MIT 2.852
Manufacturing Systems Analysis
Lectures 19–21

Scheduling: Real-Time Control of Manufacturing Systems

Stanley B. Gershwin

Spring, 2007

Definitions

- Events may be *controllable* or not, and *predictable* or not.

	controllable	*uncontrollable*
predictable	loading a part	lunch
unpredictable	???	machine failure

Definitions

- *Scheduling is the selection of times for future controllable events.*
- Ideally, scheduling systems should deal with *all* controllable events, and not just production.
  - ⋆ That is, they should select times for operations, set-up changes, preventive maintenance, etc.
  - ⋆ They should at least be *aware* of set-up changes, preventive maintenance, etc.when they select times for operations.

Definitions

- Because of recurring random events, scheduling is an on-going process, and not a one-time calculation.

- Scheduling, or shop floor control, is the bottom of the scheduling/planning hierarchy. It translates *plans* into *events.*

Issues in Factory Control

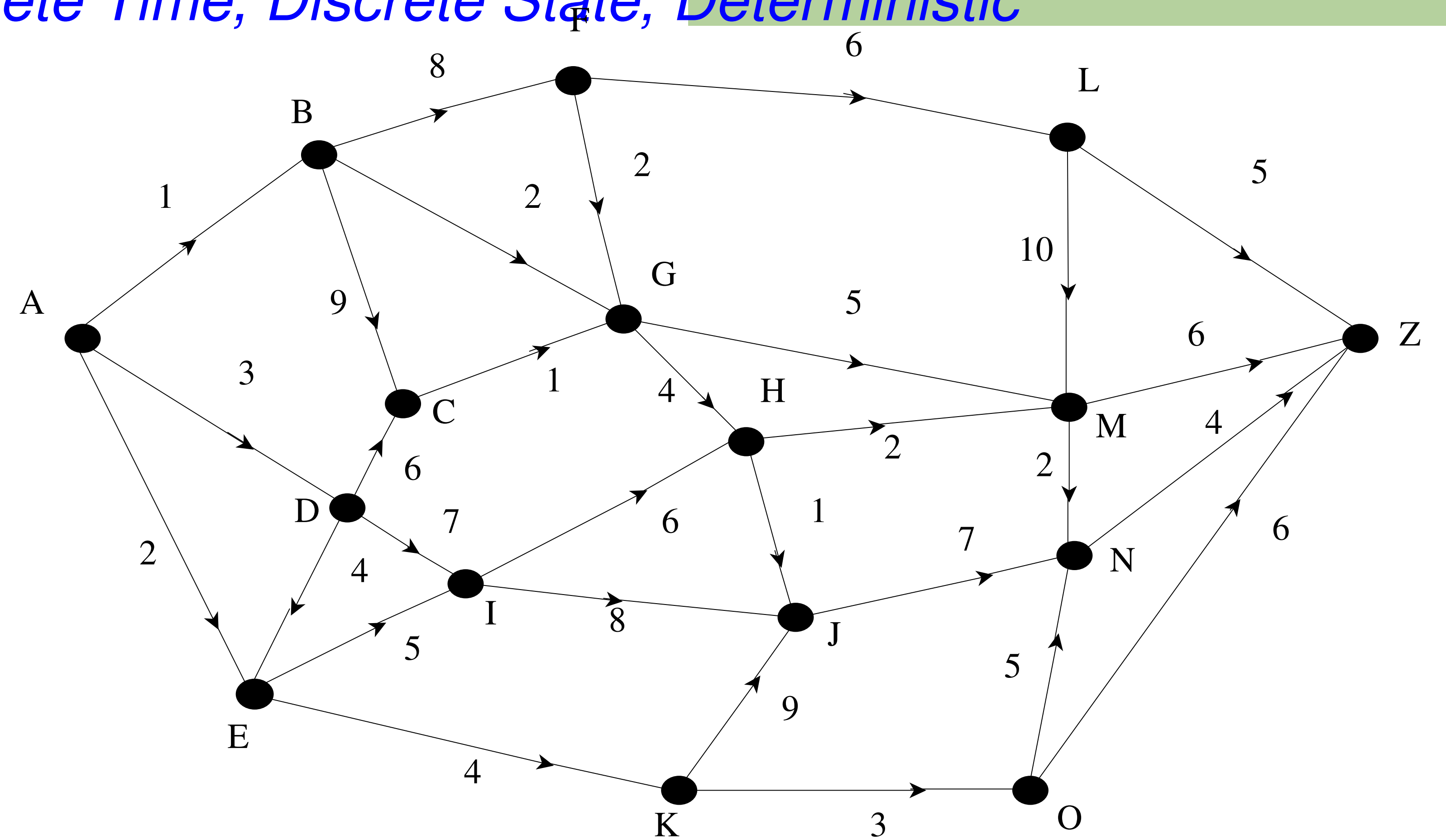
- Problems are *dynamic* ; current decisions influence future behavior and requirements.
- There are large numbers of parameters, time-varying quantities, and possible decisions.
- Some time-varying quantities are *stochastic* .
- Some relevant information (MTTR, MTTF, amount of inventory available, etc.) is not known.
- Some possible control policies are *unstable* .

Dynamic Programming

Example

Problem

Discrete Time, Discrete State, Deterministic



Problem: find the least expensive path from A to Z.

Dynamic Programming

Example

Problem

Let $g(i,j)$ be the cost of traversing the link from i to j. Let $i(t)$ be the tth node on a path from A to Z. Then the path cost is

$$\sum_{t=1}^{T} g(i(t-1), i(t))$$

where T is the number of nodes on the path, $i(0) = A$, and $i(T) = Z$.

T is not specified; it is part of the solution.

Dynamic Programming

Example

Solution

- A possible approach would be to enumerate all possible paths (possible solutions). However, there can be a lot of possible solutions.
- Dynamic programming reduces the number of possible solutions that must be considered.
  - ⋆ *Good news:* it often *greatly* reduces the number of possible solutions.
  - ⋆ *Bad news:* it often does not reduce it enough to give an exact optimal solution practically (ie, with limited time and memory). This is the *curse of dimensionality* .
  - ⋆ *Good news:* we can learn something by characterizing the optimal solution, and that sometimes helps in getting an analytical optimal solution or an approximation.
  - ⋆ *Good news:* it tells us something about stochastic problems.

Dynamic Programming

Example

Solution

Instead of solving the problem only for A as the initial point, we solve it for *all* possible initial points.

For every node i, define $J(i)$ to be the *optimal cost to go* from Node i to Node Z (the cost of the optimal path from i to Z).

We can write

$$J(i) = \sum_{t=1}^{T} g(i(t-1), i(t))$$

where $i(0) = i$; $i(T) =$Z; $(i(t-1), i(t))$ is a link for every t.

Dynamic Programming

Example

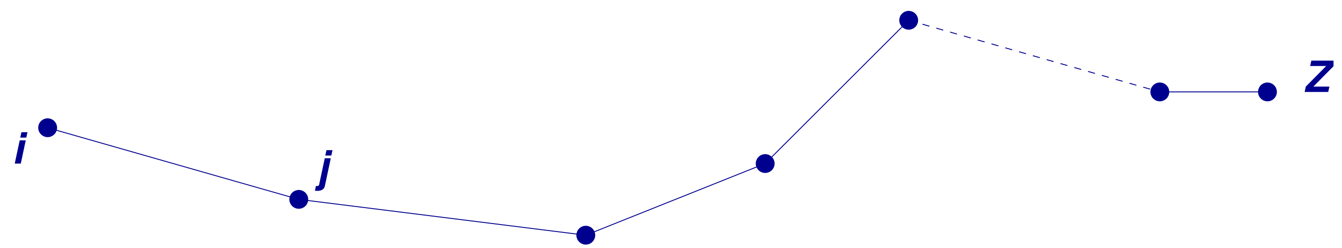
Solution

Then $J(i)$ satisfies

$$J(Z) = 0$$

and, if the optimal path from i to Z traverses link (i, j),

$$J(i) = g(i, j) + J(j).$$

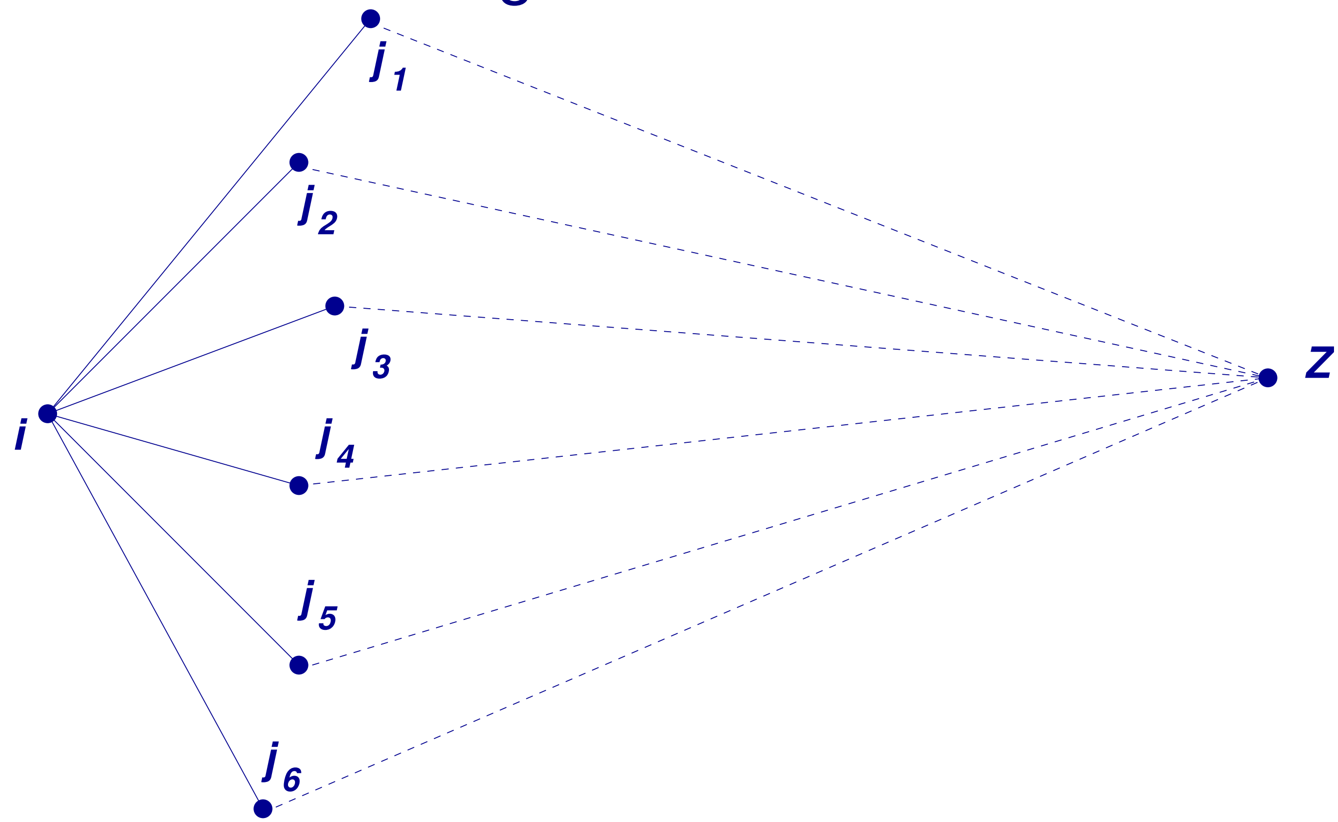


Dynamic Programming

Example

Solution

Suppose that several links go out of Node i.



Suppose that for each node j for which a link exists from i to j, the optimal path and optimal cost $J(j)$ from j to Z is known.

Dynamic Programming

Example

Solution

Then the optimal path from i to Z is the one that minimizes the sum of the costs from i to j and from j to Z. That is,

$$J(i) = \min_j \left[g(i,j) + J(j) \right]$$

where the minimization is performed over all j such that a link from i to j exists. This is the *Bellman equation* .

This is a *recursion* or *recursive equation* because $J()$ appears on both sides, although with different arguments.

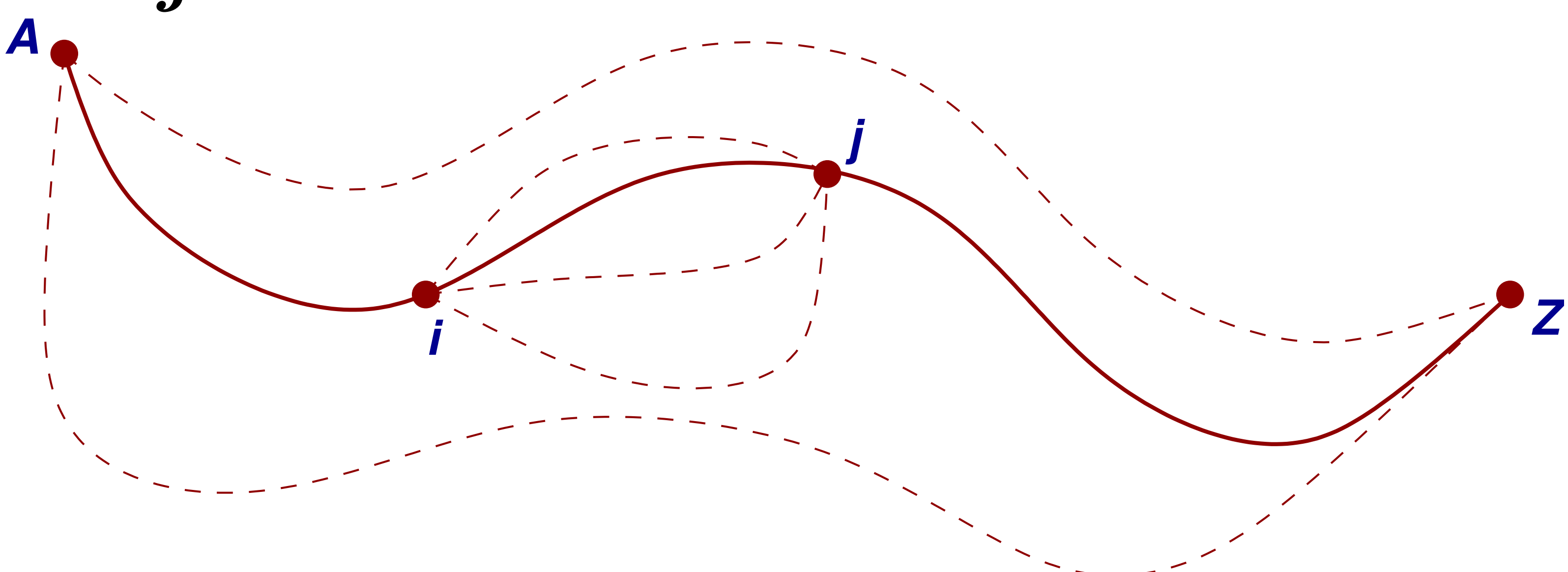
$J(i)$ can be calculated from this if $J(j)$ is known for every node j such that (i, j) is a link.

Dynamic Programming

Example

Solution

Bellman's Principle of Optimality: if i and j are nodes on an optimal path from A to Z, then the portion of that path from A to Z between i and j is an optimal path from i to j.



Example: Assume that we have determined that $J(O) = 6$ and $J(J) = 11$.

To calculate $J(K)$,

$$J(K) = \min \left\{ \begin{array}{l} g(K,O) + J(O) \\ g(K,J) + J(J) \end{array} \right\}$$

$$= \min \left\{ \begin{array}{l} 3 + 6 \\ 9 + 11 \end{array} \right\} = 9.$$

Dynamic Programming

Example

Solution

Algorithm

1. Set $J(Z) = 0$.
2. Find some node i such that
   - $J(i)$ has not yet been found, and
   - for each node j in which link (i, j) exists, $J(j)$ is already calculated.

   Assign $J(i)$ according to

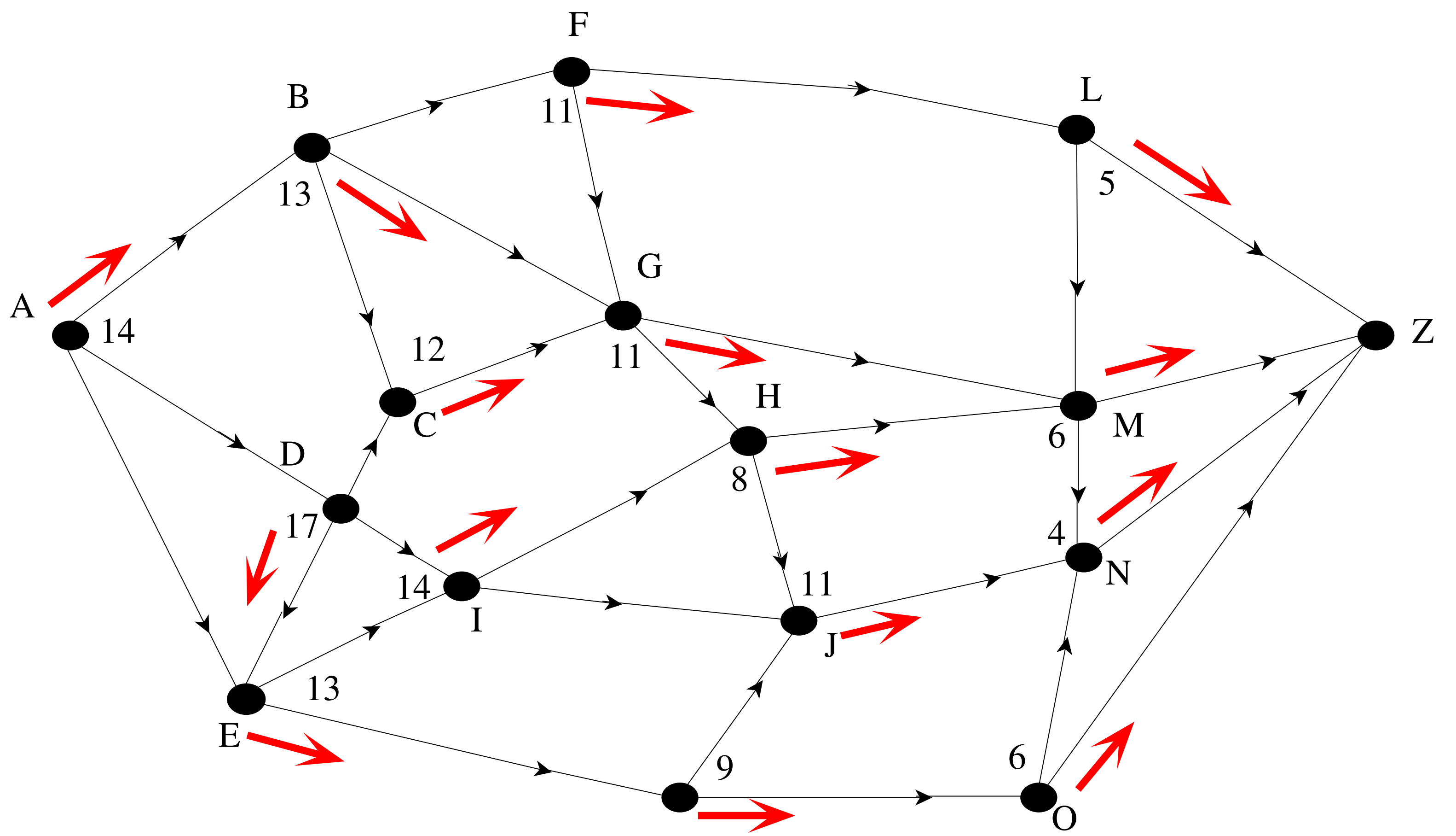
$$J(i) = \min_j \left[g(i, j) + J(j) \right]$$

3. Repeat Step 2 until all nodes, including A, have costs calculated.

Dynamic Programming

Example

Solution



The important features of a dynamic programming problem are

- *the state (i)* ;
- *the decision* (to go to j after i);
- the objective function $\left(\sum_{t=1}^{T} g(i(t-1), i(t))\right)$
- *the cost-to-go function ($J(i)$)* ;
- *the one-step recursion equation that determines $J(i)$* $(J(i) = \min_j [g(i,j) + J(j)])$;
- *that the solution is determined for every i,* not just A and not just nodes on the optimal path;
- *that $J(i)$ depends on the nodes to be visited after i,* not those between A and i. The only thing that matters is the present state and the future;
- *that $J(i)$ is obtained by working backwards.*

Dynamic Programming

Example

Solution

This problem was

- discrete time, discrete state, deterministic.

Other versions:

- discrete time, discrete state, stochastic
- continuous time, discrete state, deterministic
- continuous time, discrete state, stochastic
- continuous time, mixed state, deterministic
- continuous time, mixed state, stochastic

in stochastic systems, we optimize the *expected* cost.

Dynamic Programming

Discrete time, discrete state

Stochastic

Suppose

- $g(i, j)$ is a random variable; or
- if you are at i and you choose j, you actually go to k with probability $\mathrm{p}(i, j, k)$.

Then the cost of a sequence of choices is random. The objective function is

$$E\left(\sum_{t=1}^{T} g(i(t-1), i(t))\right)$$

and we can define

$$J(i) = E \min_j \left[g(i, j) + J(j)\right]$$

Dynamic Programming

Continuous Time, Mixed State

Stochastic Example

Context: The planning/scheduling hierarchy

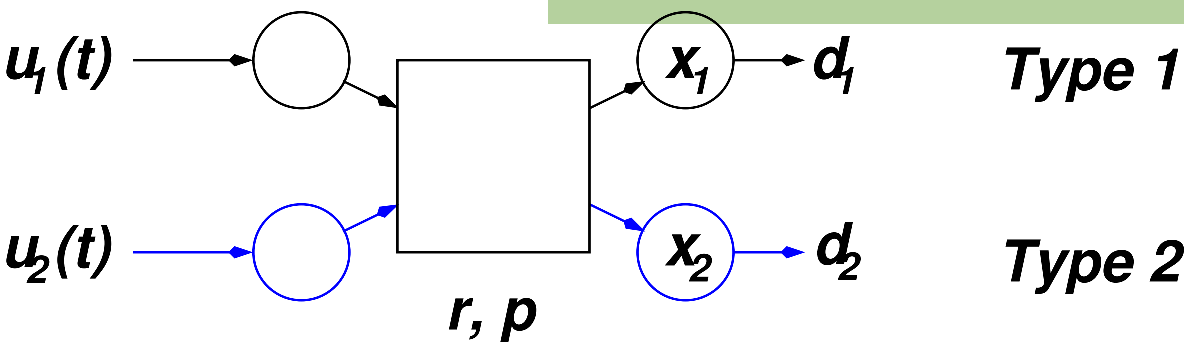
- Long term: factory design, capital expansion, etc.
- Medium term: demand planning, staffing, etc.
- Short term:
  - ⋆ response to short term events
  - ⋆ part release and dispatch

In this problem, we deal with the response to short term events. The factory and the demand are given to us; we must calculate short term production rates; these rates are the targets that release and dispatch must achieve.

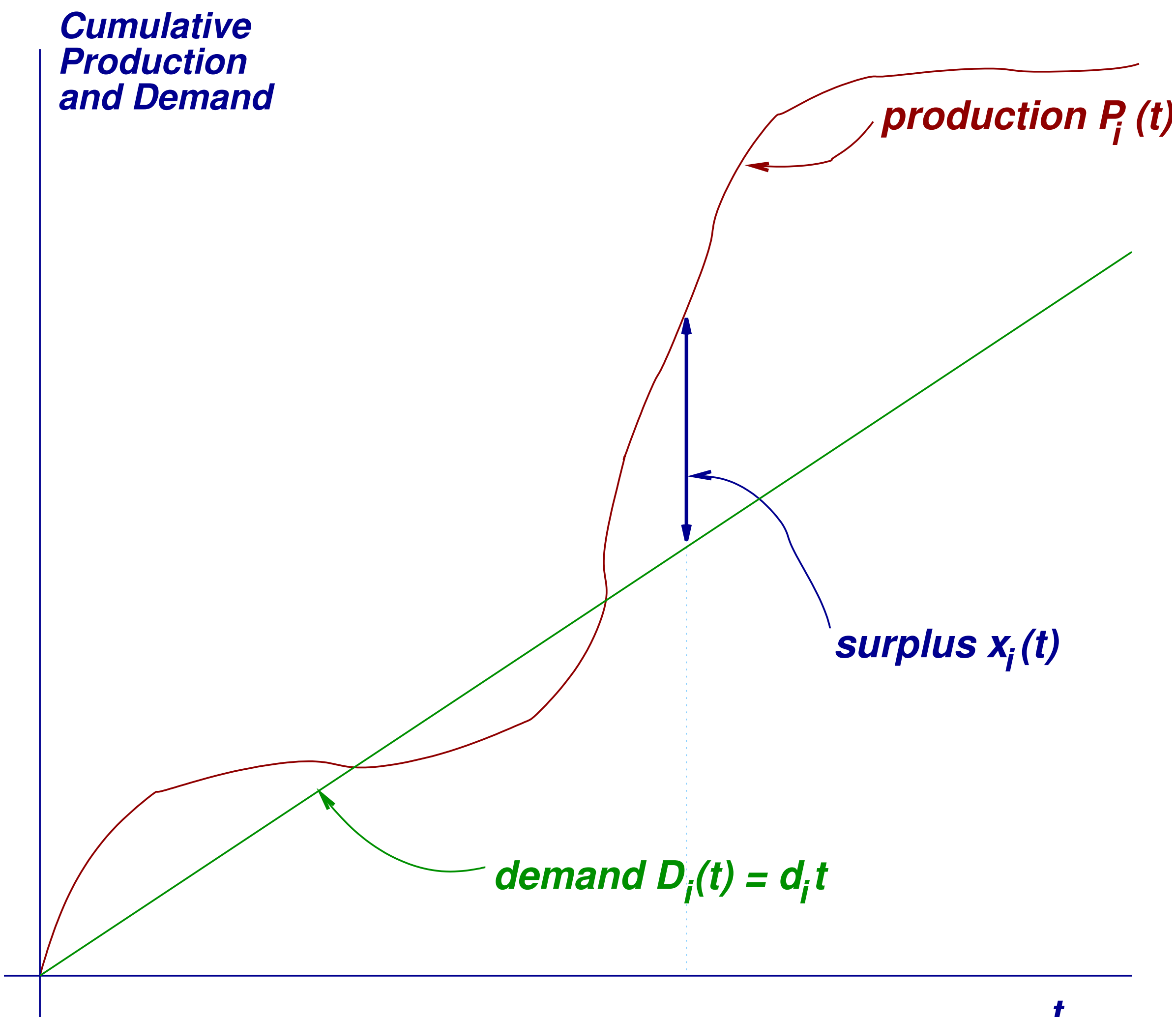
Dynamic Programming

Continuous Time, Mixed State

Stochastic Example



- Perfectly flexible machine, two part types. τ_i time units required to make Type i parts, $i = 1, 2$.
- Exponential failures and repairs with rates p and r.
- Constant demand rates d_1, d_2.
- Instantaneous production rates $u_i(t), i = 1, 2$ — *control variables* .
- Downstream surpluses $x_i(t)$.



Objective: Minimize the difference between cumulative production and cumulative demand.

The surplus satisfies

$$x_i(t) = P_i(t) - D_i(t)$$

Feasibility:

- For the problem to be feasible, it must be possible to make approximately $d_i T$ Type i parts in a long time period of length $T, i = 1, 2$. *(Why "approximately"?)*
- The time required to make $d_i T$ parts is $\tau_i d_i T$.
- During this period, the total up time of the machine — ie, the time available for production — is approximately $r/(r+p)T$.
- Therefore, we must have $\tau_1 d_1 T + \tau_2 d_2 T \leq r/(r+p)T$, or

$$\sum_{i=1}^{2} \tau_i d_i \leq \frac{r}{r+p}$$

If this condition is not satisfied, the demand cannot be met. *What will happen to the surplus?*

The feasibility condition is also written

$$\sum_{i=1}^{2} \frac{d_i}{\mu_i} \leq \frac{r}{r+p}$$

where $\mu_i = 1/\tau_i$.

If there were only one part type, this would be

$$d \leq \mu \frac{r}{r+p}$$

Look familiar?

Dynamic Programming

Continuous Time, Mixed State

Stochastic Example

The surplus satisfies

$$x_i(t) = P_i(t) - D_i(t)$$

where

$$P_i(t) = \int_0^t u_i(s)ds; \qquad D_i(t) = d_i t$$

Therefore

$$\frac{dx_i(t)}{dt} = u_i(t) - d_i$$

To define the objective more precisely, let there be a function $g(x_1, x_2)$ such that

- g is convex
- $g(0, 0) = 0$
- $\lim_{x_1 \to \infty} g(x_1, x_2) = \infty$; $\lim_{x_1 \to -\infty} g(x_1, x_2) = \infty$.
- $\lim_{x_2 \to \infty} g(x_1, x_2) = \infty$; $\lim_{x_2 \to -\infty} g(x_1, x_2) = \infty$.

Examples:

- $g(x_1, x_2) = A_1 x_1^2 + A_2 x_2^2$
- $g(x_1, x_2) = A_1 |x_1| + A_2 |x_2|$
- $g(x_1, x_2) = g_1(x_1) + g_2(x_2)$ where
  - ⋆ $g_i(x_i) = g_{(i+)} x_i^+ + g_{(i-)} x_i^-$,
  - ⋆ $x_i^+ = \max(x_i, 0)$, $x_i^- = -\min(x_i, 0)$,
  - ⋆ $g_{(i+)} > 0, g_{(i-)} > 0$.

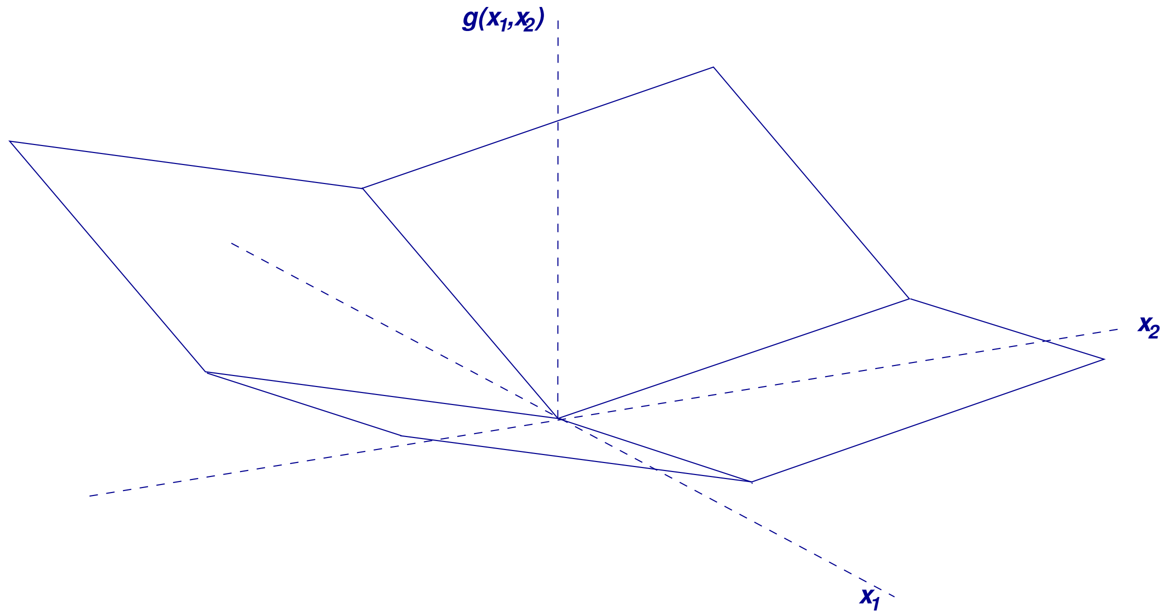
Dynamic Programming

Continuous Time, Mixed State

Stochastic Example

Objective:

$$\min E \int_0^T g(x_1(t), x_2(t))dt$$



Constraints:

$$u_1(t) \geq 0; \qquad u_2(t) \geq 0$$

Short-term capacity:

- If the machine is down at time t,

$$u_1(t) = u_2(t) = 0$$

Dynamic Programming

Continuous Time, Mixed State

Stochastic Example

- Assume the machine is up for a short period $[t, t + \delta t]$. Let δt be small enough so that u_i is constant; that is

$$u_i(s) = u_i(t), s \in [t, t + \delta t]$$

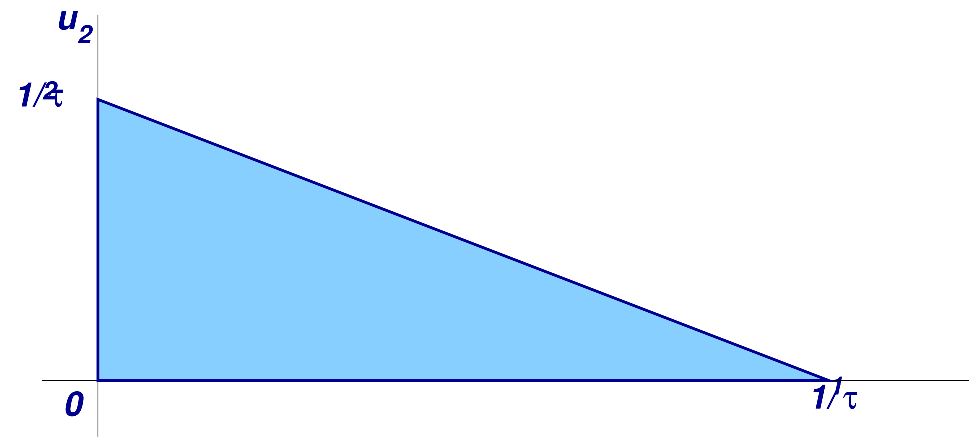
The machine makes $u_i(t)\delta t$ parts of type i. The time required to make that number of Type i parts is $\tau_i u_i(t)\delta t$.

Therefore

$$\sum_i \tau_i u_i(t)\delta t \leq \delta t$$

or

$$\sum_i \tau_i u_i(t) \leq 1$$



Dynamic Programming

Continuous Time, Mixed State

Stochastic Example

Machine state dynamics: Define $\alpha(t)$ to be the repair state of the machine at time t. $\alpha(t) = 1$ means the machine is up; $\alpha(t) = 0$ means the machine is down.

$$\text{prob}(\alpha(t + \delta t) = 0 | \alpha(t) = 1) = p\delta t + o(\delta t)$$

$$\text{prob}(\alpha(t + \delta t) = 1 | \alpha(t) = 0) = r\delta t + o(\delta t)$$

The constraints may be written

$$\sum_i \tau_i u_i(t) \leq \alpha(t); \qquad u_i(t) \geq 0$$

Dynamic programming problem formulation:

$$\min E \int_0^T g(x_1(t), x_2(t))dt$$

subject to:

$$\frac{dx_i(t)}{dt} = u_i(t) - d_i$$

$$\text{prob}(\alpha(t + \delta t) = 0 | \alpha(t) = 1) = p\delta t + o(\delta t)$$

$$\text{prob}(\alpha(t + \delta t) = 1 | \alpha(t) = 0) = r\delta t + o(\delta t)$$

$$\sum_i \tau_i u_i(t) \leq \alpha(t); \qquad u_i(t) \geq 0$$

$$x(0), \alpha(0) \text{ specified}$$

Elements of a DP Problem

- *state:* x all the information that is available to determine the future evolution of the system.
- *control:* u the actions taken by the decision-maker.
- *objective function:* J the quantity that must be minimized;
- *dynamics:* the evolution of the state as a function of the control variables and random events.
- *constraints:* the limitations on the set of allowable controls
- *initial conditions:* the values of the state variables at the start of the time interval over which the problem is described. There are also sometimes *terminal conditions* such as in the network example.

Elements of a DP Solution

- *control policy:* $u(x(t), t)$. A *stationary* or *time-invariant* policy is of the form $u(x(t))$.

- *value function:* (also called the *cost-to-go* function) the value $J(x, t)$ of the objective function when the optimal control policy is applied starting at time t, when the initial state is $x(t) = x$.

Bellman's Equation

Continuous x, t

Deterministic

Problem:

$$\min_{u(t), 0 \leq t \leq T} \int_0^T g(x(t), u(t))dt + F(x(T))$$

such that

$$\frac{dx(t)}{dt} = f(x(t), u(t), t)$$

$x(0)$ specified

$$h(x(t), u(t)) \leq 0$$

$x \in R^n, u \in R^m, f \in R^n, h \in R^k$, and g and F are scalars.

Data: $T, x(0)$, and the functions f, g, h, and F.

Bellman's Equation

Continuous x, t

Deterministic

The cost-to-go function is

$$J(x,t) = \min \int_t^T g(x(s),u(s))ds + F(x(T))$$

$$J(x(0),0) = \min \int_0^T g(x(s),u(s))ds + F(x(T))$$

$$= \min_{\substack{u(t),\\ 0\leq t\leq T}} \left\{ \int_0^{t_1} g(x(t),u(t))dt + \int_{t_1}^T g(x(t),u(t))dt + F(x(T)) \right\}.$$

Bellman's Equation

Continuous x, t

Deterministic

$$= \min_{\substack{u(t),\\ 0\le t\le t_1}} \left\{ \int_0^{t_1} g(x(t),u(t))dt + \min_{\substack{u(t),\\ t_1\le t\le T}} \left[\int_{t_1}^{T} g(x(t),u(t))dt + F(x(T)) \right] \right\}$$

$$= \min_{\substack{u(t),\\ 0\le t\le t_1}} \left\{ \int_0^{t_1} g(x(t),u(t))dt + J(x(t_1),t_1) \right\}.$$

Bellman's Equation

Continuous x, t

Deterministic

where

$$J(x(t_1), t_1) = \min_{u(t), t_1 \leq t \leq T} \int_{t_1}^{T} g(x(t), u(t))dt + F(x(T))$$

such that

$$\frac{dx(t)}{dt} = f(x(t), u(t), t)$$

$x(t_1)$ specified

$$h(x(t), u(t)) \leq 0$$

Bellman's Equation

Continuous x, t

Deterministic

Break up $[t_1, T]$ into $[t_1, t_1 + \delta t] \cup [t_1 + \delta t, T]$:

$$J(x(t_1), t_1) = \min_{u(t_1)} \left\{ \int_{t_1}^{t_1+\delta t} g(x(t), u(t))dt \right.$$

$$\left. + J(x(t_1 + \delta t), t_1 + \delta t) \right\}$$

where δt is small enough so that we can approximate $x(t)$ and $u(t)$ with constant $x(t_1)$ and $u(t_1)$, during the interval. Then, approximately,

$$J(x(t_1), t_1) = \min_{u(t_1)} \left\{ g(x(t_1), u(t_1))\delta t + J(x(t_1 + \delta t), t_1 + \delta t) \right\}$$

Bellman's Equation

Continuous x, t

Deterministic

Or,

$$J(x(t_1), t_1) = \min_{u(t_1)} \left\{ g(x(t_1), u(t_1))\delta t + J(x(t_1), t_1) + \right.$$

$$\left. \frac{\partial J}{\partial x}(x(t_1), t_1)(x(t_1 + \delta t) - x(t_1)) + \frac{\partial J}{\partial t}(x(t_1), t_1)\delta t \right\}$$

Note that

$$x(t_1 + \delta t) = x(t_1) + \frac{dx}{dt}\delta t = x(t_1) + f(x(t_1), u(t_1), t_1)\delta t$$

Bellman's Equation

Continuous x, t

Deterministic

Therefore

$$J(x, t_1) = J(x, t_1)$$

$$+ \min_u \left\{ g(x,u)\delta t + \frac{\partial J}{\partial x}(x, t_1) f(x, u, t_1)\delta t + \frac{\partial J}{\partial t}(x, t_1)\delta t \right\}$$

where $x = x(t_1); u = u(t_1) = u(x(t_1), t_1)$.

Then (dropping the t subscript)

$$-\frac{\partial J}{\partial t}(x, t) = \min_u \left\{ g(x, u) + \frac{\partial J}{\partial x}(x, t) f(x, u, t) \right\}$$

Bellman's Equation

Continuous x, t

Deterministic

This is the *Bellman equation* . It is the counterpart of the recursion equation for the network example.

- If we had a guess of $J(x,t)$ (for all x and t) we could confirm it by performing the minimization.
- If we knew $J(x,t)$ for all x and t, we could determine u by performing the minimization. U could then be written

$$u = U\left(x, \frac{\partial J}{\partial x}, t\right).$$

  This would be a *feedback law* .

The Bellman equation is usually impossible to solve analytically or numerically.

There are some important special cases that can be solved analytically.

Bellman's Equation

Continuous x, t

Example

Bang-Bang Control

$$\min \int_0^\infty |x| dt$$

subject to

$$\frac{dx}{dt} = u$$

$x(0)$ specified

$$-1 \leq u \leq 1$$

Bellman's Equation

Continuous x, t

Example

The Bellman equation is

$$-\frac{\partial J}{\partial t}(x,t) = \min_{\substack{u, \\ -1 \leq u \leq 1}} \left\{ |x| + \frac{\partial J}{\partial x}(x,t)u \right\}.$$

$J(x,t) = J(x)$ is a solution because the time horizon is infinite and t does not appear explicitly in the problem data (ie, $g(x) = |x|$ is not a function of t.

Therefore

$$0 = \min_{\substack{u, \\ -1 \leq u \leq 1}} \left\{ |x| + \frac{dJ}{dx}(x)u \right\}.$$

$J(0) = 0$ because if $x(0) = 0$ we can choose $u(t) = 0$ for all t. Then $x(t) = 0$ for all t and the integral is 0. There is no possible choice of $u(t)$ that will make the integral less than 0, so this is the minimum.

The minimum is achieved when

$$\mathbf{u} = \begin{cases} -1 & \text{if } \dfrac{dJ}{dx}(x) > 0 \\[2ex] 1 & \text{if } \dfrac{dJ}{dx}(x) < 0 \\[2ex] \text{undetermined} & \text{if } \dfrac{dJ}{dx}(x) = 0 \end{cases}$$

Why?

Bellman's Equation

Continuous x, t

Example

Consider the set of x where $dJ/dx(x) < 0$. For x in that set, $u = 1$, so

$$0 = |x| + \frac{dJ}{dx}(x)$$

or

$$\frac{dJ}{dx}(x) = -|x|$$

Similarly, if x is such that $dJ/dx(x) > 0$ and $u = -1$,

$$\frac{dJ}{dx}(x) = |x|$$

Bellman's Equation

Continuous x, t

Example

To complete the solution, we must determine where $dJ/dx > 0$, < 0, and $= 0$.

We already know that $J(0) = 0$. We must have $J(x) > 0$ for all $x \neq 0$ because $|x| > 0$ so the integral of $|x(t)|$ must be positive.

Since $J(x) > J(0)$ for all $x \neq 0$, we must have

$$\frac{dJ}{dx}(x) < 0 \text{ for } x < 0$$

$$\frac{dJ}{dx}(x) > 0 \text{ for } x > 0$$

Bellman's Equation

Continuous x, t

Example

Therefore

$$\frac{dJ}{dx}(x) >= x$$

so

$$J = \frac{1}{2}x^2$$

and

$$u = \begin{cases} 1 & \text{if} \quad x < 0 \\ 0 & \text{if} \quad x = 0 \\ -1 & \text{if} \quad x > 0 \end{cases}$$

Bellman's Equation

Continuous x, t, Discrete α

Stochastic

$$J(x(0), \alpha(0), 0) = \min_u E \left\{ \int_0^T g(x(t), u(t)) dt + F(x(T)) \right\}$$

such that

$$\frac{dx(t)}{dt} = f(x, \alpha, u, t)$$

prob $[\alpha(t + \delta t) = i \mid \alpha(t) = j] = \lambda_{ij} \delta t$ for all $i, j, i \neq j$

$x(0), \alpha(0)$ specified

$$h(x(t), \alpha(t), u(t)) \leq 0$$

Bellman's Equation

Continuous x, t, Discrete α

Stochastic

Getting the Bellman equation in this case is more complicated because α changes by large amounts when it changes.

Let $H(\alpha)$ be some function of α. We need to calculate

$$\tilde{E}H(\alpha(t+\delta t)) = E\left\{H(\alpha(t+\delta t)) \mid \alpha(t)\right\}$$

$$= \sum_{j} H(j)\text{prob }\left\{\alpha(t+\delta t) = j \mid \alpha(t)\right\}$$

Bellman's Equation

Continuous x, t, Discrete α

Stochastic

$$= \sum_{j \neq \alpha(t)} H(j)\lambda_{j\alpha(t)}\delta t + H(\alpha(t))\left(1 - \sum_{j \neq \alpha(t)} \lambda_{j\alpha(t)}\delta t\right) + o(\delta t)$$

$$= \sum_{j \neq \alpha(t)} H(j)\lambda_{j\alpha(t)}\delta t + H(\alpha(t))\left(1 + \lambda_{\alpha(t)\alpha(t)}\delta t\right) + o(\delta t)$$

$$E\left\{H(\alpha(t+\delta t)) \mid \alpha(t)\right\} = H(\alpha(t)) + \left[\sum_{j} H(j)\lambda_{j\alpha(t)}\right]\delta t + o(\delta t)$$

We use this in the derivation of the Bellman equation.

Bellman's Equation

Continuous x, t, Discrete α

Stochastic

$$J(x(t), \alpha(t), t) = \min_{\substack{u(s), \\ t \leq s < T}} E\left\{ \int_t^T g(x(s), u(s))ds + F(x(T)) \right\}$$

Bellman's Equation

Continuous x, t, Discrete α

Stochastic

$$= \min_{\substack{u(s),\\ 0 \le s \le t+\delta t}} E \left\{ \int_t^{t+\delta t} g(x(s), u(s))\, ds \right.$$

$$\left. + \min_{\substack{u(s),\\ t+\delta t \le s \le T}} E \left[\int_{t+\delta t}^{T} g(x(s), u(s))\, ds + F(x(T)) \right] \right\}$$

Bellman's Equation

Continuous x, t, Discrete α

Stochastic

$$= \min_{\substack{u(s),\\ t\leq s\leq t+\delta t}} \tilde{E}\left\{\int_t^{t+\delta t} g(x(s),u(s))ds\right.$$

$$\left. + J(x(t+\delta t),\alpha(t+\delta t),t+\delta t)\right\}$$

Next, we expand the second term in a Taylor series about $x(t)$.

We leave $\alpha(t+\delta t)$ alone, for now.

Bellman's Equation

Continuous x, t, Discrete α

Stochastic

$$J(x(t), \alpha(t), t) =$$

$$\min_{u(t)} \tilde{E} \Big\{ g(x(t), u(t))\delta t + J(x(t), \alpha(t+\delta t), t) +$$

$$\frac{\partial J}{\partial x}(x(t), \alpha(t+\delta t), t)\delta x(t) + \frac{\partial J}{\partial t}(x(t), \alpha(t+\delta t), t)\delta t \Big\} + o(\delta t).$$

where

$$\delta x(t) = x(t+\delta t) - x(t) = f(x(t), \alpha(t), u(t), t)\delta t + o(\delta t)$$

Bellman's Equation

Continuous x, t,Discrete α

Stochastic

Using the expansion of $\tilde{E}H(\alpha(t+\delta t))$,

$$J(x(t),\alpha(t),t) = \min_{u(t)} \Bigg\{ g(x(t),u(t))\delta t + J(x(t),\alpha(t),t) + \sum_j J(x(t),j,t)\lambda_{j\alpha(t)}\delta t + \frac{\partial J}{\partial x}(x(t),\alpha(t),t)\delta x(t) + \frac{\partial J}{\partial t}(x(t),\alpha(t),t)\delta t \Bigg\} + o(\delta t)$$

We can clean up notation by replacing $x(t)$ with x, $\alpha(t)$ with α, and $u(t)$ with u.

Bellman's Equation

Continuous x, t, Discrete α

Stochastic

$$J(x, \alpha, t) =$$

$$\min_u \left\{ g(x, u)\delta t + J(x, \alpha, t) + \sum_j J(x, j, t)\lambda_{j\alpha}\delta t \right.$$

$$\left. + \frac{\partial J}{\partial x}(x, \alpha, t)\delta x + \frac{\partial J}{\partial t}(x, \alpha, t)\delta t \right\} + o(\delta t)$$

We can subtract $J(x, \alpha, t)$ from both sides and use the expression for δx to get ...

Bellman's Equation

Continuous x, t,Discrete α

Stochastic

$$0 = \min_{u} \left\{ g(x, u)\delta t + \sum_{j} J(x, j, t)\lambda_{j\alpha}\delta t \right.$$

$$\left. + \frac{\partial J}{\partial x}(x, \alpha, t) f(x, \alpha, u, t)\delta t + \frac{\partial J}{\partial t}(x, \alpha, t)\delta t \right\} + o(\delta t)$$

or,

Bellman's Equation

Continuous x, t, Discrete α

Stochastic

$$-\frac{\partial J}{\partial t}(x, \alpha, t) = \sum_j J(x, j, t)\lambda_{j\alpha} +$$

$$\min_u \left\{ g(x, u) + \frac{\partial J}{\partial x}(x, \alpha, t) f(x, \alpha, u, t) \right\}$$

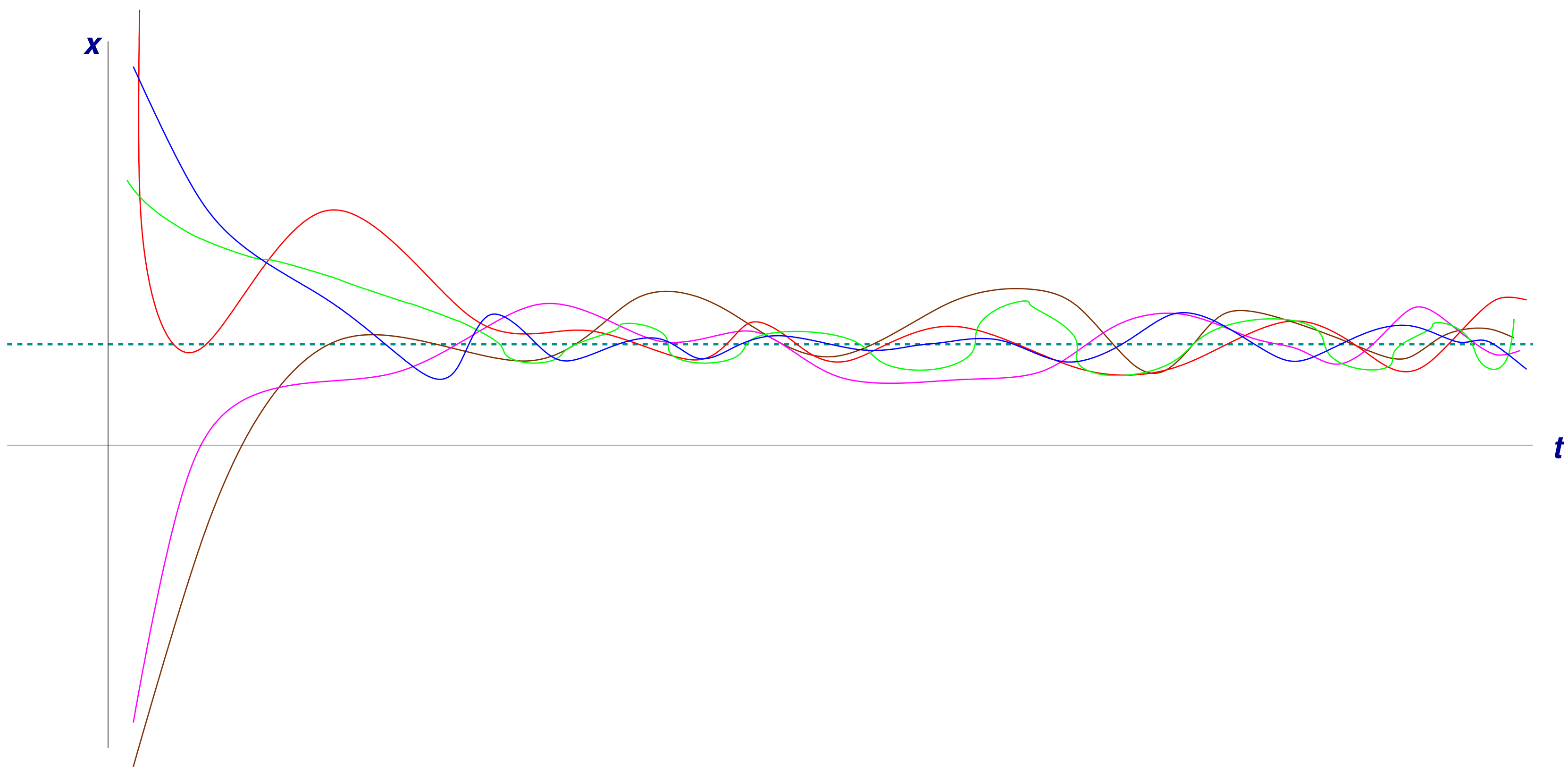
- *Bad news:* usually impossible to solve;
- *Good news:* insight.

Bellman's Equation

Continuous x, t, Discrete α

Stochastic

An approximation: when T is large and f is not a function of t, typical trajectories look like this:



Bellman's Equation

Continuous x, t, Discrete α

Stochastic

That is, in the long run, x approaches a steady-state probability distribution. Let J^* be the expected value of $g(x, u)$, where u is the optimal control.

Suppose we started the problem with $x(0)$ a random variable whose probability distribution is the steady-state distribution. Then, for large T,

$$EJ = \min_u E\left\{\int_0^T g(x(t), u(t))dt + F(x(T))\right\}$$

$$\approx J^*T$$

Bellman's Equation

Continuous x, t, Discrete α

Stochastic

For $x(0)$ and $\alpha(0)$ specified

$$J(x(0), \alpha(0), 0) \approx J^*T + W(x(0), \alpha(0))$$

or, more generally, for $x(t) = x$ and $\alpha(t) = \alpha$ specified,

$$J(x, \alpha, t) \approx J^*(T - t) + W(x, \alpha)$$

Flexible Manufacturing System Control

Single machine, multiple part types. x, u, d are N-dimensional vectors.

$$\min E \int_0^T g(x(t))dt$$

subject to:

$$\frac{dx_i(t)}{dt} = u_i(t) - d_i, \quad i = 1, ..., N$$

$$\text{prob}(\alpha(t + \delta t) = 0 | \alpha(t) = 1) = p\delta t + o(\delta t)$$

$$\text{prob}(\alpha(t + \delta t) = 1 | \alpha(t) = 0) = r\delta t + o(\delta t)$$

$$\sum_i \tau_i u_i(t) \leq \alpha(t); \qquad u_i(t) \geq 0$$

$$x(0), \alpha(0) \text{ specified}$$

Flexible Manufacturing System Control

Define $\Omega(\alpha) = \{u| \sum_i \tau_i u_i \leq \alpha\}$. Then, for $\alpha = 0, 1$,

$$-\frac{\partial J}{\partial t}(x, \alpha, t) = \sum_j J(x, j, t)\lambda_{j\alpha} +$$

$$\min_{u \in \Omega(\alpha)} \left\{ g(x) + \frac{\partial J}{\partial x}(x, \alpha, t)(u - d) \right\}$$

Flexible Manufacturing System Control

Approximating J with $J^*(T-t)+W(x,\alpha)$ gives:

$$J^* = \sum_j (J^*(T-t) + W(x,j))\lambda_{j\alpha} +$$

$$\min_{u \in \Omega(\alpha)} \left\{ g(x) + \frac{\partial W}{\partial x}(x,\alpha,t)(u-d) \right\}$$

Recall that

$$\sum_j \lambda_{j\alpha} = 0...$$

Flexible Manufacturing System Control

so

$$J^* = \sum_j W(x,j)\lambda_{j\alpha} +$$

$$\min_{u \in \Omega(\alpha)} \left\{ g(x) + \frac{\partial W}{\partial x}(x,\alpha,t)(u-d) \right\}$$

for $\alpha = 0, 1$

Flexible Manufacturing System Control

This is actually two equations, one for $\alpha = 0$, one for $\alpha = 1$.

$$J^* = g(x) + W(x,1)r - W(x,0)r - \frac{\partial W}{\partial x}(x,0)d,$$

$$\text{for } \alpha = 0,$$

$$J^* = g(x) + W(x,0)p - W(x,1)p + \min_{u \in \Omega(1)} \left[\frac{\partial W}{\partial x}(x,1)(u-d) \right]$$

$$\text{for } \alpha = 1.$$

Flexible Manufacturing System Control

Single-part-type case

Technically, not flexible!

Now, x and u are scalars, and

$$\Omega(1) = [0, 1/\tau] = [0, \mu]$$

$$J^* = g(x) + W(x,1)r - W(x,0)r - \frac{dW}{dx}(x,0)d,$$
for $\alpha = 0$,

$$J^* = g(x) + W(x,0)p - W(x,1)p + \min_{0 \le u \le \mu} \left[\frac{dW}{dx}(x,1)(u-d) \right]$$
for $\alpha = 1$.

Flexible Manufacturing System Control

Single-part-type case

See book, Sections 2.6.2 and 9.3; see Probability slides # 91–120.

When $\alpha = 0, u = 0$.

When $\alpha = 1$,

- if $\frac{dW}{dx} < 0, \quad u = \mu$,
- if $\frac{dW}{dx} = 0, \quad u$ unspecified,
- if $\frac{dW}{dx} > 0, \quad u = 0$.

$W(x,\alpha)$ has been shown to be convex in x. If the minimum of $W(x,1)$ occurs at $x = Z$ and $W(x,1)$ is differentiable for all x, then

- $\frac{dW}{dx} < 0 \leftrightarrow x < Z$
- $\frac{dW}{dx} = 0 \leftrightarrow x = Z$
- $\frac{dW}{dx} > 0 \leftrightarrow x > Z$

Therefore,

- if $x < Z$, $u = \mu$,
- if $x = Z$, u unspecified,
- if $x > Z$, $u = 0$.

Flexible Manufacturing System Control

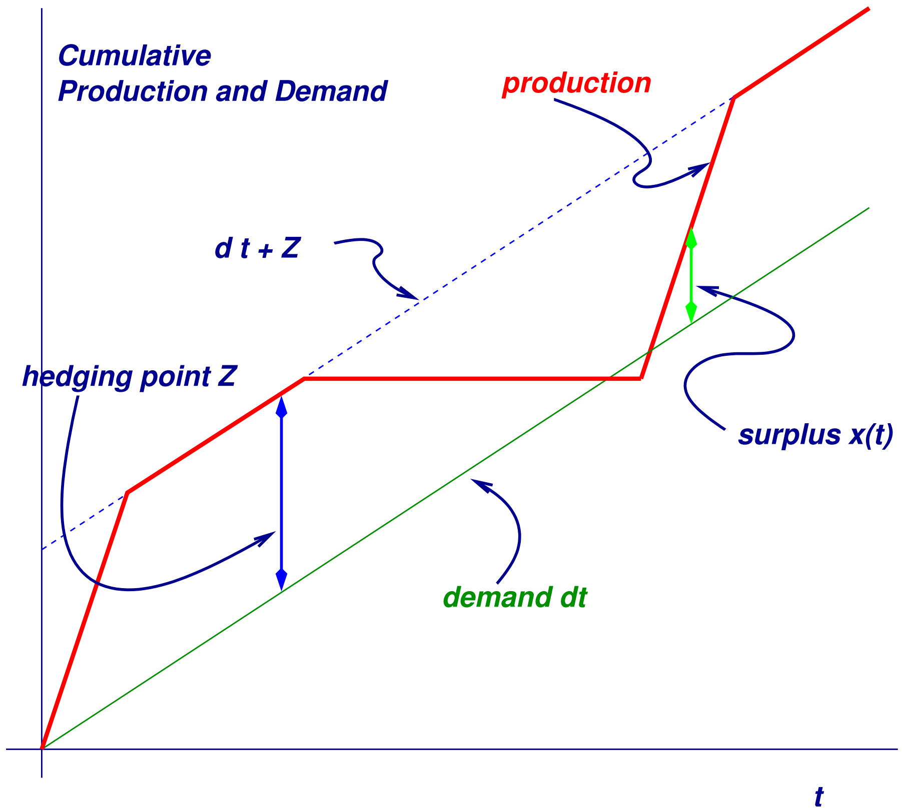
Single-part-type case

Surplus, or inventory/backlog:

$$\frac{dx(t)}{dt} = u(t) - d$$

Production policy: Choose Z (the *hedging point*) Then,

- if $\alpha = 1$,
  - ⋆ if $x < Z$, $u = \mu$,
  - ⋆ if $x = Z$, $u = d$,
  - ⋆ if $x > Z$, $u = 0$;
- if $\alpha = 0$,
  - ⋆ $u = 0$.



How do we choose Z?

Determination of Z

$$J^* = Eg(x) = g(Z)P(Z,1) + \int_{-\infty}^{Z} g(x)\left[f(x,0) + f(x,1)\right] dx$$

in which P and f form the steady-state probability distribution of x. *We choose Z to minimize J^*.* P and f are given by

$$f(x,0) = Ae^{bx}$$

$$f(x,1) = A\frac{d}{\mu - d}e^{bx}$$

$$P(Z,1) = A\frac{d}{p}e^{bZ}$$

where

$$b = \frac{r}{d} - \frac{p}{\mu - d}$$

and A is chosen so that

$$\int_{-\infty}^{Z} [f(x,0) + f(x,1)]\, dx + P(Z,1) = 1$$

After some manipulation,

$$A = \left[\frac{bp(\mu - d)}{db(\mu - d) + \mu p}\right] e^{-bZ}$$

and

$$P(Z,1) = \frac{db(\mu - d)}{db(\mu - d) + \mu p}$$

Since $g(x) = g_+ x^+ + g_- x^-$,

- if $Z \leq 0$, then

$$J^* = -g_- Z P(Z,1) - \int_{-\infty}^{Z} g_- x \left[f(x,0) + f(x,1) \right] dx;$$

- if $Z > 0$,

$$J^* = g_+ Z P(Z,1) - \int_{-\infty}^{0} g_- x \left[f(x,0) + f(x,1) \right] dx$$

$$+ \int_{0}^{Z} g_+ x \left[f(x,0) + f(x,1) \right] dx.$$

To minimize J^*:

- if $g_+ - Kb(g_+ + g_-) < 0, Z = \dfrac{\ln\left(Kb(1 + \frac{g_-}{g_+})\right)}{b}$.

- if $g_+ - Kb(g_+ + g_-) \geq 0, Z = 0$

where $K =$

$$\frac{\mu p}{b(\mu b d - d^2 b + \mu p)} = \frac{\mu p}{b(r + p)(\mu - d)} = \frac{1}{b}\left[\frac{\mu p}{db(\mu - d) + \mu p}\right]$$

Z is a function of d, μ, r, p, g_+, g_-.

That is, we choose Z such that

$$e^{bZ} = \min\left\{1, Kb\left(\frac{g_+ + g_-}{g_+}\right)\right\}$$

or

$$e^{-bZ} = \max\left\{1, \frac{1}{Kb}\left(\frac{g_+}{g_+ + g_-}\right)\right\}$$

Single-part-type case

Determination of Z

$$\text{prob}(x \leq 0) = \int_{-\infty}^{0} (f(x,0) + f(x,1))dx$$

$$= A\left(1 + \frac{d}{\mu - d}\right)\int_{-\infty}^{0} e^{bx}dx$$

$$= A\left(1 + \frac{d}{\mu - d}\right)\frac{1}{b} = A\frac{\mu}{b(\mu - d)}$$

$$= \left[\frac{bp(\mu - d)}{db(\mu - d) + \mu p}\right] e^{-bZ}\frac{\mu}{b(\mu - d)}$$

$$= \left[\frac{\mu p}{db(\mu - d) + \mu p}\right] e^{-bZ}$$

Or,

$$\text{prob}(x \le 0) = \left[\frac{\mu p}{db(\mu - d) + \mu p}\right] \max\left\{1, \frac{1}{Kb}\left(\frac{g_+}{g_+ + g_-}\right)\right\}$$

It can be shown that

$$Kb = \frac{\mu p}{\mu p + bd(\mu - d)}$$

Therefore

$$\text{prob}(x \le 0) = Kb \max\left\{1, \frac{1}{Kb}\left(\frac{g_+}{g_+ + g_-}\right)\right\}$$

$$= \max\left\{\frac{\mu p}{\mu p + bd(\mu - d)}, \quad \frac{g_+}{g_+ + g_-}\right\}$$

That is,

- if $\dfrac{\mu p}{\mu p + bd(\mu - d)} > \dfrac{g_+}{g_+ + g_-}$, then $Z = 0$ and

$$\text{prob}(x \leq 0) = \frac{\mu p}{\mu p + bd(\mu - d)};$$

- if $\dfrac{\mu p}{\mu p + bd(\mu - d)} < \dfrac{g_+}{g_+ + g_-}$, then $Z > 0$ and

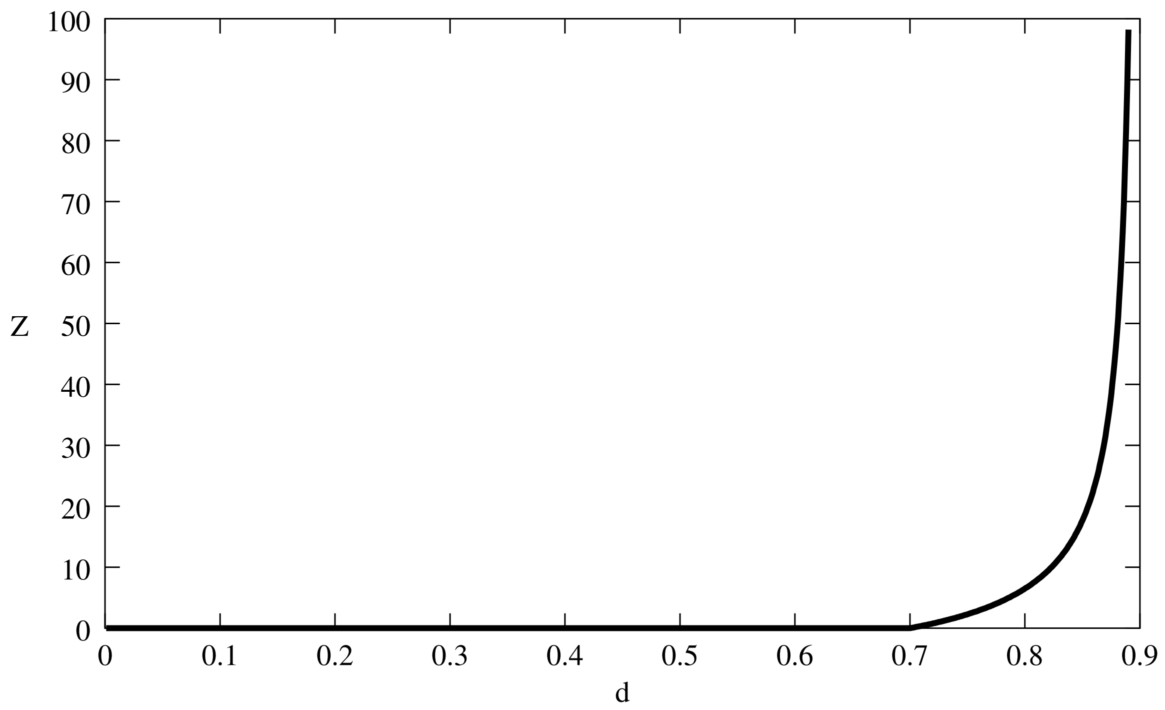
$$\text{prob}(x \leq 0) = \frac{g_+}{g_+ + g_-}.$$

This looks a lot like the solution of the “newsboy problem.”

Single-part-type case

Z vs. d

Base values: $g_+ = 1$, $g_- = 10$ $d = .7$, $\mu = 1.$, $r = .09$, $p = .01$.

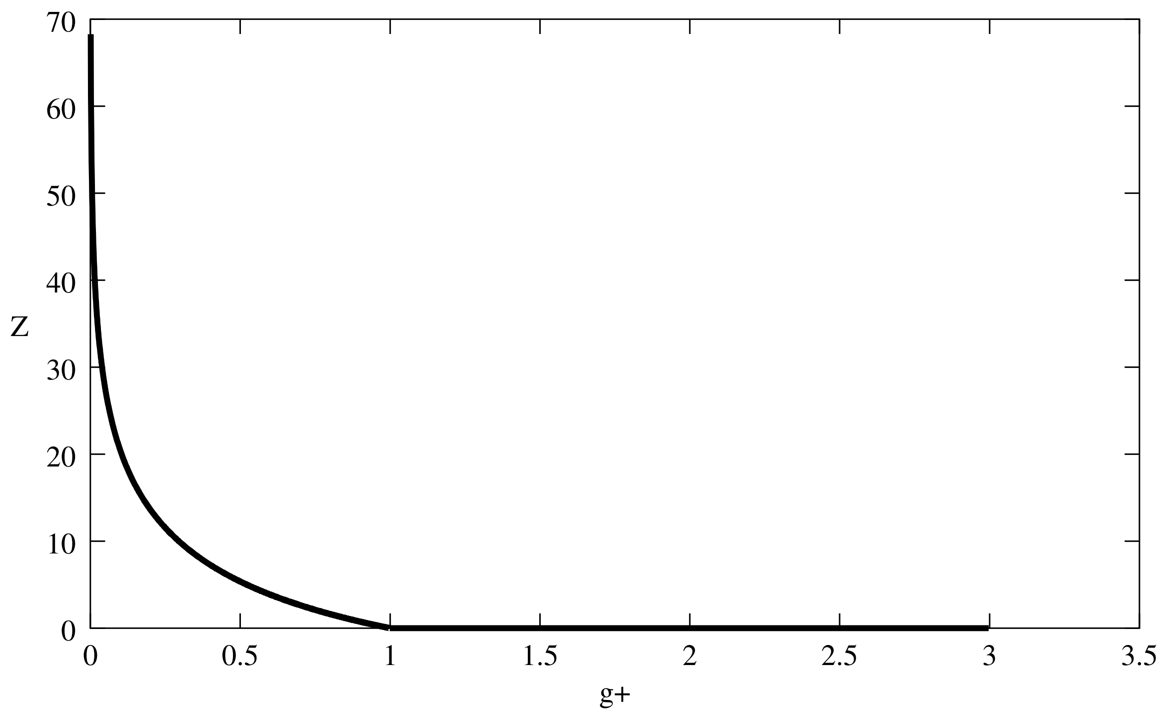


Flexible Manufacturing System Control

Single-part-type case

Z vs. g_+

Base values: $g_+ = 1$, $g_- = 10$ $d = .7$, $\mu = 1.$, $r = .09$, $p = .01$.

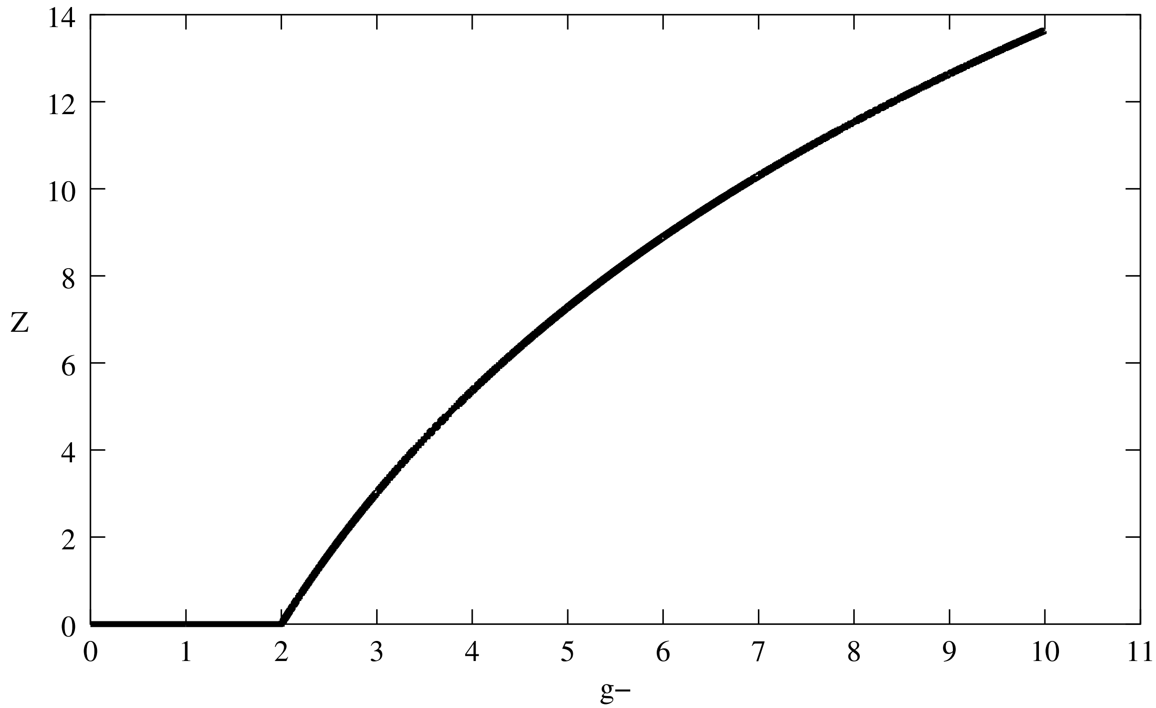


Flexible Manufacturing System Control

Single-part-type case

Z vs. g_-

Base values: $g_+ = 1$, $g_- = 10$ $d = .7$, $\mu = 1.$, $r = .09$, $p = .01$.

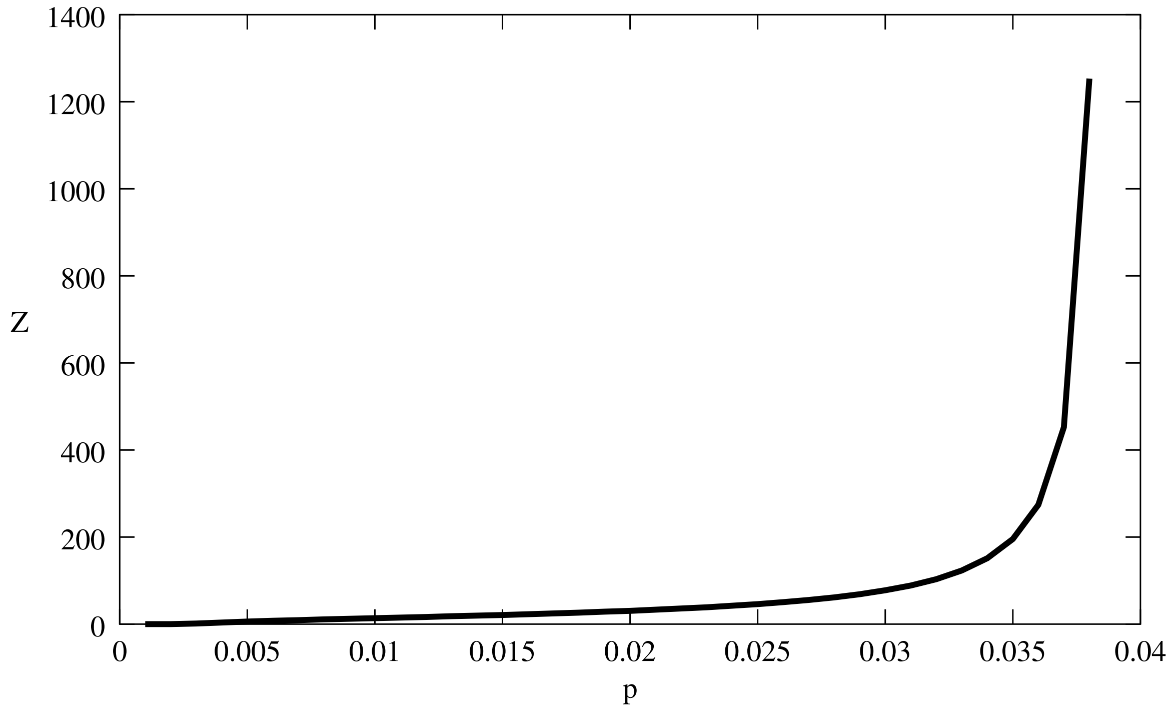


Flexible Manufacturing System Control

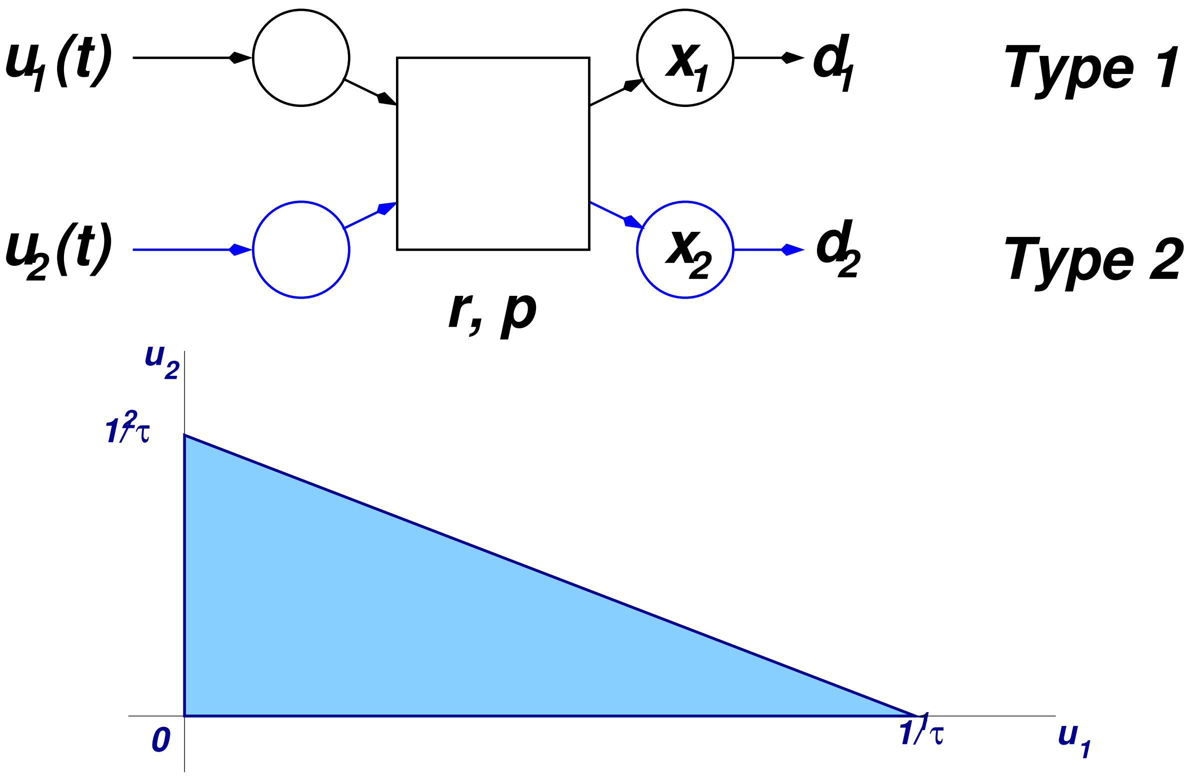
Single-part-type case

Z vs. p

Base values: $g_+ = 1$, $g_- = 10$ $d = .7$, $\mu = 1.$, $r = .09$, $p = .01$.



Two-part-type case



Capacity set $\Omega(1)$ when machine is up.

Flexible Manufacturing System Control

Two-part-type case

We must find $u(x, \alpha)$ to satisfy

$$\min_{u \in \Omega(\alpha)} \left\{ \frac{\partial W}{\partial x}(x, \alpha, t) \right\} u$$

Partial solution of LP:

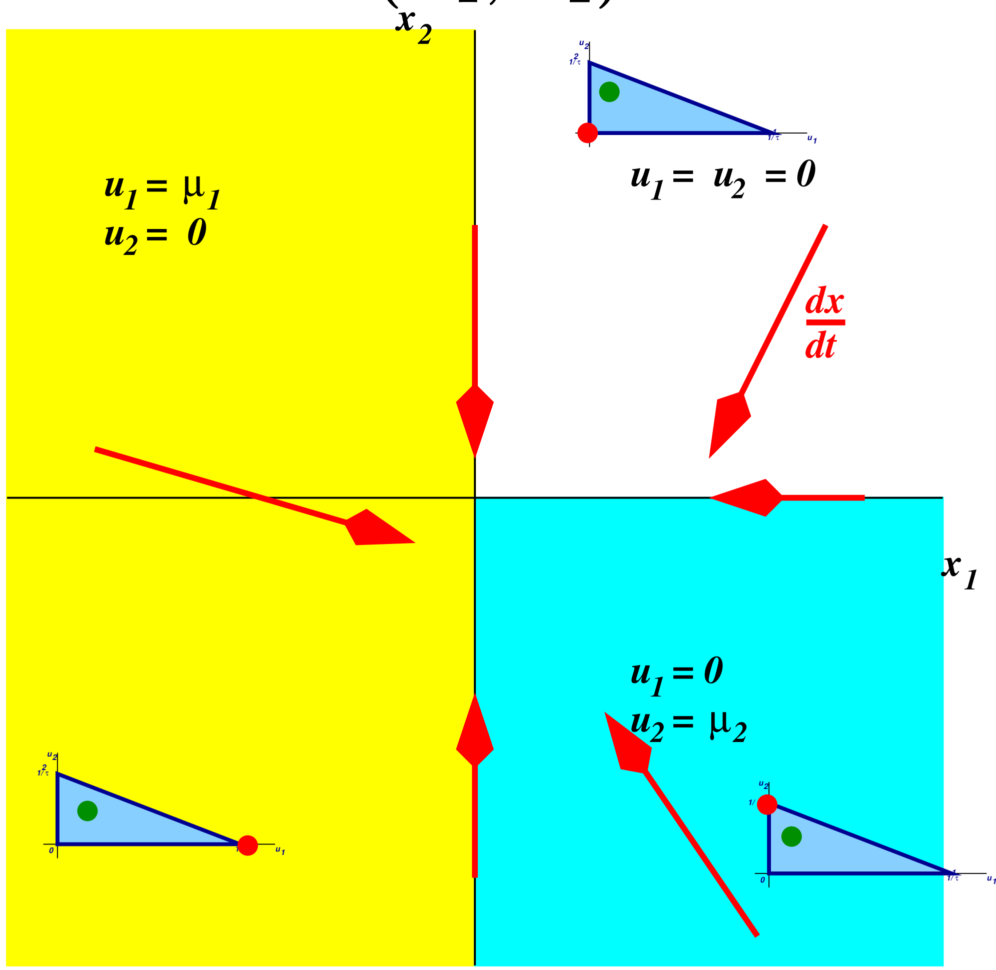
- If $\partial W / \partial x_1 > 0$ and $\partial W / \partial x_2 > 0$, $u_1 = u_2 = 0$.
- If $\partial W / \partial x_1 < \partial W / \partial x_2 < 0$, $u_1 = \mu_1, u_2 = 0$.
- If $\partial W / \partial x_2 < \partial W / \partial x_1 < 0$, $u_2 = \mu_2, u_1 = 0$.

Problem: no complete analytical solution available.

Flexible Manufacturing System Control

Two-part-type case

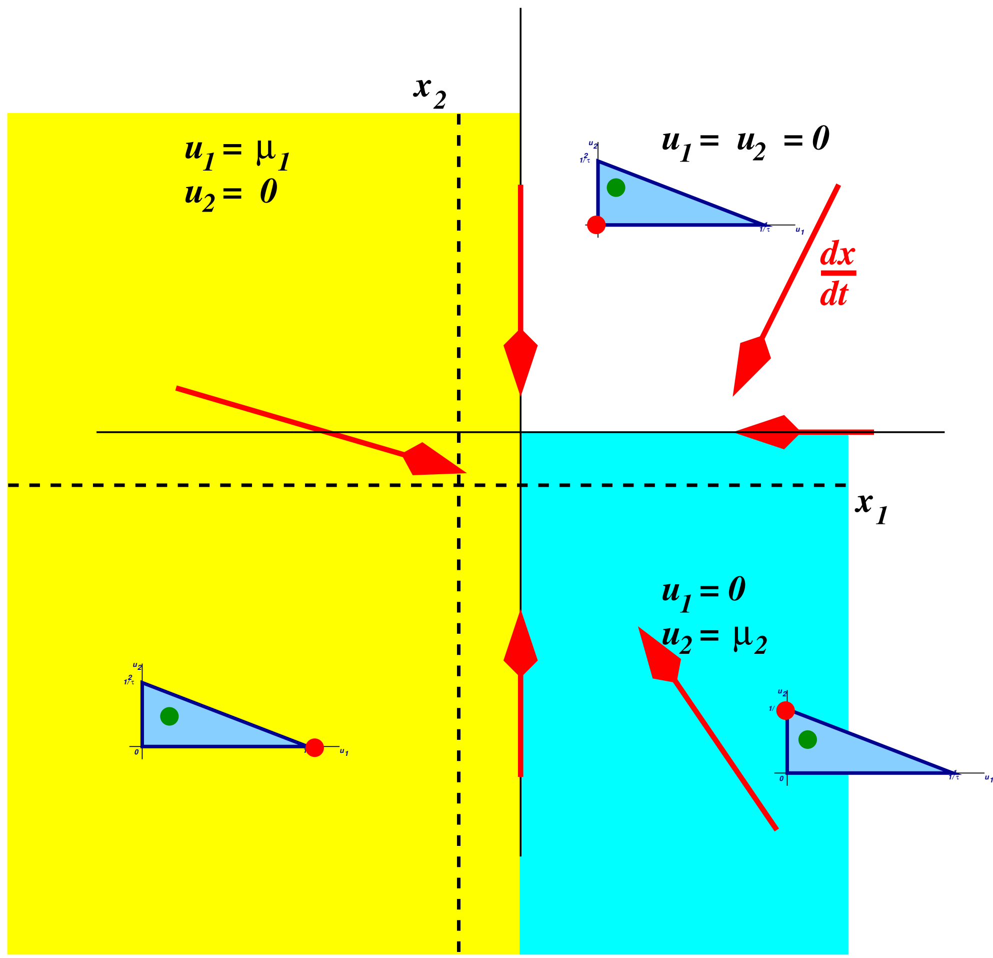
Case: Exact solution if $Z = (Z_1, Z_2) = 0$



Flexible Manufacturing System Control

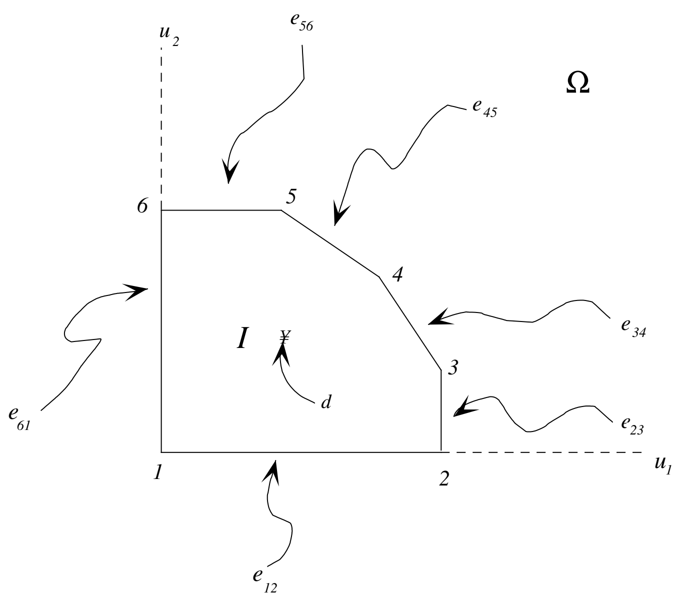
Two-part-type case

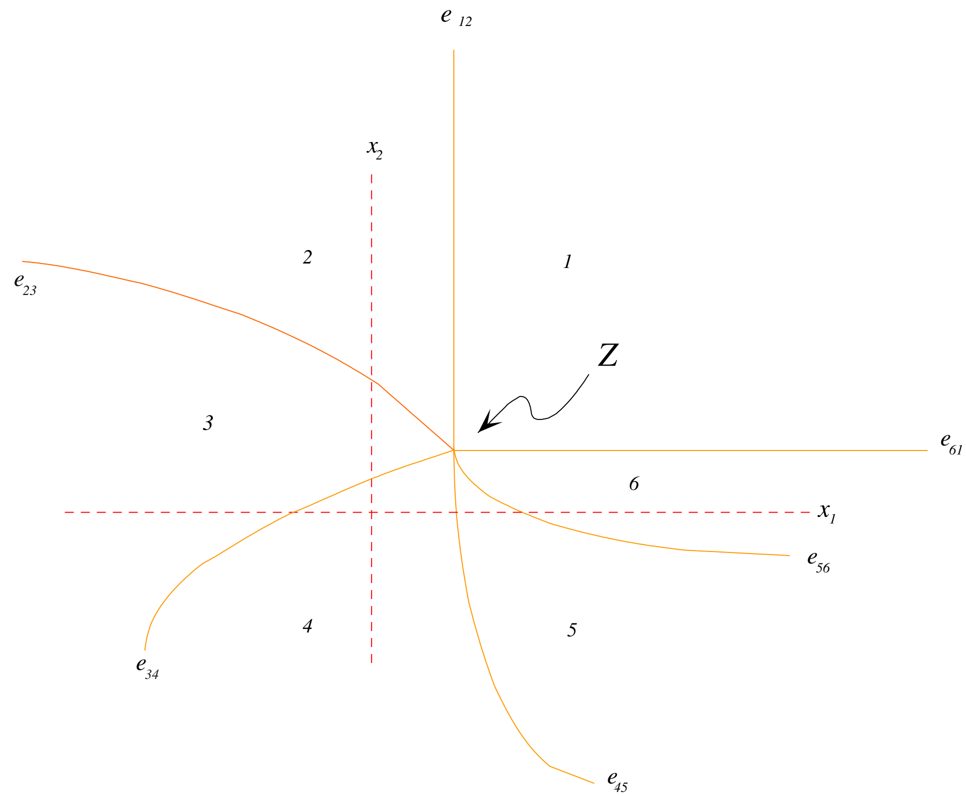
Case: Approximate solution if $Z > 0$



Two-part-type case

Two parts, multiple machines without buffers:



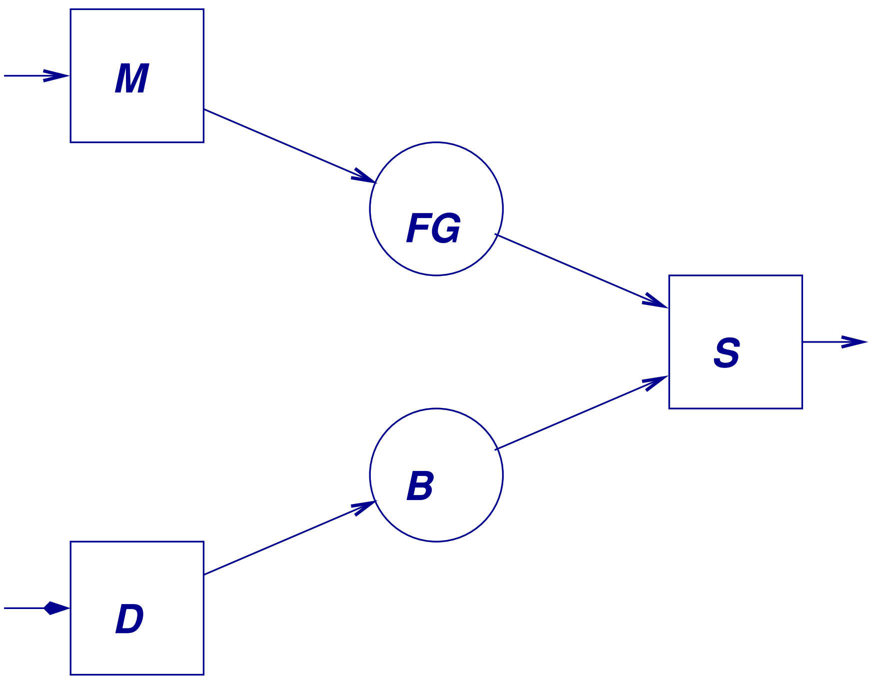


Two-part-type case

- Proposed approximate solution for multiple-part, single machine system:
  - ⋆ *Rank order the part types, and bring them to their hedging points in that order.*

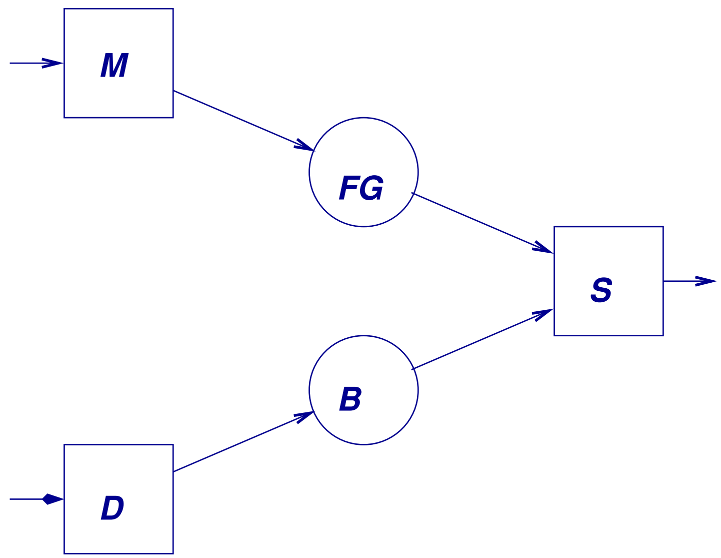
Single-part-type case

Surplus and tokens



- *Operating Machine M according to the hedging point policy is equivalent to operating this assembly system according to a finite buffer policy.*

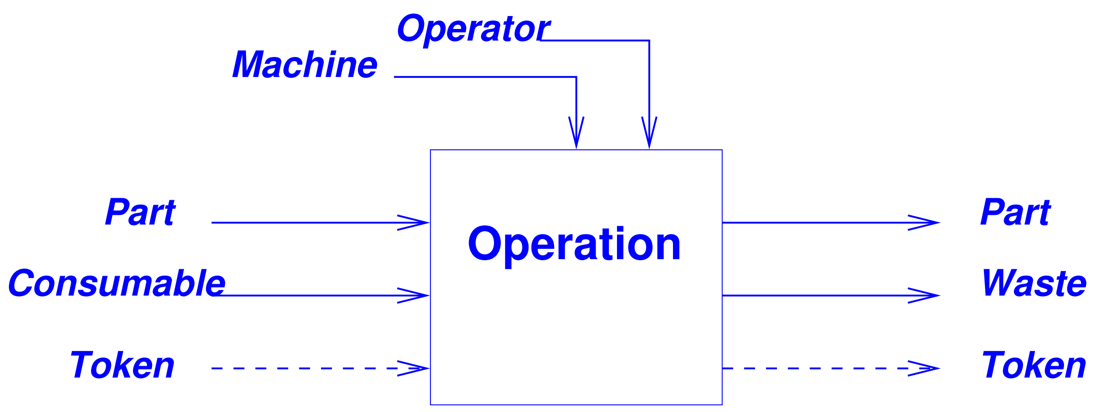
- D is a *demand generator* .
  - ⋆ Whenever a demand arrives, D sends a token to B.
- S is a synchronization machine.
  - ⋆ S is perfectly reliable and infinitely fast.
- FG is a finite finished goods buffer.
- B is an infinite backlog buffer.



Flexible Manufacturing System Control

Single-part-type case

Material/token policies

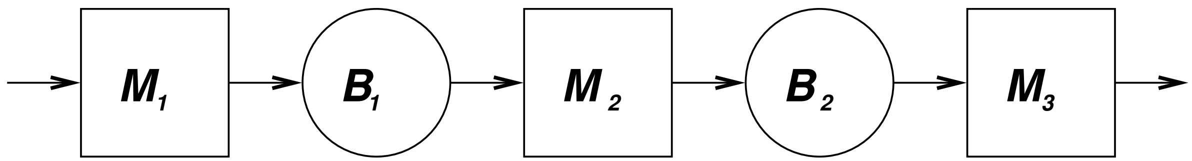


- An operation cannot take place unless there is a token available.
- Tokens *authorize* production.

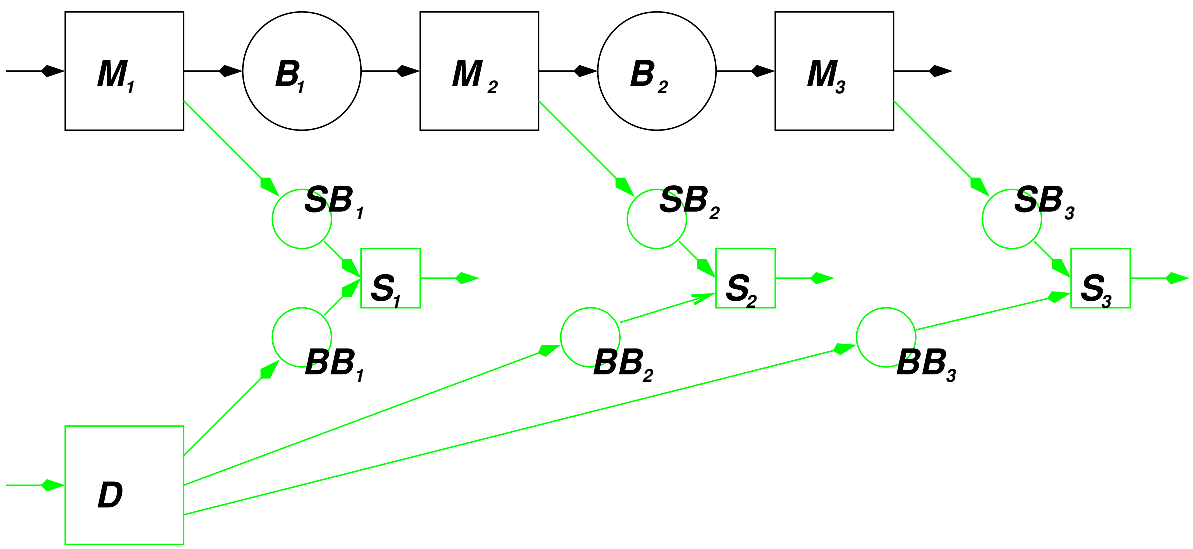
- These policies can often be implemented *either* with finite buffer space, or a finite number of tokens. Mixtures are also possible.
- Buffer space could be shelf space, or floor space indicated with paint or tape.

Proposed policy

To control

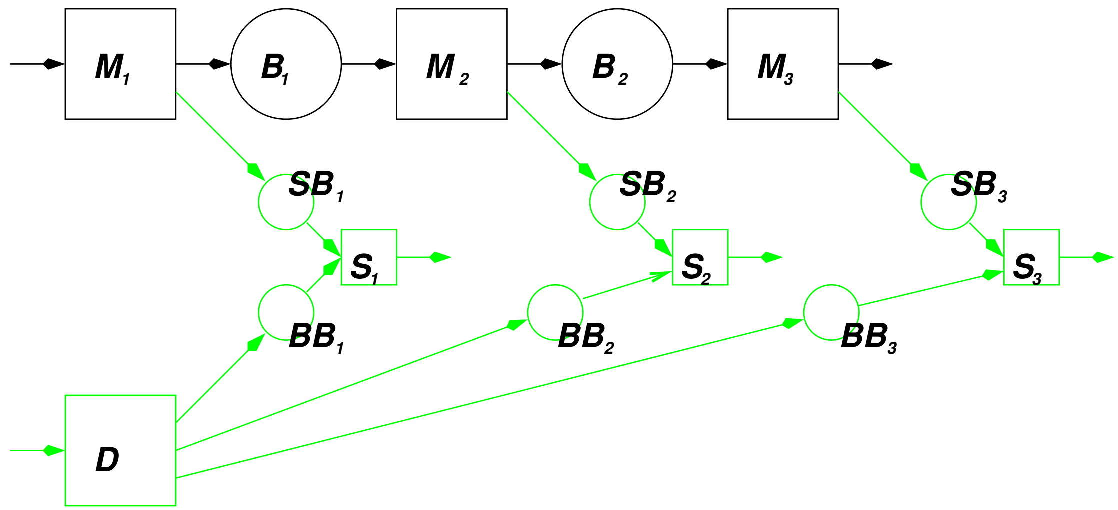


add an information flow system:



Multi-stage systems

Proposed policy



- 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 system.

Three Views of Scheduling

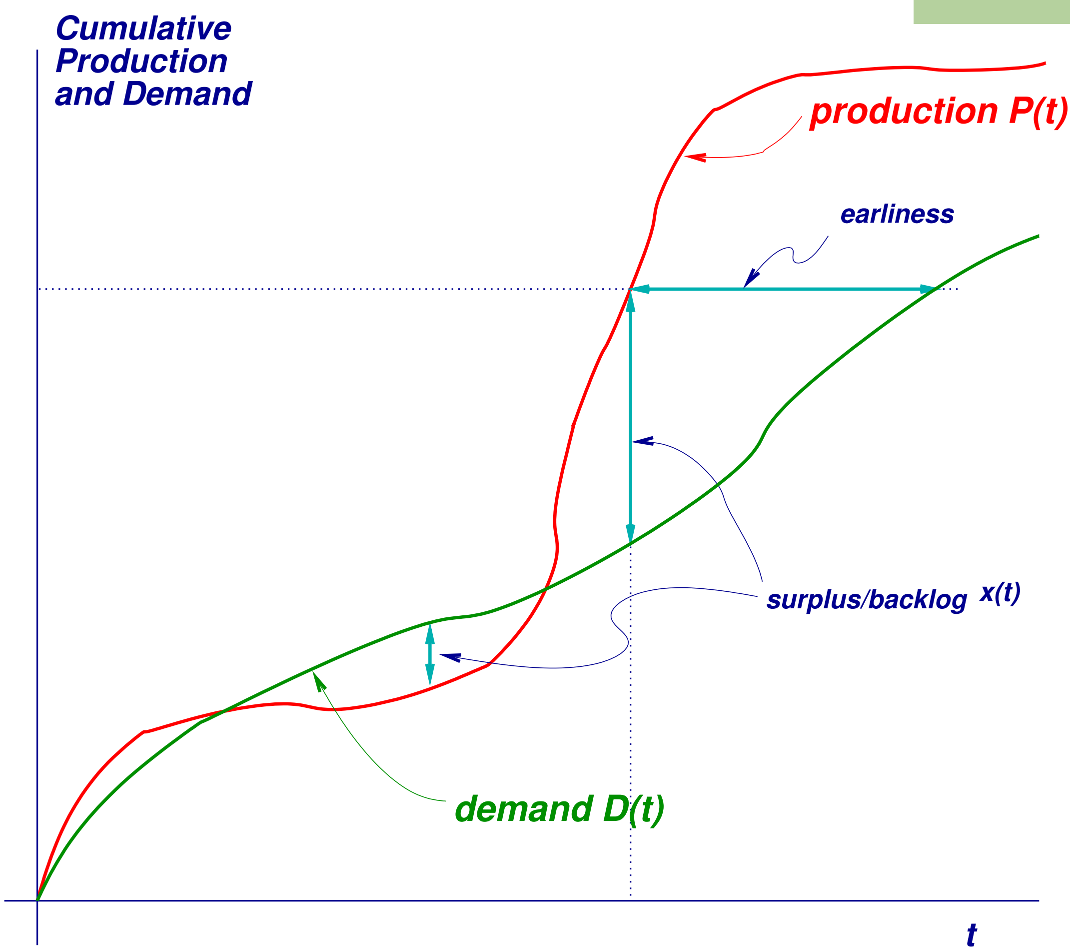
Three kinds of scheduling policies, which are sometimes exactly the same.

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

Multi-stage systems

Objective of Scheduling

Surplus and time

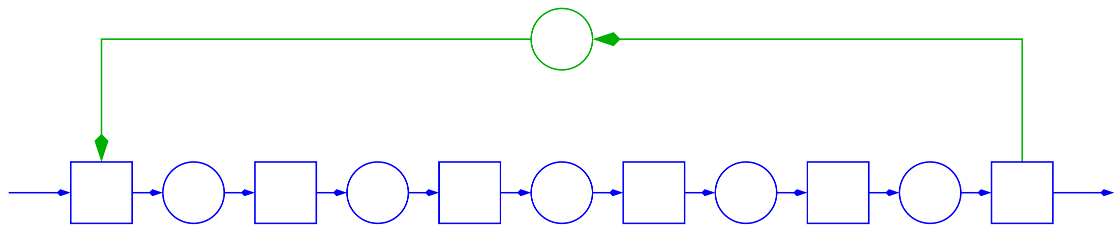


- Objective is to keep cumulative production close to cumulative demand.
- Surplus-based policies look at vertical differences between the graphs.
- Time-based policies look at the horizontal differences.

Multi-stage systems

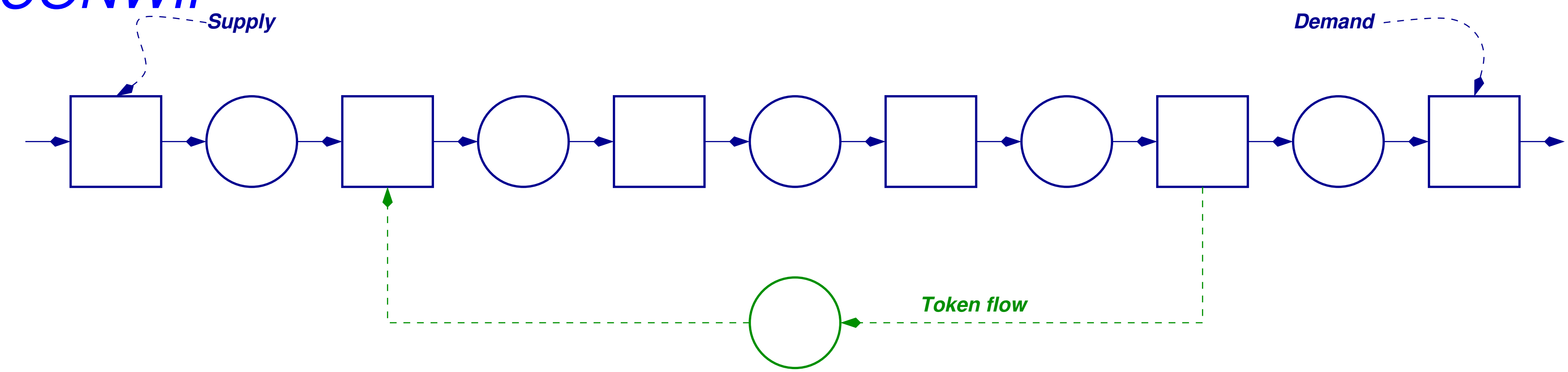
Other policies

CONWIP, kanban, and hybrid



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

CONWIP



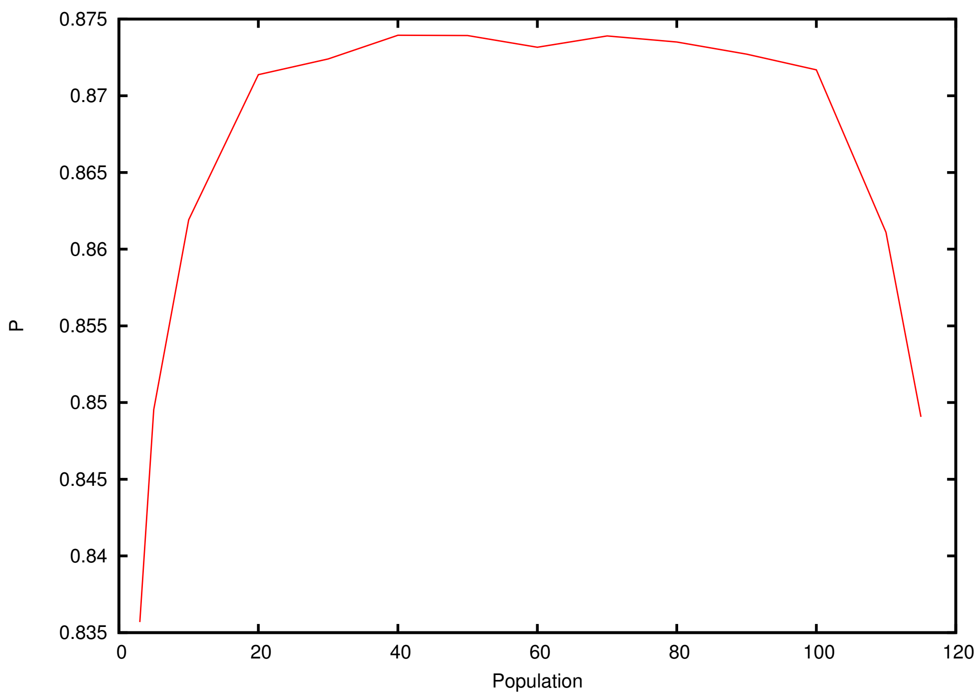
Demand is less than capacity.

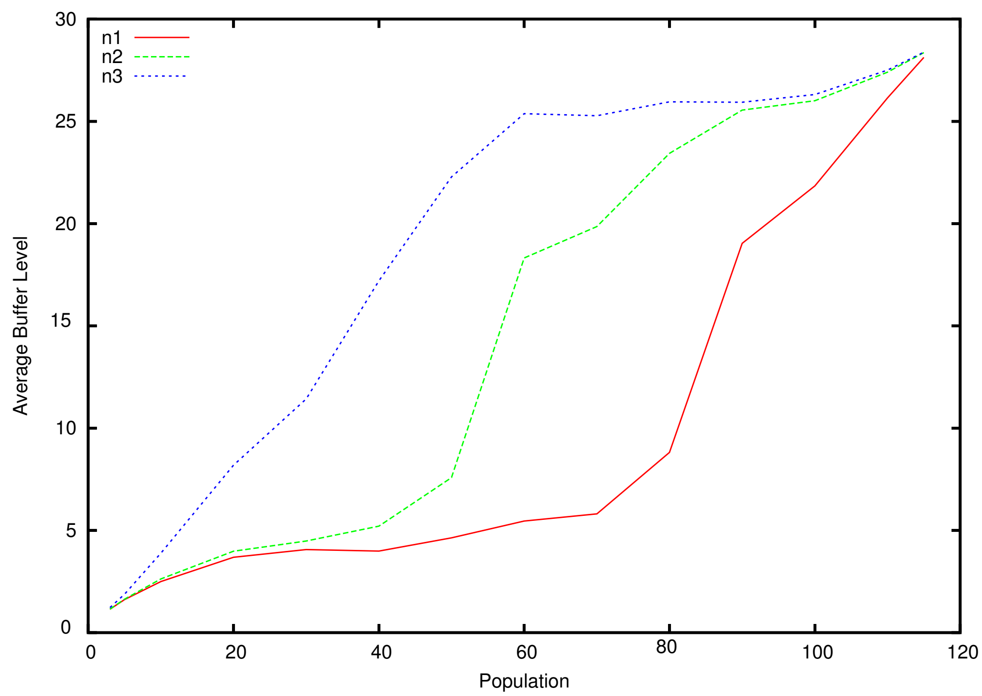
How does the number of tokens affect performance (production rate, inventory)?

Multi-stage systems

Other policies

CONWIP, kanban, and hybrid

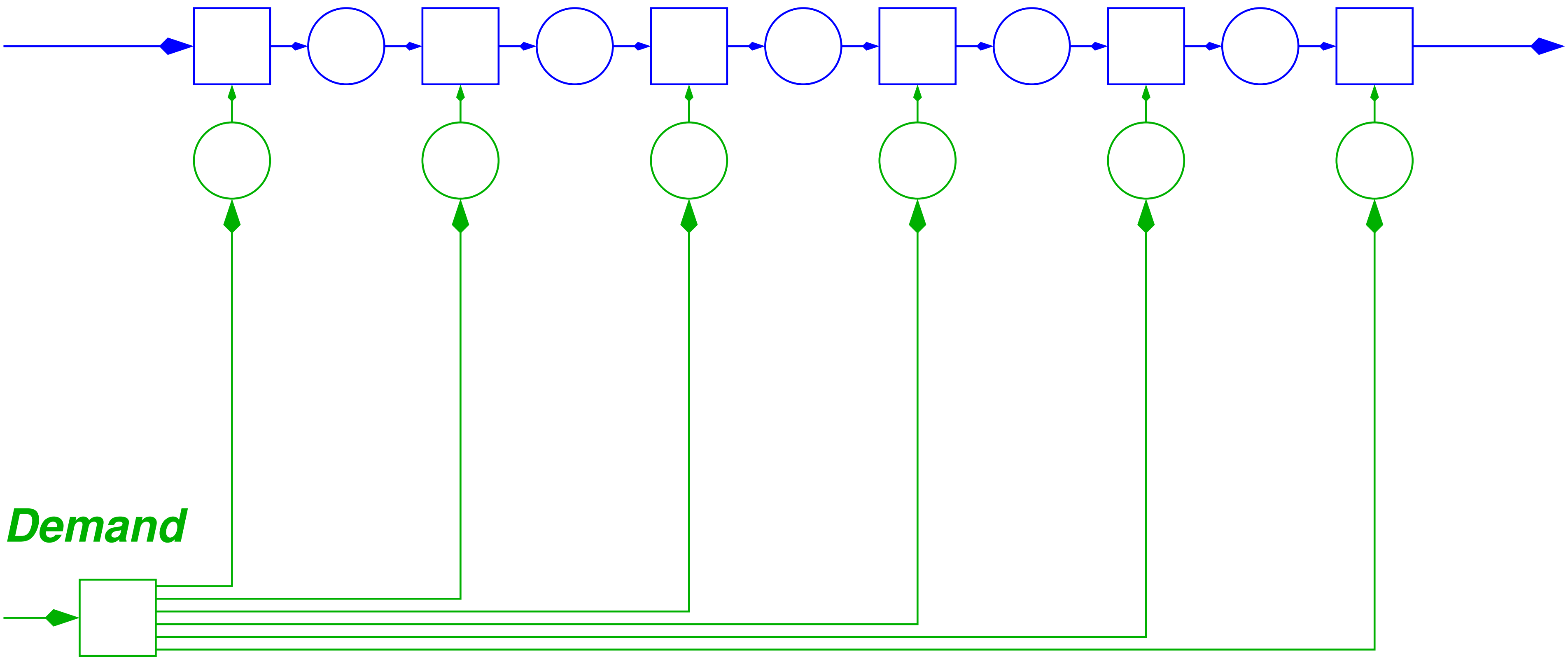




Multi-stage systems

Other policies

Basestock



FIFO

- First-In, First Out.
- Simple conceptually, but you have to keep track of arrival times.
- Leaves out much important information:
  - ⋆ due date, value of part, current surplus/backlog state, etc.

Multi-stage systems

Other policies

EDD

- Earliest due date.
- Easy to implement.
- Does not consider work remaining on the item, value of the item, etc..

Multi-stage systems

Other policies

SRPT

- *Shortest Remaining Processing Time*
- Whenever there is a choice of parts, load the one with least remaining work before it is finished.
- Variations: include waiting time with the work time. Use expected time if it is random.

Multi-stage systems

Other policies

Critical ratio

- Widely used, but many variations. One version:
  - ⋆ Define $\text{CR} = \dfrac{\text{Processing time remaining until completion}}{\text{Due date - Current time}}$
  - ⋆ Choose the job with the highest ratio (provided it is positive).
  - ⋆ If a job is late, the ratio will be negative, or the denominator will be zero, and that job should be given highest priority.
  - ⋆ If there is more than one late job, schedule the late jobs in SRPT order.

Multi-stage systems

Other policies

Least Slack

- This policy considers a part's due date.
- Define *slack* = due date - remaining work time
- When there is a choice, select the part with the least slack.
- Variations involve different ways of estimating remaining time.

Multi-stage systems

Other policies

Drum-Buffer-Rope

- Due to Eli Goldratt.
- Based on the idea that every system has a bottleneck.
- *Drum:* the common production rate that the system operates at, which is the rate of flow of the bottleneck.
- *Buffer:* DBR establishes a CONWIP policy between the entrance of the system and the bottleneck. The buffer is the CONWIP population.
- *Rope:* the limit on the difference in production between different stages in the system.
- But: What if bottleneck is not well-defined?

Conclusions

- Many policies and approaches.
- No simple statement telling which is better.
- Policies are not all well-defined in the literature or in practice.
- My opinion:
  - ⋆ This is because policies are not *derived* from first principles.
  - ⋆ Instead, they are tested and compared.
  - ⋆ Currently, we have little intuition to guide policy development and choice.

2.852 Manufacturing Systems Analysis
Spring 2010