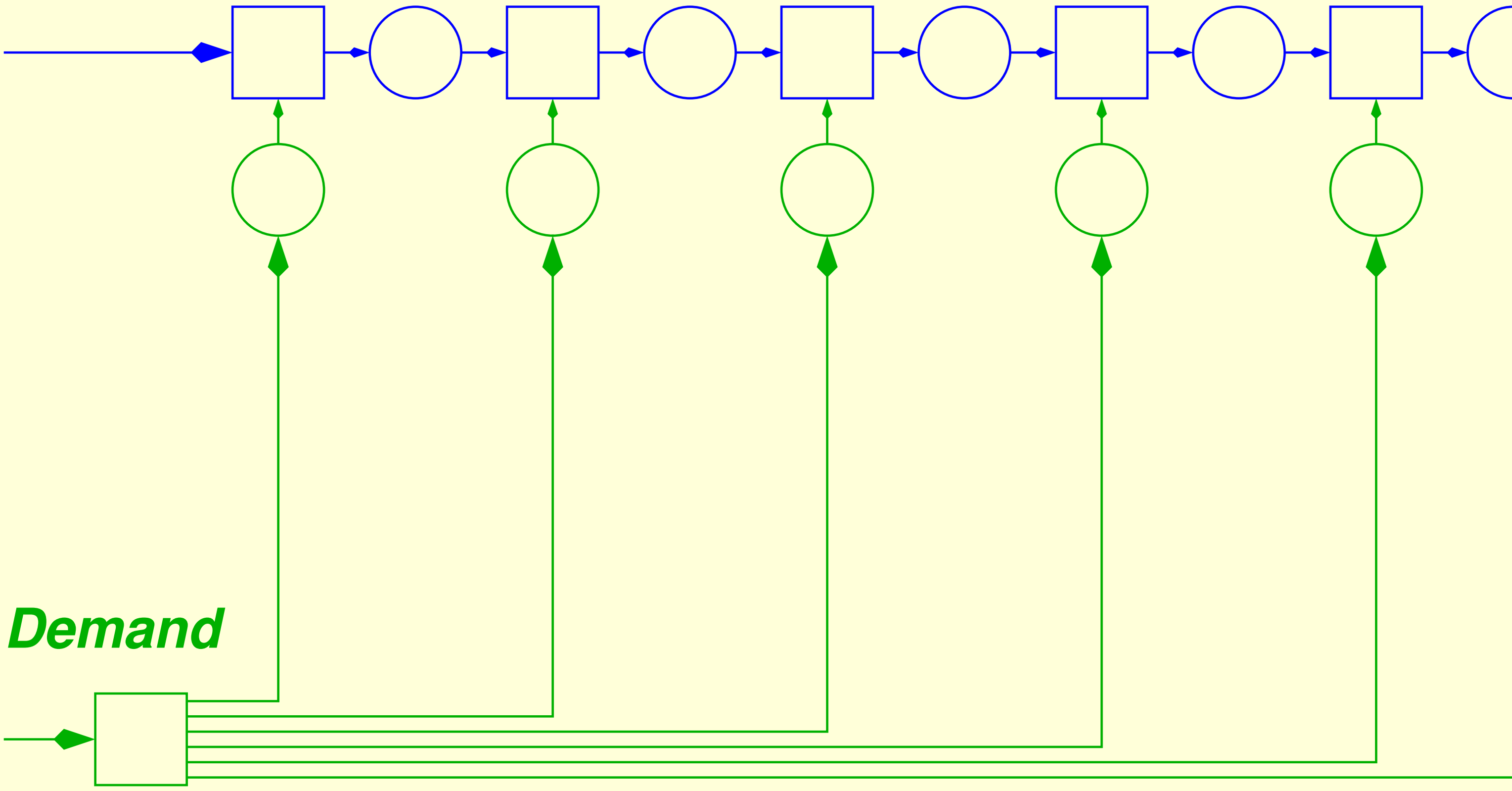


- *CONWIP:* finite population, infinite buffers
- *kanban:* infinite population, finite buffers
- *hybrid:* finite population, finite buffers

Real-time scheduling

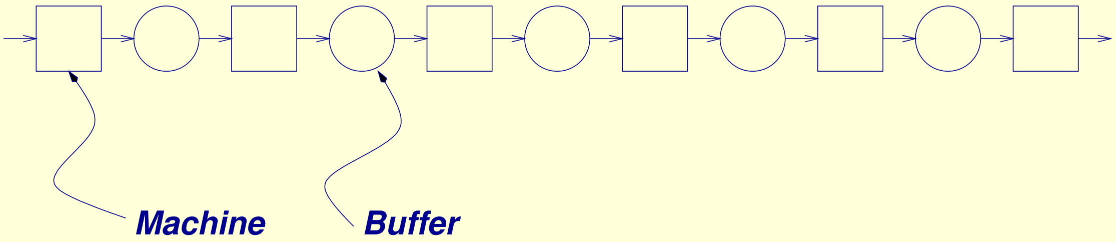
Other policies

Basestock



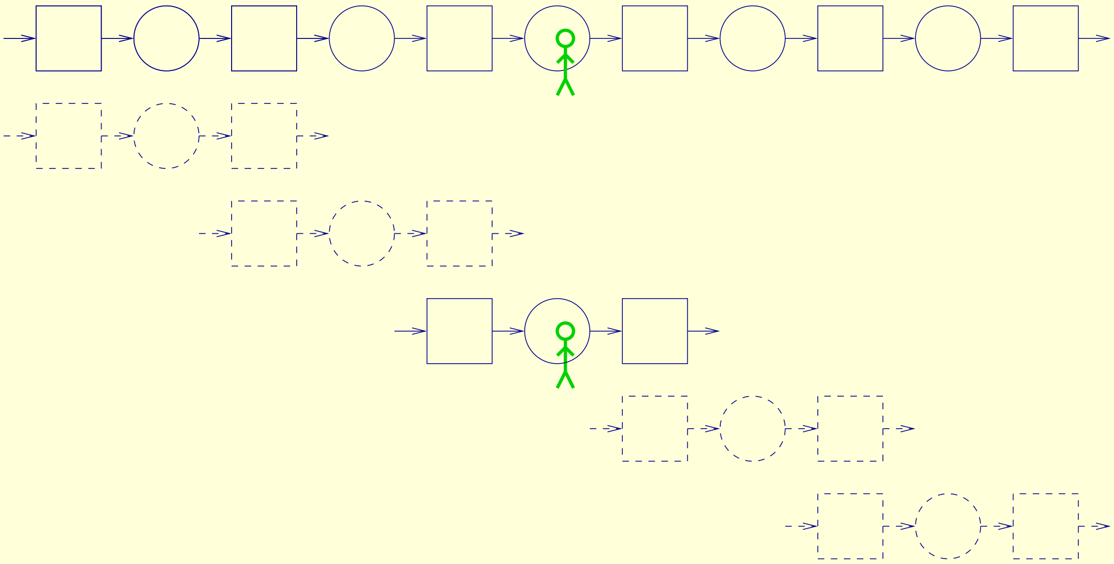
Need

Decomposition



- System can be modeled using Markov chains/proc
- Buffers are *finite* , so
  - ⋆ there is no exact analytical solution, and
  - ⋆ the state space is too large for an exact numerica
    $2^k\Pi_i^{k-1}(N_i + 1)$ states for a k-machine line wit
    N_1, ... N_{k-1}. If $k = 20$ and $N_i = 10$, this is ove
    6.41×10^{25}.
- An approximation is needed.

Decomposition



- Decomposition breaks up systems and then reunit
- Conceptually: put an observer in a buffer, and tell h in the buffer of a two-machine line.
- Question: *What would an observer see, and how c convinced?*

Decomposition

Original versio

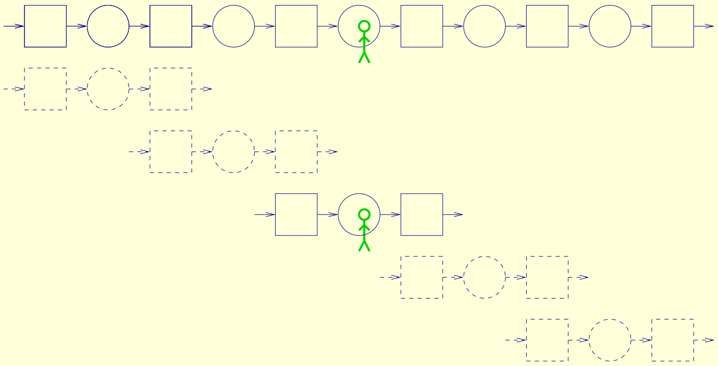
(Gershwin, 19

- Operation time is deterministic and equal to 1 unit machines.
- Each machine can be up or down — two states.
  - ⋆ ... in both the original line and in each two-machi building block.
  - ⋆ Time to fail and time to repair are geometrically d
  - ⋆ Each machine has two parameters:
    - $*$ p = failure probability = 1/MTTF,
    - $*$ r = repair probability = 1/MTTR.

Decomposition

Original versio

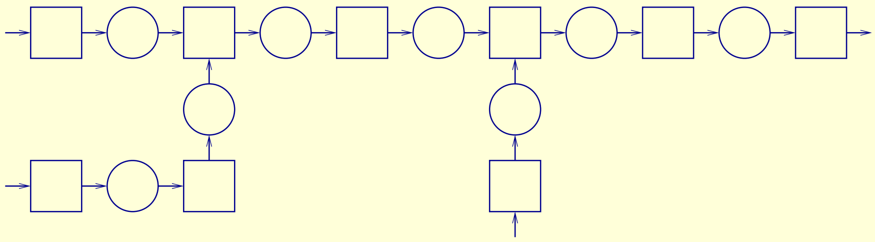
(Gershwin, 19



- There are $k - 1$ two-machine lines and 4 paramet so we need $4(k - 1)$ equations.
- Note: *not* a function of N_i. If $k = 20$, this is 76.
- One set of equations *(conservation of flow)* say tha rates through all the small lines must be the same.
- Others take longer to describe.
- Equations are nonlinear, but efficient algorithms ex

Decomposition

- Different processing times.
- Tree-structured assembly/disassembly sys

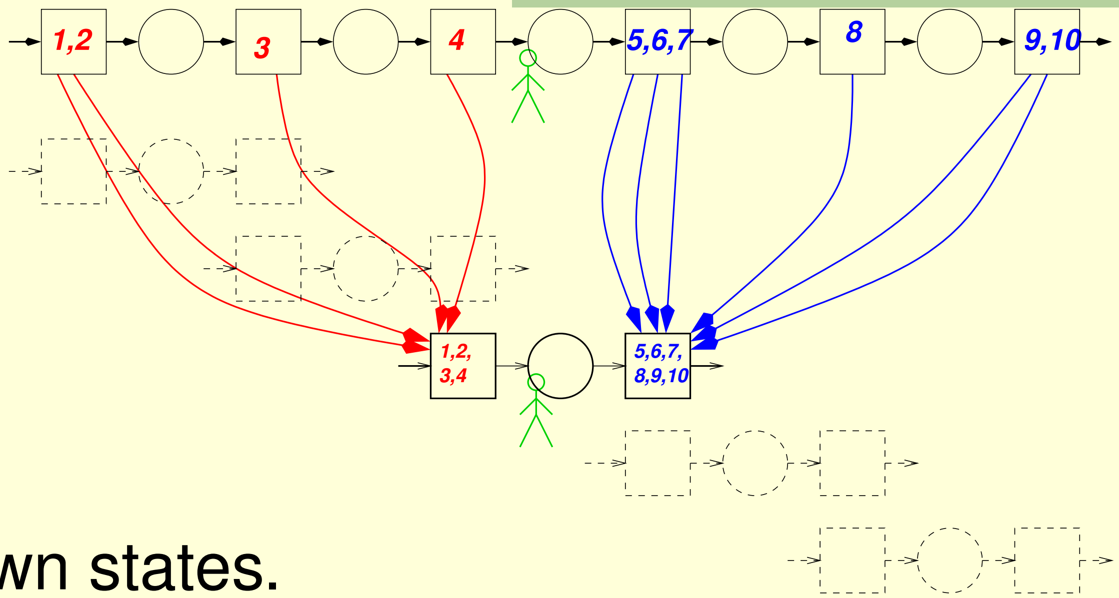


- Imperfect yield.

Decomposition

Tolio decomposi

Tolio and Matta

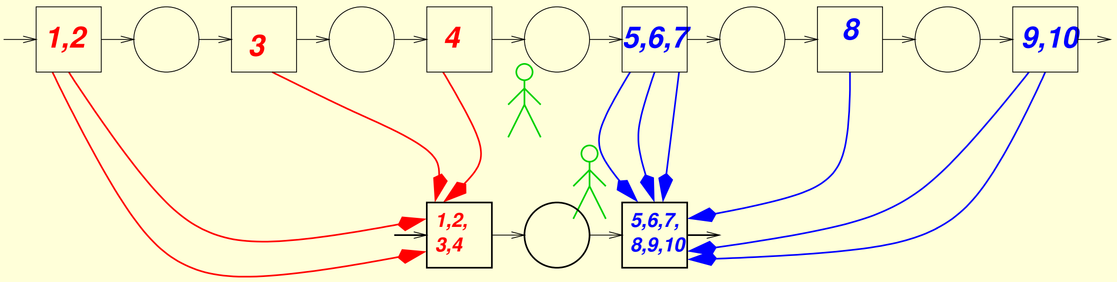


- Multiple down states.
- Needed because original decomposition does not
information.
- *However,* more than two parameters per machine

Decomposition

Tolio decomposi

Tolio and Matta

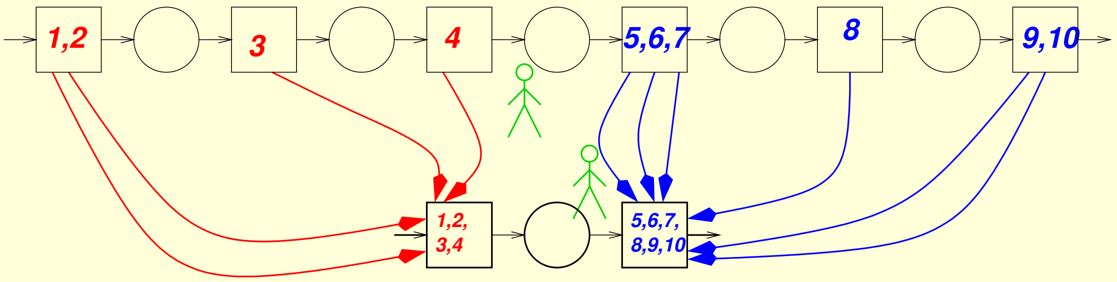


- Each machine in the original line *may* and in the tw
lines *must* have multiple failure modes.
- For each failure mode downstream of a given buffe
corresponding mode in the downstream machine o
two-machine line.
- Similarly for upstream modes.

Decomposition

Tolio decomposi

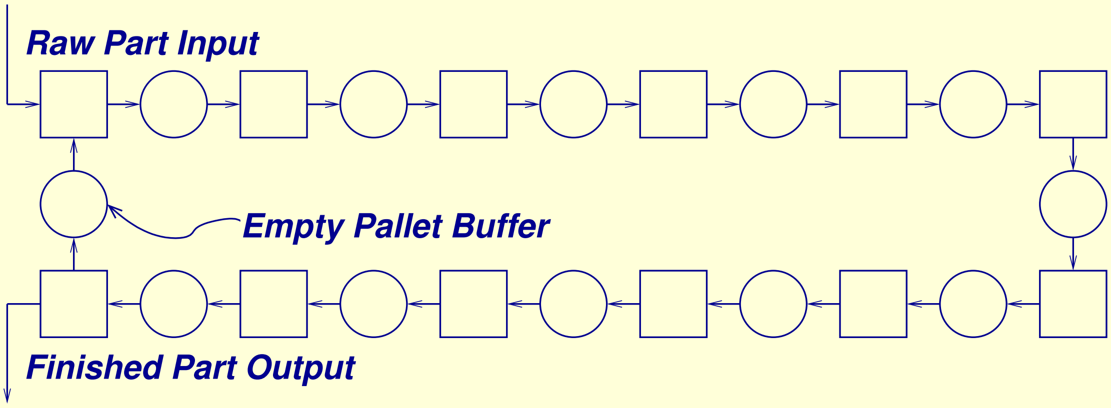
Tolio and Matta



- The downstream failure modes appear to the obse propagation through *blockage* .
- The upstream failure modes appear to the observe propagation through *starvation* .
- The two-machine lines are more complex that in ea decompositions but the decomposition equations a

Loops

- Many systems have a loop structure.

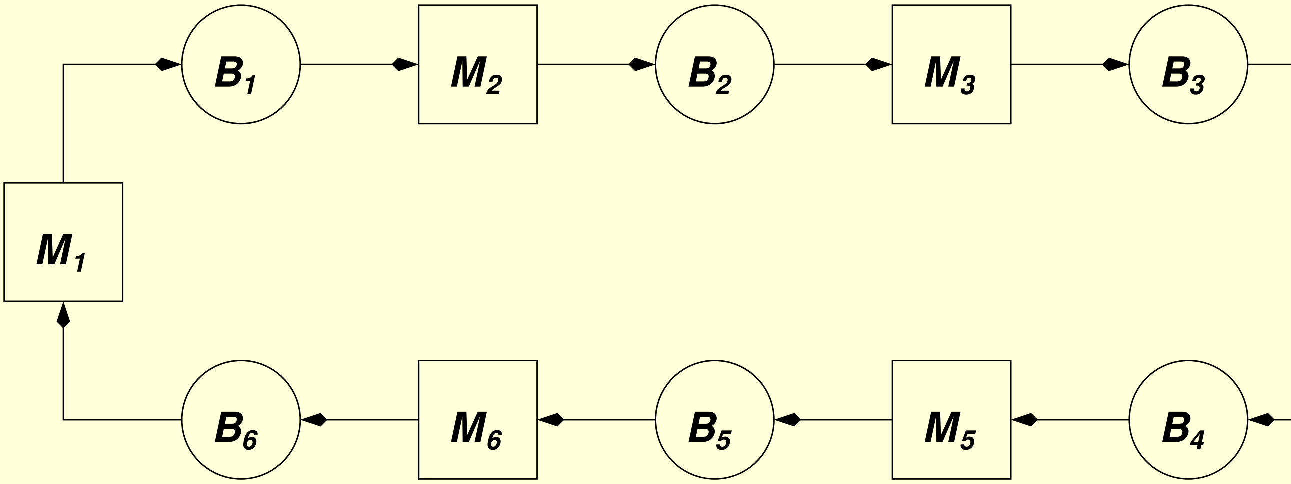


- Control policies involve loops.

Loops

Single-loop syst

Problem State



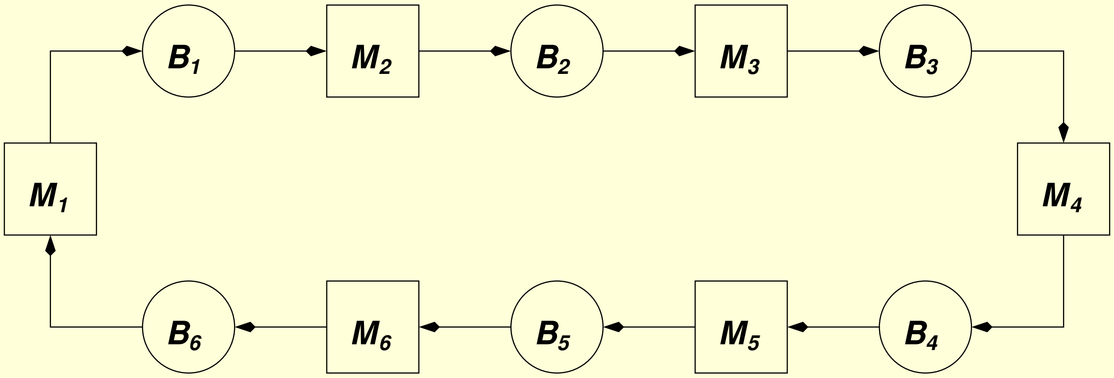
- Finite buffers $(0 \leq n_i(t) \leq N_i)$.
- Closed loop — fixed population $(\sum_i n_i(t) = N)$ –
- Buzacott model (deterministic processing time; geo and down times). Repair probability = r_i failure pro
- *Goal:* calculate production rate and inventory distr

- *Desire:* a decomposition method similar t the line, which can be extended to multiple systems.
- *Difficulty:* the correlation among buffer lev
  - ⋆ A method exists which is based on the o decomposition (Frein, Commault, and Da 1996) but is only accurate for large loops

Loops

Single-loop syst

Ranges

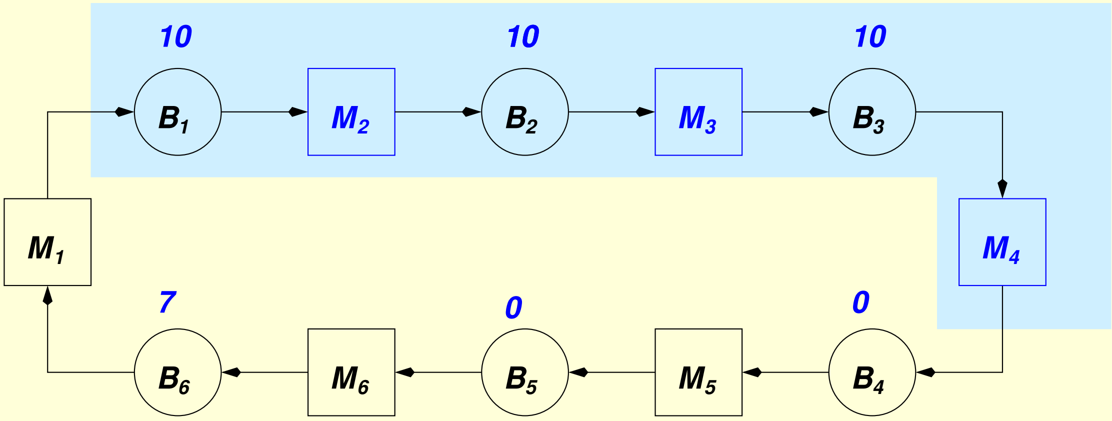


- The *range of blocking of a machine* is the set of al
  that could block it if they stayed down for a long en
- The *range of starvation of a machine* is the set of
  that could starve it if they stayed down for a long e

Loops

Ranges

Range of Bloc

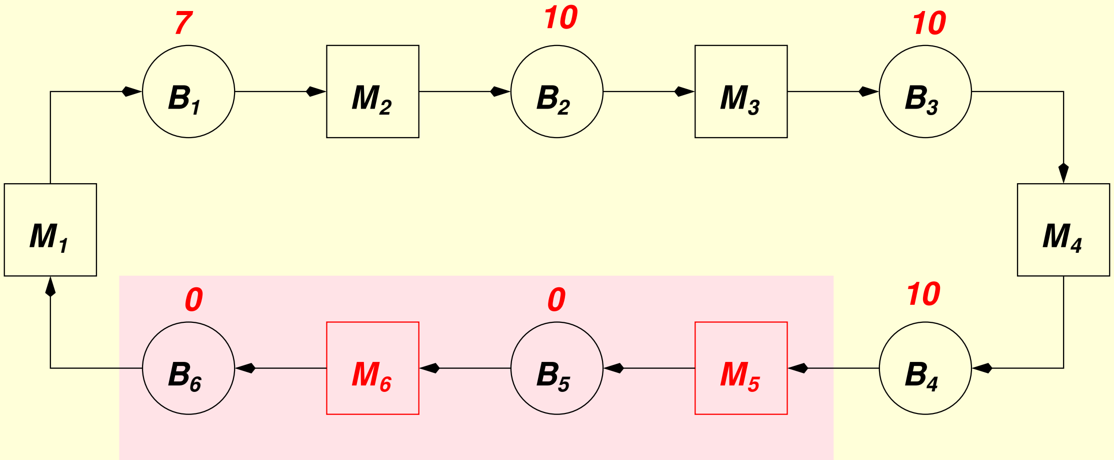


- All buffer sizes are 10.
- Population is 37.
- If M_4 stays down for a long time, it will block M_1.
- Therefore M_4 is in the range of blocking of M_1.
- Similarly, M_2 and M_3 are in the range of blocking

Loops

Ranges

Range of starv

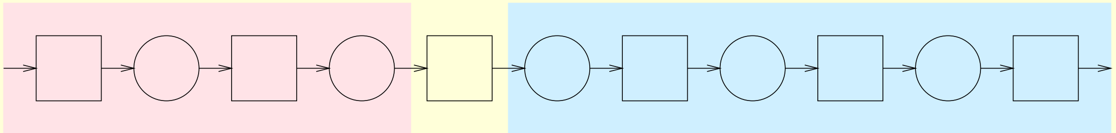


- If M_5 stays down for a long time, it will starve M_1.
- Therefore M_5 is in the range of starvation of M_1.
- Similarly, M_6 is in the range of starvation of M_1.

Loops

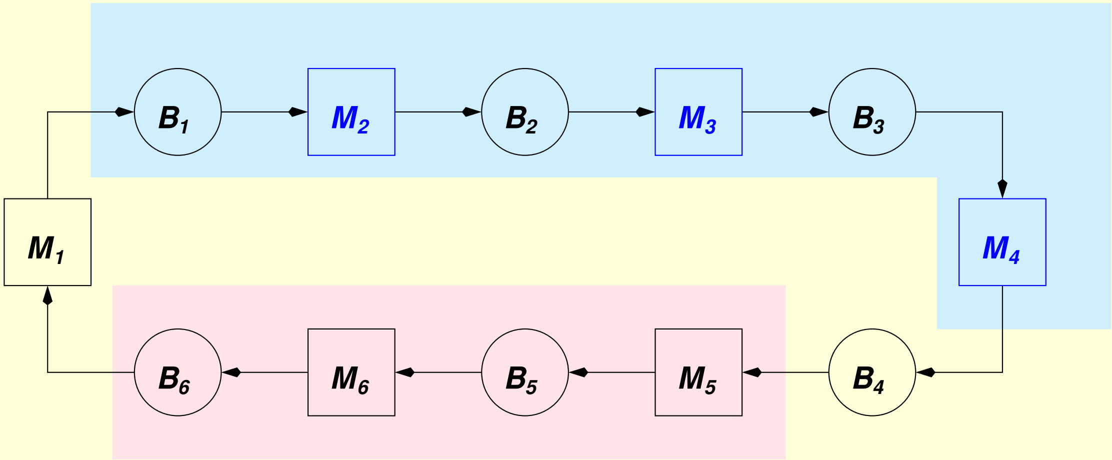
Ranges

Line



- The range of blocking of a machine in a line is the downstream part of the line.
- The range of starvation of a machine in a line is the upstream part of the line.

Loops



- Use the Tolio decomposition, but adjust the ranges
  and starvation accordingly.
- Split buffers and introduce perfectly reliable machi
  the total population fits perfectly in a sequence of b
  starting at each machine.

Loop decomposition

Numerical resu

Accuracy

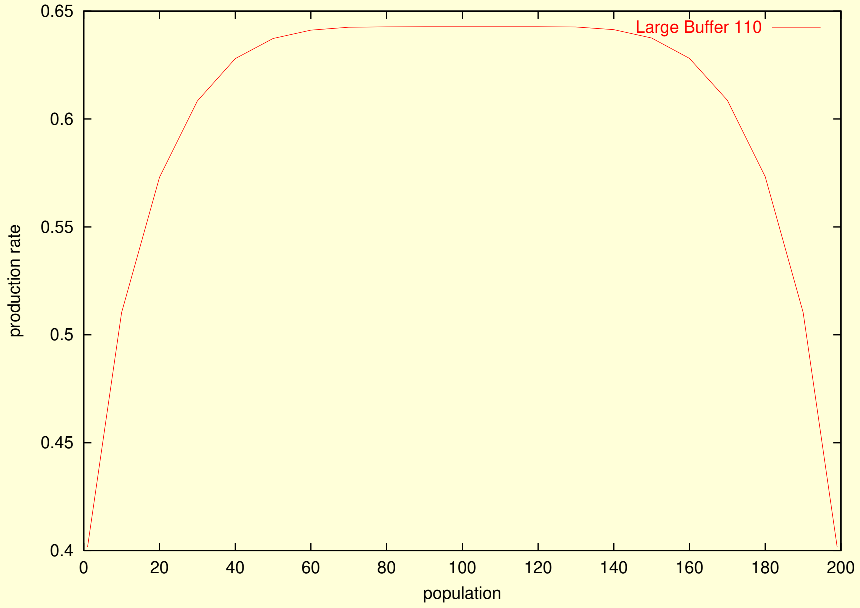
- Many cases were compared with simulatio
  - ⋆ *Three-machine cases:* all throughput errors und level errors averaged 3%, but were as high as 10
  - ⋆ *Six-machine cases:* mean throughput error 1.1% maximum of 2.7%; average buffer level error 5% maximum of 21%.
  - ⋆ *Ten-machine cases:* mean throughput error 1.4% maximum of 4%; average buffer level error 6% w maximum of 44%.

- Convergence reliability: almost always.
- Speed: execution time increases rapidly w
size.
- Maximum size system: growing each time
implement.

Loop decomposition

Numerical resu

Loop behav

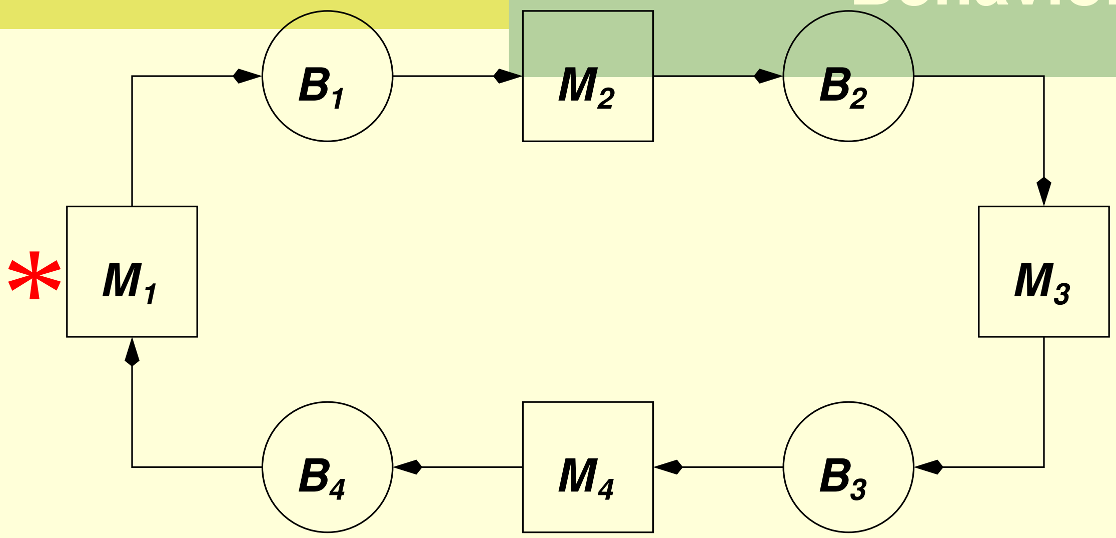


- Production rate when population is traffic congestion.

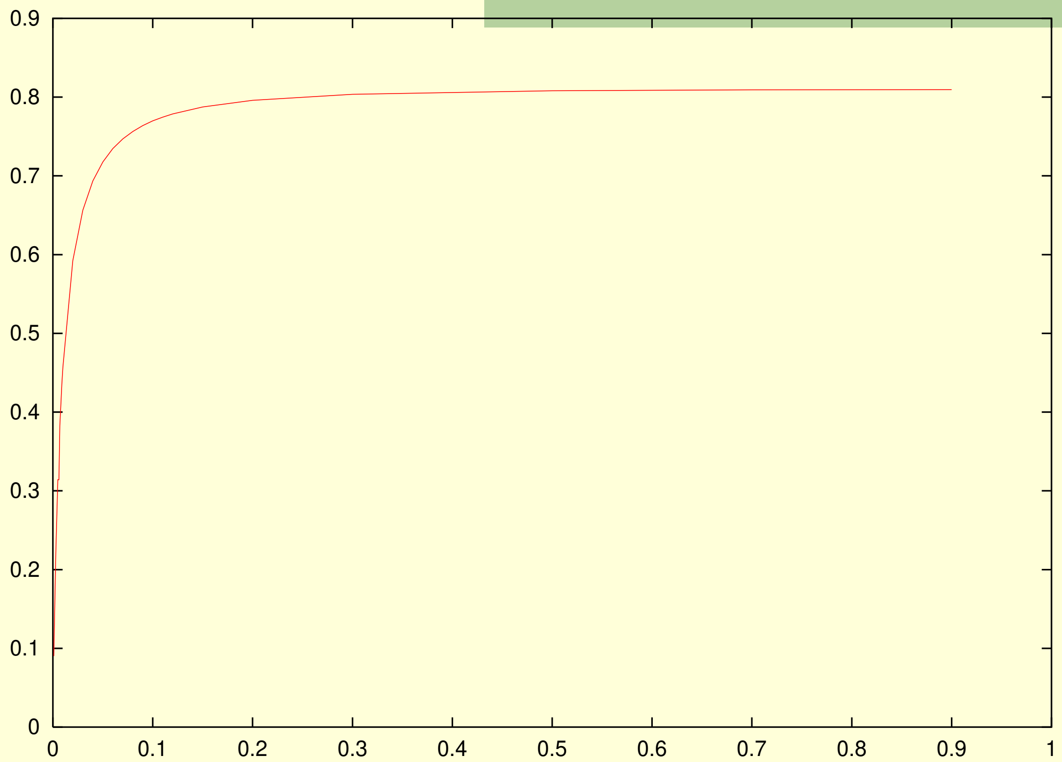
Loop decomposition

Numerical resu

Behavior



- All buffer sizes 10. Population 15. Identica except for M_1.
- Observe average buffer levels and produc a function of r_1.

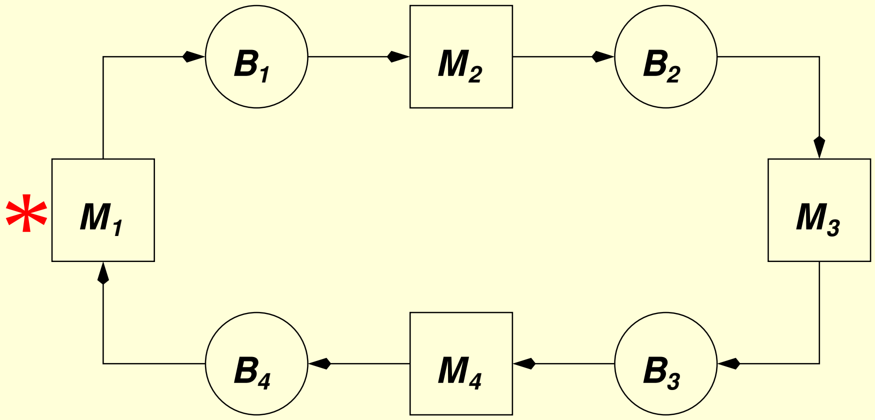


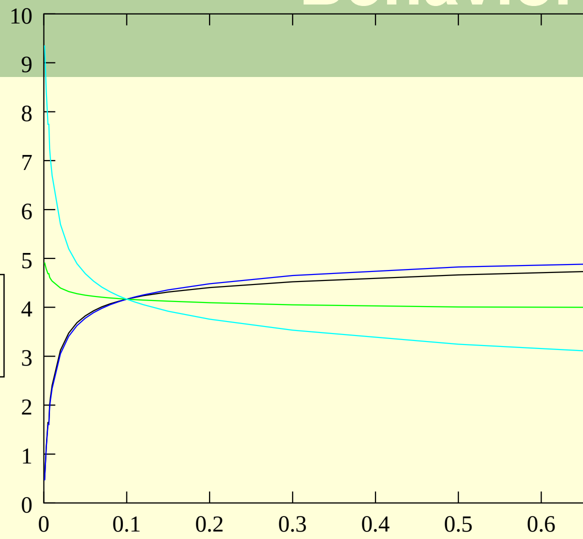
- Usual saturating graph.

Loop decomposition

Numerical resu

Behavior



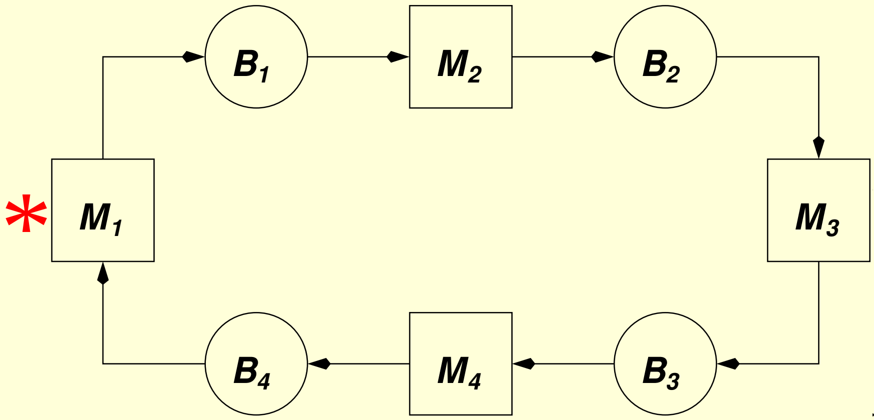


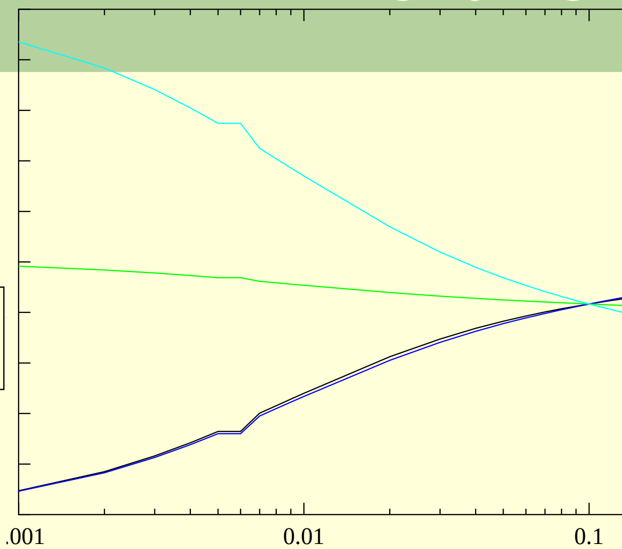
- When r_1 is small, M_1 is a bottleneck, so B
  parts, B_3 holds 5 parts, and the others are
- As r_1 increases, material is more evenly d
  When $r_1 = 0.1$, the network is totally sym

Loop decomposition

Numerical resu

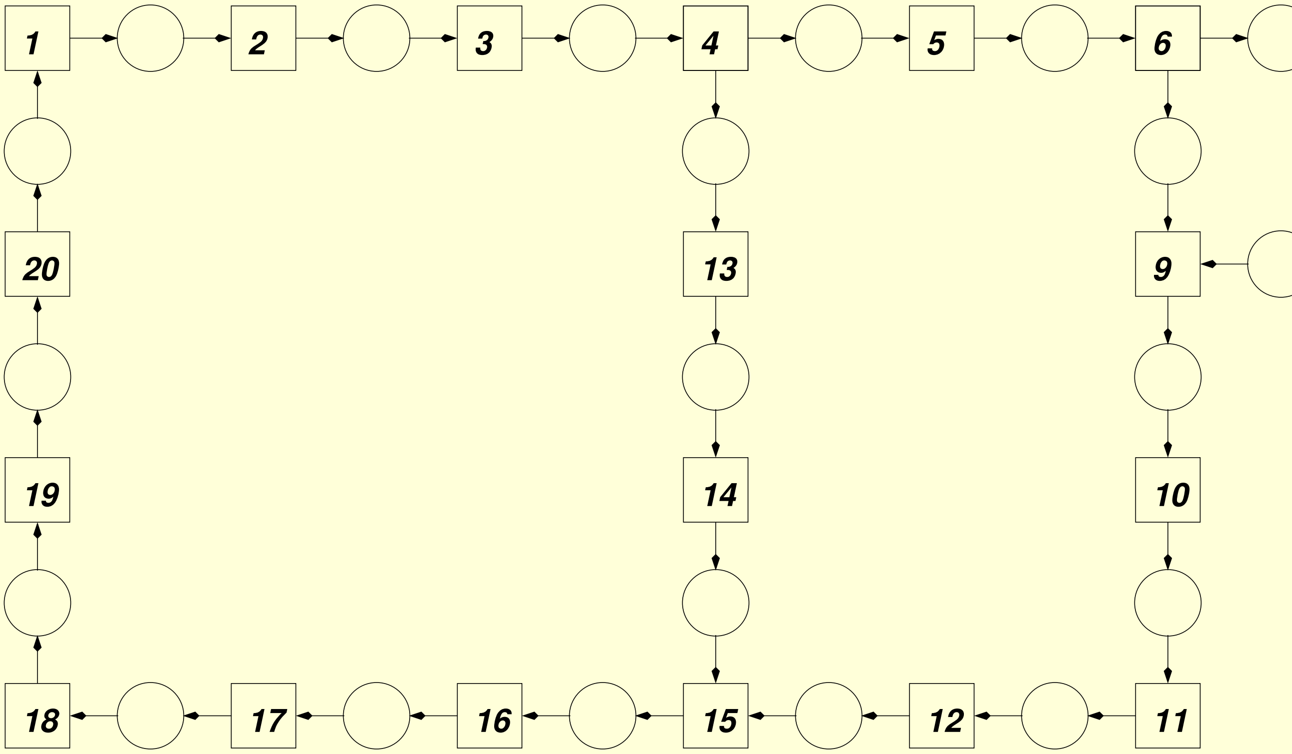
Behavior





- When r_1 is small, M_1 is a bottleneck, so B
  parts, B_3 holds 5 parts, and the others are
- As r_1 increases, material is more evenly d
  When $r_1 = 0.1$, the network is totally sym

Multiple-loop systems



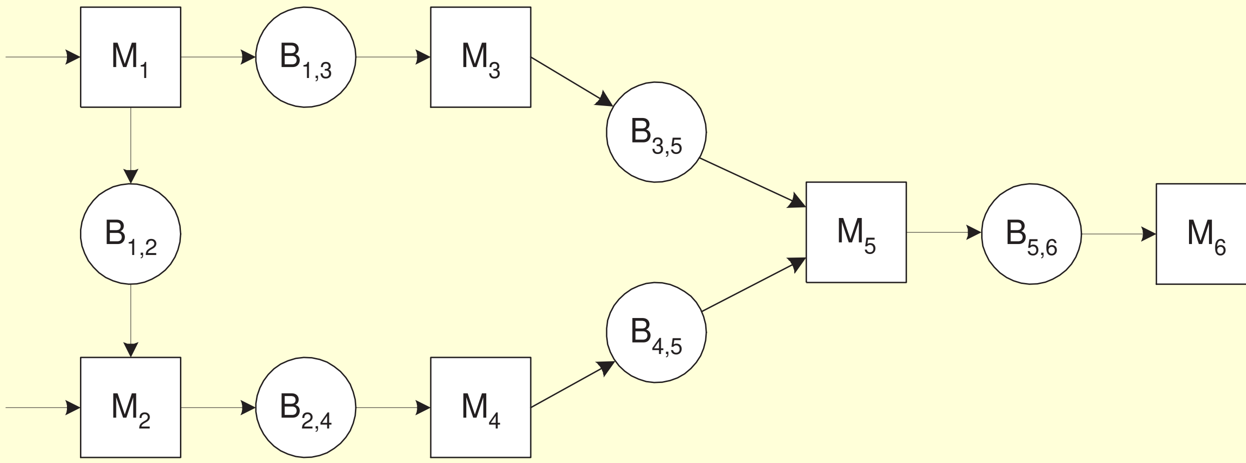
• Ranges of blocking and starvation are more compl
possible to determine.

• Algorithm is an extension of single-loop algorithm.

Multiple-loop systems

- One *invariant* for each independent loop.
- Ranges of starvation and blockage need n contiguous.
- More buffers may have to be split and mor machines introduced (one set for each ind loop).

Multiple-loop systems



- Addition of token buffer $B_{1,2}$ reduces real inventor
- Without $B_{1,2}$ and with all buffer sizes 10, productio
  0.6387 and average inventory = 26.84.
- With $B_{1,2} = 30$, and with $B_{1,3} = 11, B_{3,5} = 13,$
  production rate = 0.6405, average inventory = 22.2

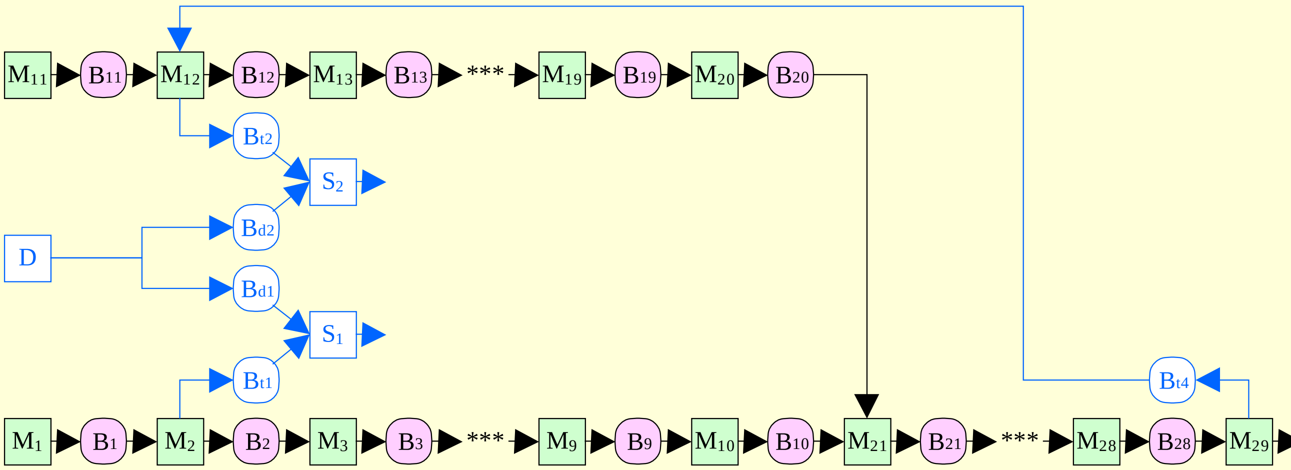
Multiple-loop systems

To develop new control policies:

- Start with the physical network and add th
  flow network.
- Optimize over buffer sizes and invariants.
- Consider alternative token flow networks –
  over structure.

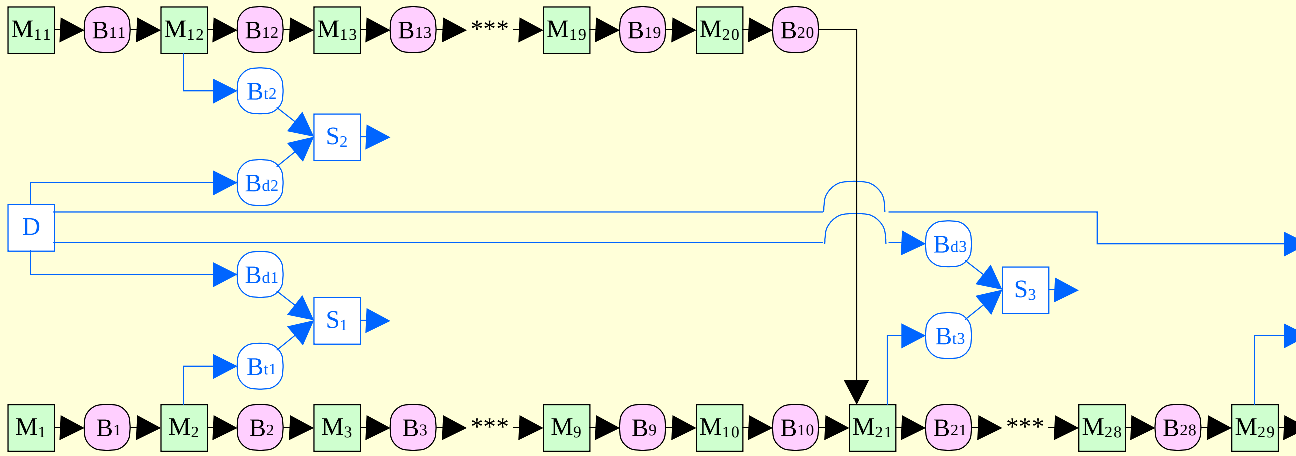
Multiple-loop systems

Control applicati



Multiple-loop systems

Control applicati



- Algorithm has worked successfully for this

Conclusions and Future Research

- Comparison with simulation
- Numerical experimentation
- Optimization of buffer sizes
- Multiple part types
- Real world implementation experiments