Lecture outline

- Basic inventory problems
- The economic order quantity
- An inventory game
- Multiperiod lot sizing
  - » Math programming formulation
  - » Heuristics
  - » Wagner-Whitin algorithm

Basic inventory problems

- Examples:
  - Products:
    - Customers consume products over time.
    - Store replenishes periodically.
  - People with specialized training:
    - People randomly leave the company over time.
    - Company periodically hires new graduates.
  - Water (management of dams).
    - Rainfall randomly replenishes reservoirs.
    - Release water from dam to maintain level.
  - Oil being stored in storage tanks:
    - Oil is steadily consumed.
    - Periodically is replenished from tankers.

Basic inventory problems

- Examples:
  - » Housing stock
    - Houses are continually being purchased.
    - Developers produce new developments or apartment buildings.
  - » Financial resources (startup company)
    - Cash is used to build up the company.
    - Periodically fresh capital is raised from venture capitalists.
  - » Features in a software program:
    - Accumulating features in a software program in response to user requests and the ideas of developers, or due to bug fixes.
    - Periodically ship a new version of the program.
  - » Purchasing stock:
    - Funds become available for investment.
    - Periodically purchase new shares of stock.

Basic inventory problems

Statement Date	Account Number	Account Summary for the Period	Replenishment Amount	Replenishment Method
1/4/2002		11/4/2001 to 1/3/2002	25.00	AMEX

Tag Deposit	Beginning Balance	Tolls & Fees	Payments & Credits	Ending Balance	Replenishment Threshold
0.00	22.07	27.25	25.00	19.82	10.00

Date/Time	Tag	Transaction	Entry Plaza	Entry Lane	Exit Plaza	Exit Lane	Class	Amount	Balance
11/03 14:20	02200917779	New Jersey Turnpike Toll	9	05E	13A	13X	1	-1.55	20.52
11/06 22:21	02200917779	New Jersey Turnpike Toll	13A	05E	9	13X	1	-1.45	19.07
11/11 15:09	02200917779	New York State Thruway Toll	15	05E	24	08S	1	-3.65	15.42
11/13 14:27	02200917779	New York State Thruway Toll	24	04E	15	08W	1	-3.65	11.77
11/13 17:35	02200917779	New Jersey Turnpike Toll	10	11E	7A	06X	1	-0.75	11.02
12/15 16:59	02200917779	New Jersey Turnpike Toll	9	05E	16E	01X	1	-2.30	8.72
12/15 17:42	02200917779	PANYNJ Toll			LT	01	1	-5.00	3.72
12/16 10:53	02200917779	New Jersey Turnpike Toll	14C	08E	9	12X	1	-2.40	1.32
12/17 16:50	02200917779	New Jersey Turnpike Toll	9	05E	13A	13X	1	-1.55	-0.23
12/18 02:35		Replenishment						25.00	24.77
12/19 20:53	02200917779	New Jersey Turnpike Toll	13A	06E	9	13X	1	-1.45	23.32
12/26 13:44	02200917779	New York State Thruway Toll	15	02E	23	06E	1	-3.50	19.82

Basic inventory problems

- Mutual fund cash balance

Stock market

Cash

Investor

How much cash do we keep on hand to strike a balance between the deposits and withdrawals of investors, and the behavior of the market?

Basic inventory problems

- There are a number of ways to refer to storing resources for the future:
  - » Physical resources
    - Inventory
    - Stockpile
    - Stock
  - » Financial resources
    - Savings
    - Nest egg
    - Reserve

Basic inventory problems

- Reasons for holding inventories:
  - » Economies of scale
    - Batches of goods
    - Discounts (purchasing)
    - Transportation economies (e.g. shipping in bulk)
  - » Uncertainties
    - Demand
    - Order lead times
    - Supply/price of raw materials (OPEC)
    - Supply/price of components (strikes)
    - Quality control
  - » Speculation
    - commodities prices
    - currency fluctuations

Basic inventory problems

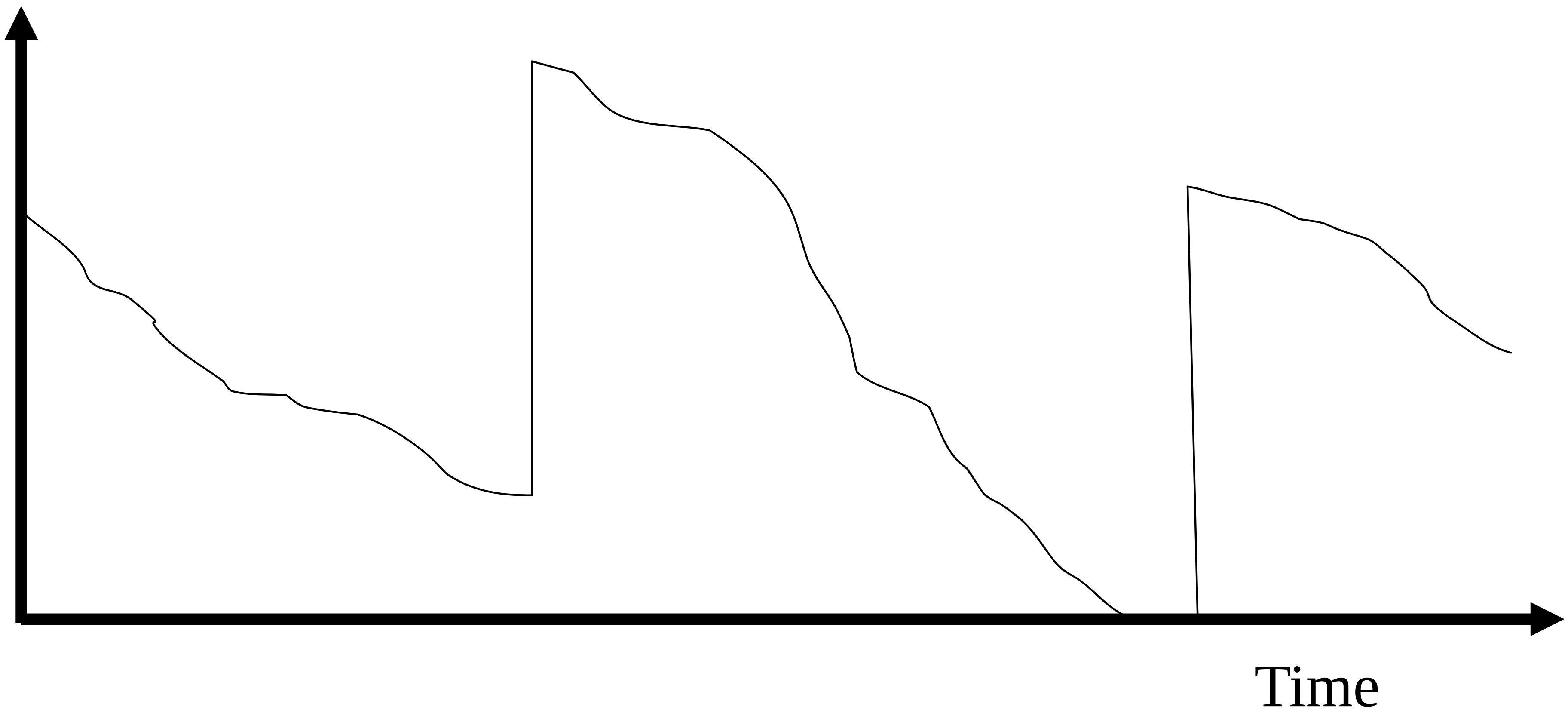
- Reasons for holding inventories
  - » Transportation
    - In-transit or pipeline inventories
  - » Smoothing production
    - Respond to seasonal patterns in demand
    - Seasonal production of some items
      - Certain foods
      - Syrup
      - Snow
      - Students
  - » Control costs
    - Lower inventories requires more sophisticated control systems

Basic inventory problems

- The lot sizing problem
  - » Often, there are economies of scale when ordering new resources:
    - Raising operating capital
      - – There is a fixed cost to going to the capital markets
      - – Just as much work to raise $1m as $5m
    - Shipping the latest version of a software program
      - – New features are added over time
      - – There is a fixed cost of shipping a new version of the code
      - – How many new features do you add before you ship the code?
    - Ordering new product for a store shelf
      - – Fixed cost for placing and shipping an order

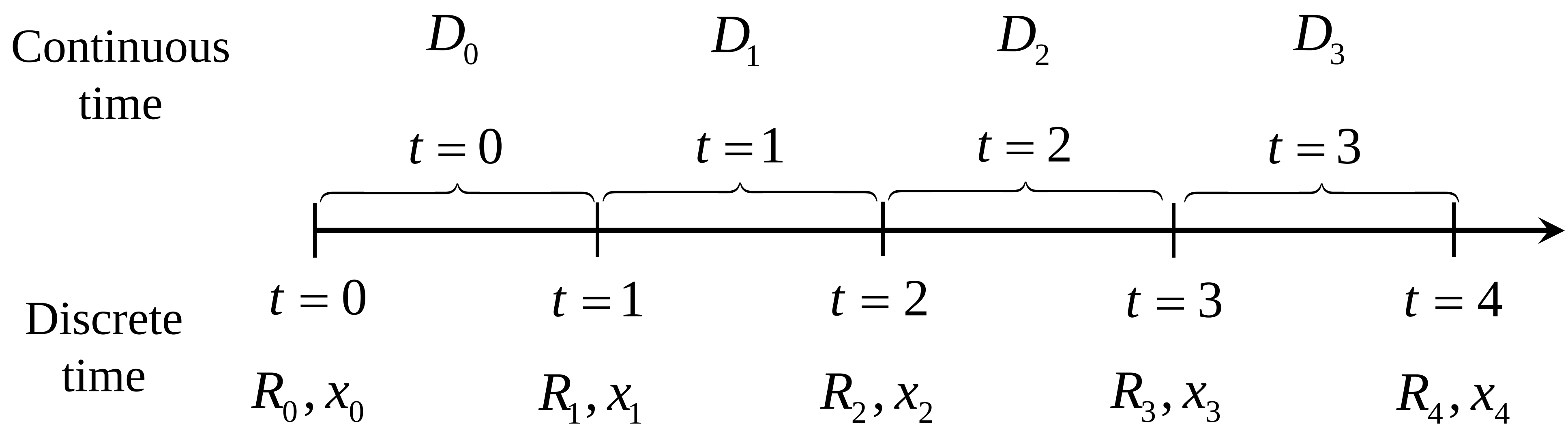
Basic inventory problems

- An aging and replenishment process (negative drift):
  - » State is inventory – drift is due to customer demand.



Basic inventory problems

- Indexing time:
  - » Deterministic indexing – Index based on when something happens.
  - » Stochastic indexing – Index based on when something becomes known.
  - » Deterministic indexing:



Basic inventory problems

- Basic inventory equation
  - » Deterministic indexing

$$R_{t+1} = \max\left\{0, R_t + x_t - D_t\right\} = \left[R_t + x_t - D_t\right]^+$$

Demand (random at time *t*)

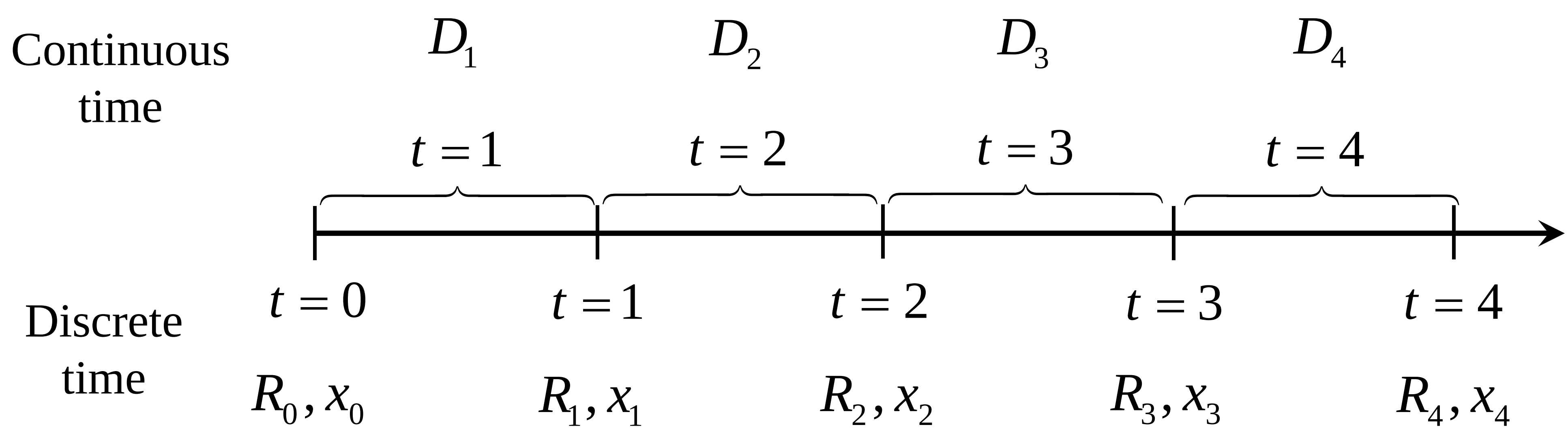
New orders

State (e.g. inventory level)

  - » In deterministic indexing, everything is modeled at the beginning of a time period.

Basic inventory problems

- Stochastic indexing
  - » Information arrives continuously over time
  - » *A variable indexed by t contains exogenous information up through time t.*



  - » In stochastic indexing, everything is indexed at the end of a period.

Basic inventory problems

- Basic inventory equation
  - » Stochastic indexing

$$R_{t+1} = \max\left\{0, R_t + x_t - D_{t+1}\right\} = \left[R_t + x_t - D_{t+1}\right]^+$$

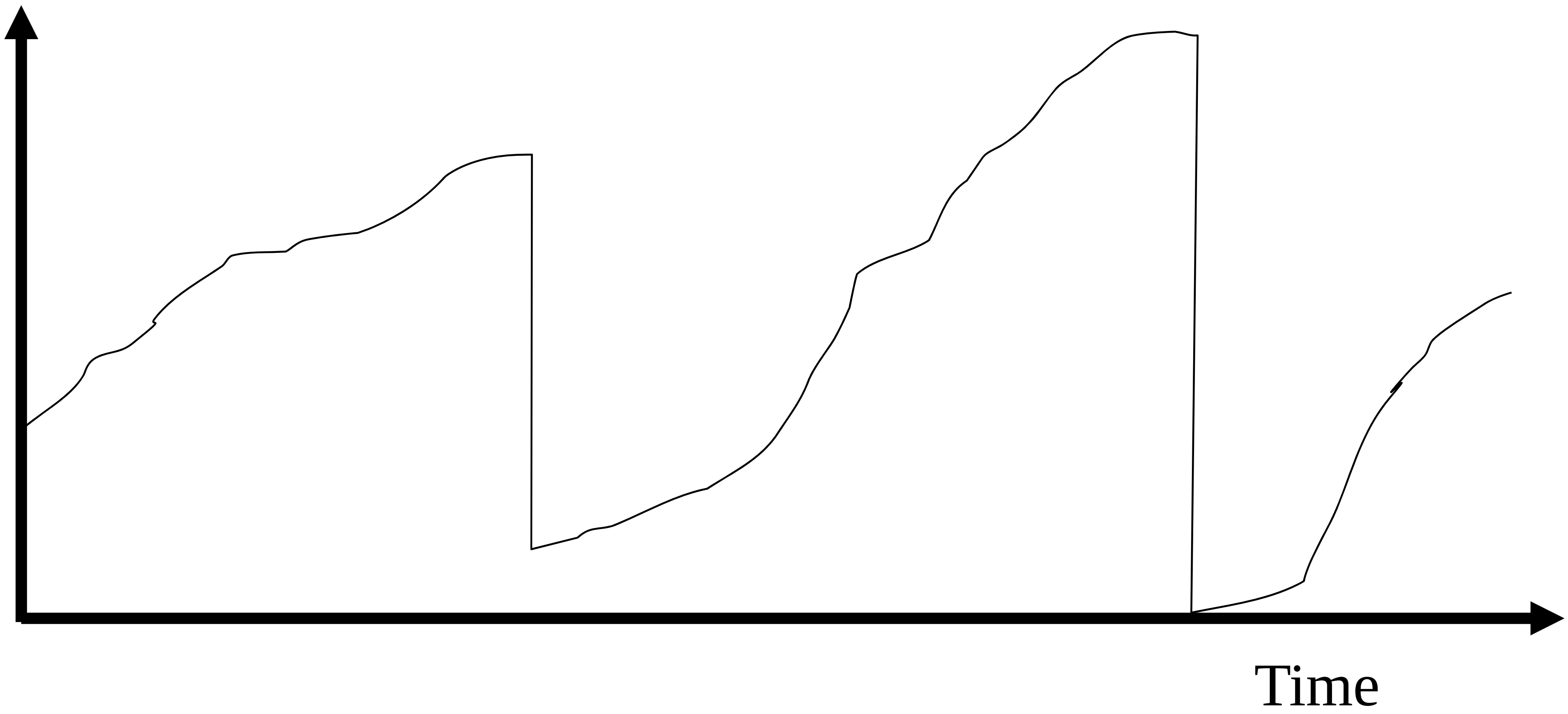
Demand (random at time *t*)

New orders

State (e.g. inventory level)

Basic inventory problems

- Inventory with positive drift
  - » Cash is deposited in an account. Periodically the cash is invested in batch amounts to reduce transaction costs.



Basic inventory problems

- Basic inventory with positive drift
  - » Blood donations, water reservoirs, …

$$R_{t+1} = \left[R_t - x_t \right]^+ + D_{t+1}$$

Exogenous drift

Endogenously controlled shift

State (e.g. accumulation)

Outline

- Basic inventory problems
- The economic order quantity
- An inventory game
- Multiperiod lot sizing
  - » Math programming formulation
  - » Heuristics
  - » Wagner-Whitin algorithm

The economic order quantity

- Order costs
  - » Fixed cost of placing an order
    - Paperwork, forms, telephone
    - Sending a truck out
    - Setting up a machine
    - Price structure imposed by supplier
  - » Variable cost of ordering a certain amount of product
    - Variable cost may be fixed (a linear function).
    - Or there may be economies of placing larger orders.

The economic order quantity

- Holding cost
  - » Storage (heat, electricity, supervision, etc.)
  - » Taxes and insurance
  - » Breakage, spoilage, deterioration and obsolescence
    - Careful with these - these "costs" convert to reduction in quantity.
  - » Opportunity cost (interest)
    - Hurdle rate for a company is generally much higher than bank interest rates.
    - Let I = "interest rate"
    - c^p = purchase cost of item
    - $c^h = Ic^p$ = holding cost (be careful with units; if I is interest rate per year, c^h is holding cost per *year*).

The economic order quantity

- Stockout cost
  - » Cost of lost customers
  - » Cost of pushing orders to future time periods
- Notes:
  - » Stockout costs are not relevant in our simplest inventory system, because they cannot happen.
  - » Stockouts arise when:
    - Demand is random
    - Demand varies over time with production capacities
    - Order costs may vary as a function of time, possibly exceeding the “benefit” of covering demand.

The economic order quantity

- Some assumptions:
  » Demand is deterministic with rate λ per unit time.

  » Rate λ is constant – stationary process.

  » Costs are stationary.

  » Orders arrive immediately.

  » All orders must be filled.

The economic order quantity

The basic inventory equation:

$$R_{t+1} = [R_t + x_t - D_{t+1}]^+$$

where:

R_t = Inventory at start of time t

x_t = Amount ordered at time t

D_t = Demand during period t

Notation:

$$[x]^+ = \max\{x, 0\}$$

The economic order quantity

- The cost function:

$$c\ (x_t, R_t) = \text{Total costs during period } t \text{ given order quantity } x_t \text{ and initial inventory } R_t$$

$$= c^o(x_t) + c^h(R_t, x_t)$$

where:

$$c^o(x_t) = \text{Order costs}$$

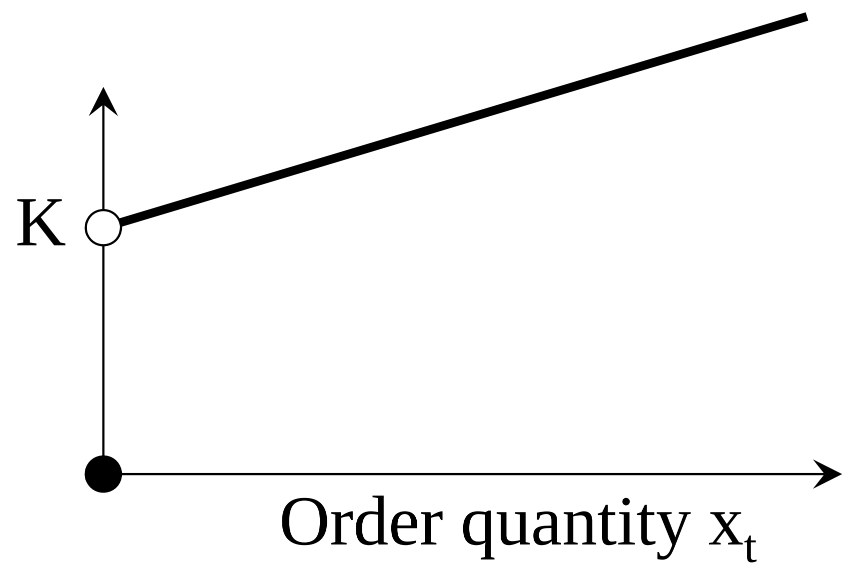
$$= \begin{cases} K + c^p x_t & x_t > 0 \\ 0 & x_t = 0 \end{cases}$$

$$c^p = \text{Unit purchasing costs}$$

$$c^h(R_t, x_t) = \text{Holding costs}$$

$$= c^h \int_0^{\Delta t} \left[R_t + x_t - \lambda z \right]^+ dz$$

$$c^h = \text{Unit holding costs per time}$$



The economic order quantity

- The cost function:

We can transform the nonlinear cost function into a linear one:

$$c^o(x_t) \quad = \ Ky_t + c^p x_t$$

where:

$$x_t \geq 0$$

$$x_t \leq My_t \qquad \text{M} \ = \ \text{"big M"}$$

$$y_t \in (0,1)$$

If $y_t = 0$ then we force $x_t \ = \ 0$. Now we have transformed a nonlinear cost function into a linear one, but we have added an integer variable.

The economic order quantity

- Infinite horizon problem:

We would like to solve:

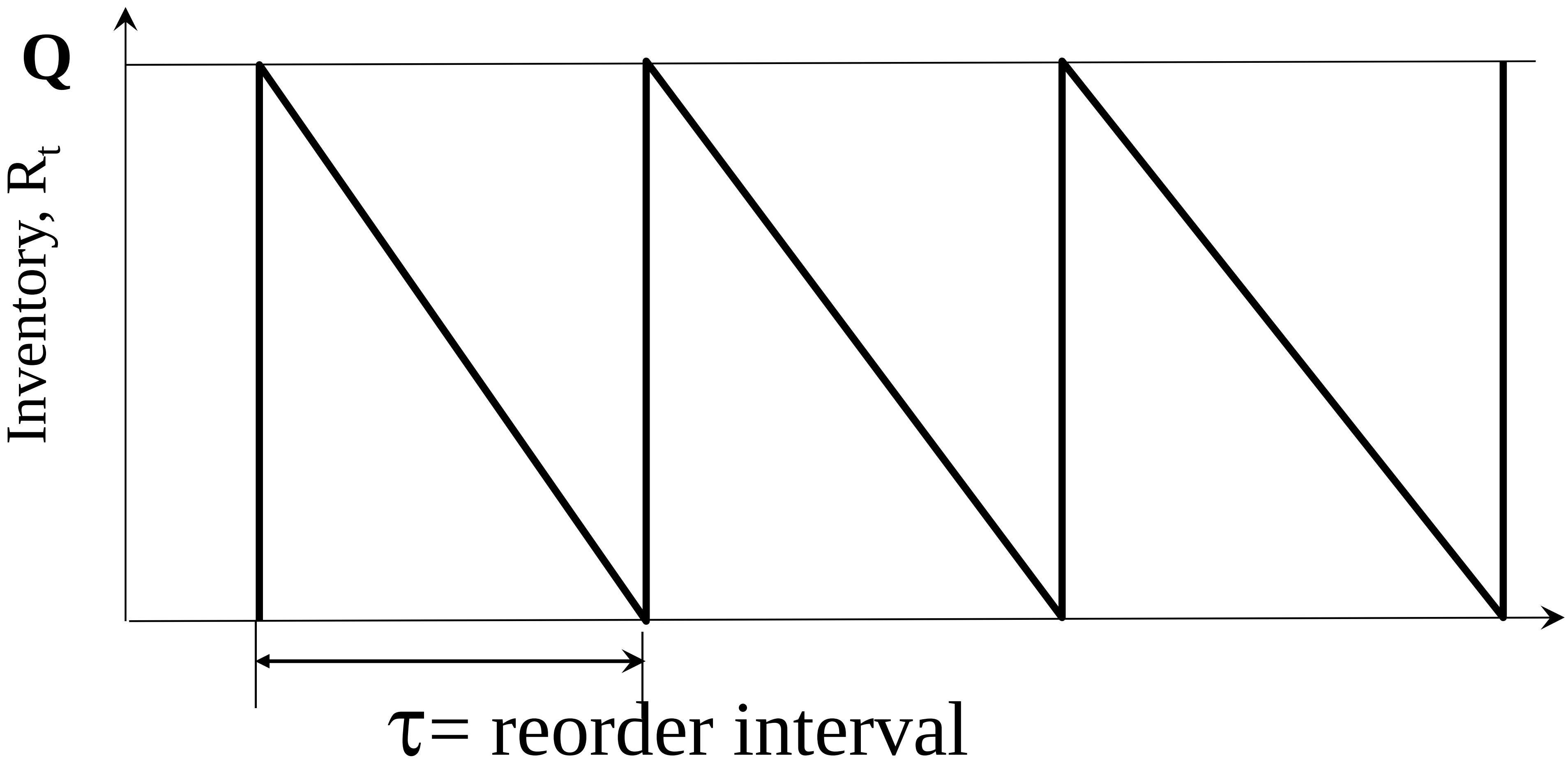
$$\min_{x_t, y_t} \sum_{t=0}^{\infty} c\left(R_t, x_t, y_t\right)$$

This is a really big number! A more formal way to write it is as an average cost:

$$\lim_{T \to \infty} \frac{1}{T} \left\{ \min_{x_t, y_t} \sum_{t=0}^{T} c\left(R_t, x_t, y_t\right) \right\}$$

The economic order quantity

Intuition suggests that we let inventories drop to zero, and then "order up to" an amount Q:



The economic order quantity

Reformulate the problem in terms of cost per order interval:

$Q =$ order quantity

$\lambda =$ Demand rate per unit time (assumed constant and deterministic)

$$c^o(Q) = \text{Order cost per order interval}$$

$$= \text{K} \quad \text{(since we always make one order per interval).}$$

$$\tau = \text{Length of order interval}$$

$$= \frac{\text{Q}}{\lambda}$$

$$c^p(Q) = \text{Purchase costs per order interval}$$

$$= \text{c}^\text{p} Q$$

The economic order quantity

$$c^{h}(Q) = \text{ Holding cost per order interval}$$

$$= \text{ (holding cost)*(average inventory)*(length of interval)}$$

$$= c^{h}\left(\frac{Q}{2}\right)\tau$$

$$= c^{h}\left(\frac{Q}{2}\right)\left(\frac{Q}{\lambda}\right) = c^{h}\left(\frac{Q^2}{2\lambda}\right)$$

$$c(Q) = \text{ Total cost per order interval}$$

$$= c^{o}(Q) + c^{p}(Q) + c^{h}(Q)$$

$$= K + c^{p}Q + c^{h}\left(\frac{Q^2}{2\lambda}\right)$$

The economic order quantity

How do we find the optimum value of Q?

We can try differentiating C(Q) with respect to Q:

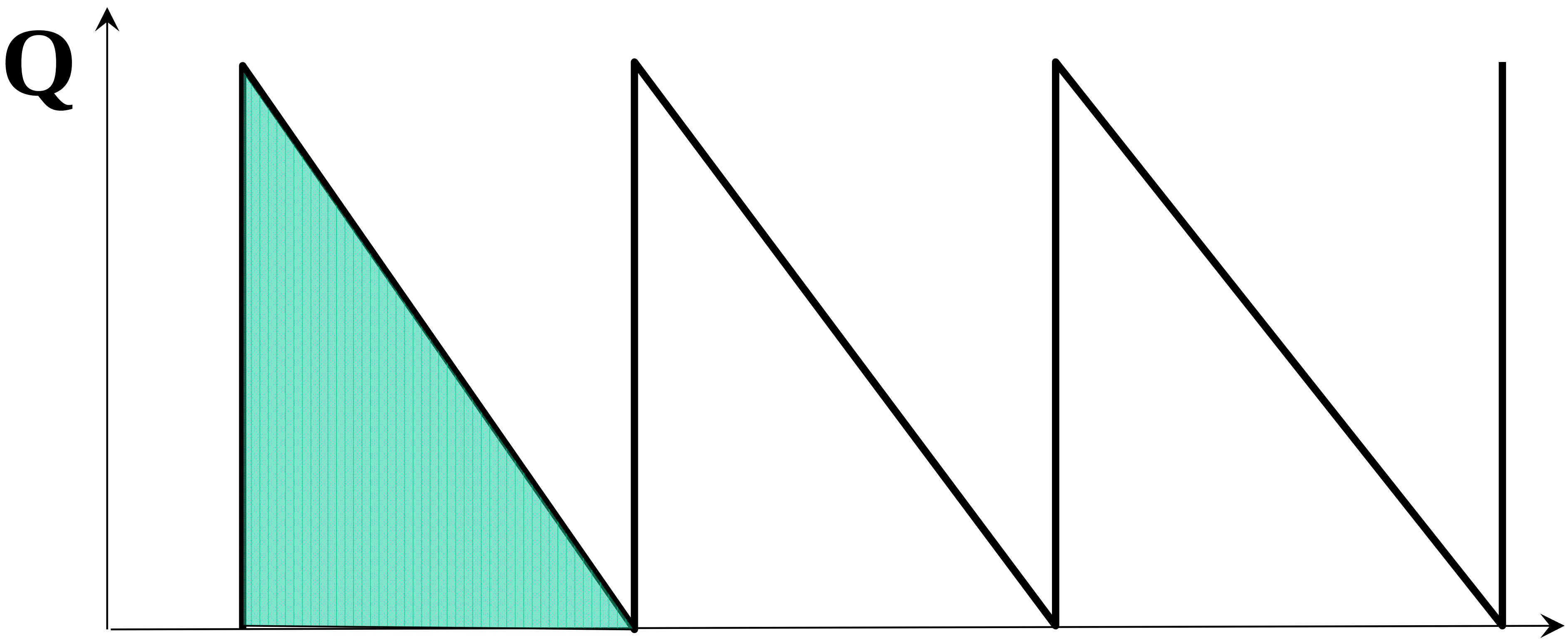
$$\frac{dC(Q)}{dQ} = 0 + c^p + 2c^h \frac{Q}{2\lambda} = 0$$

Solving for Q gives us:

$$\mathrm{Q}^* = -\frac{c^p \lambda}{c^h} \qquad\qquad !!!!!$$

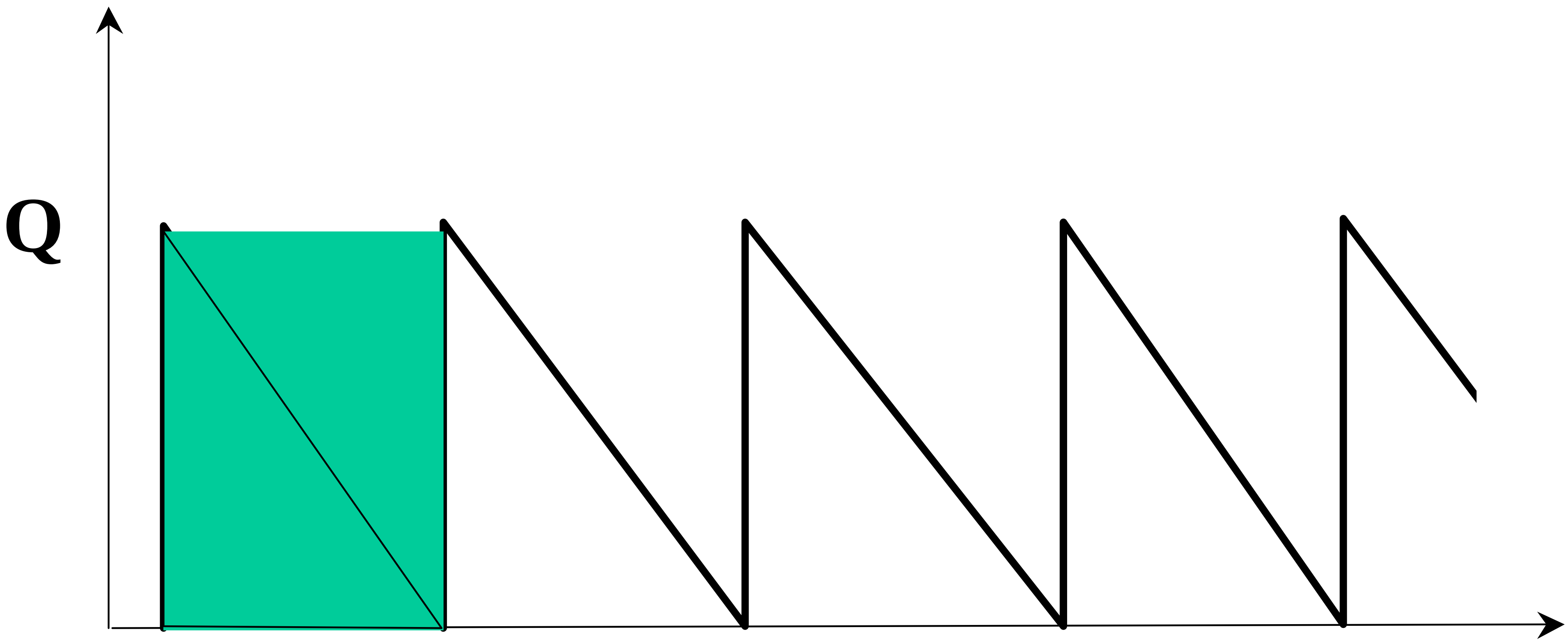
What went wrong?

The economic order quantity

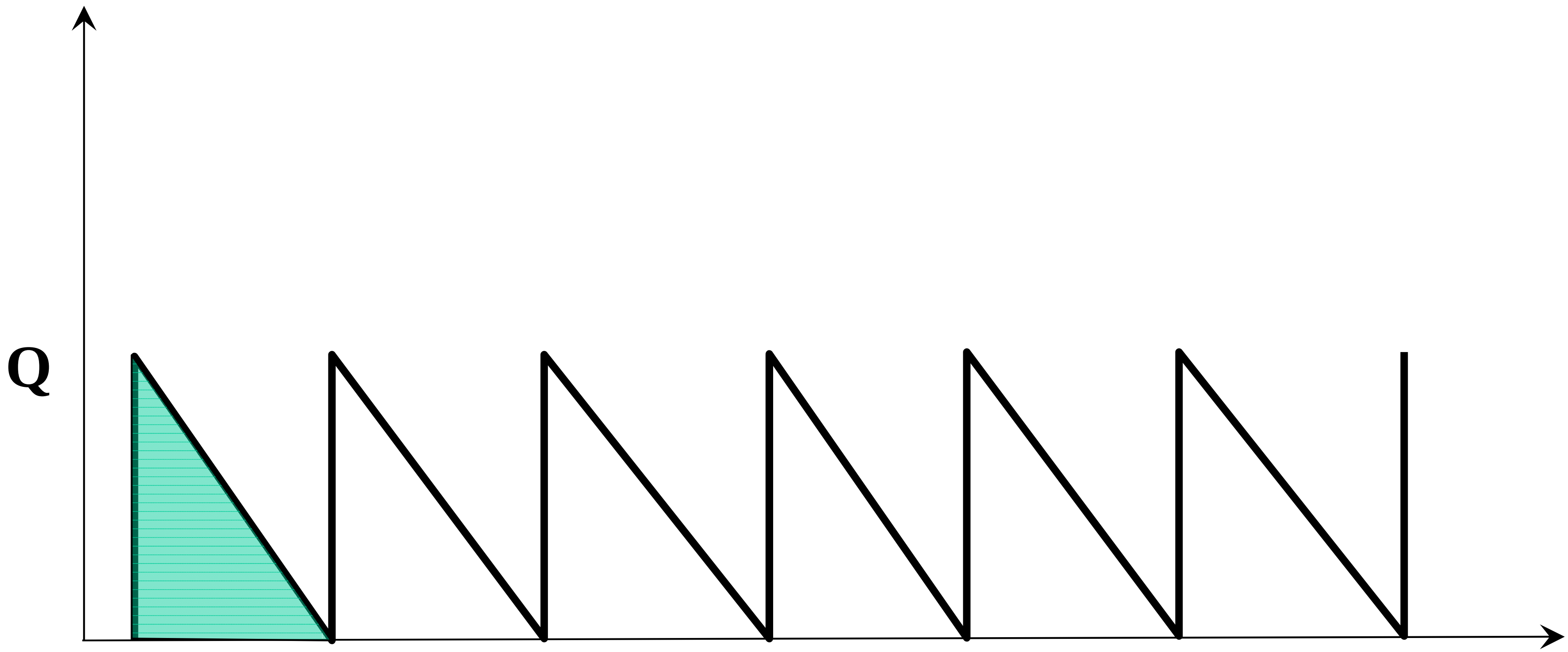


Cost per cycle =K plus quantity proportional to green area

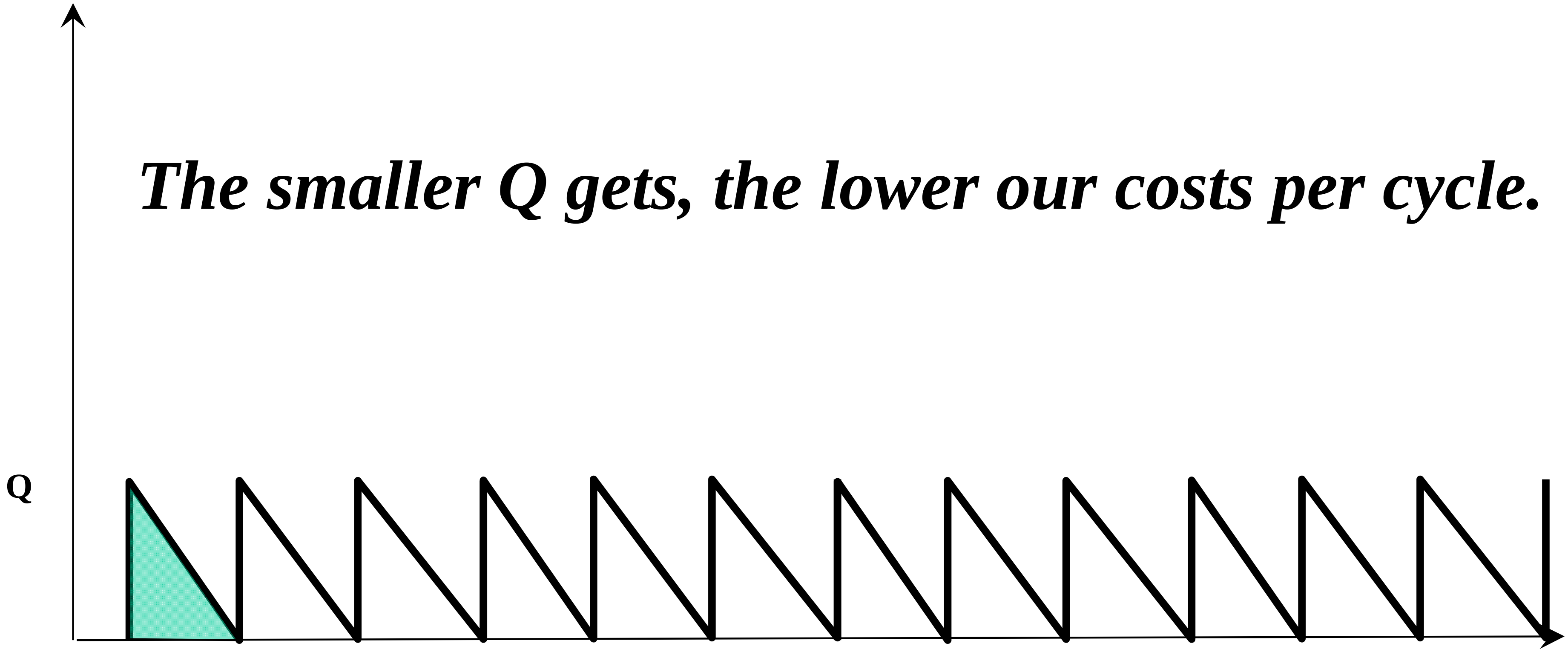
The economic order quantity



The economic order quantity



The economic order quantity



The smaller Q gets, the lower our costs per cycle.

The economic order quantity

Need to minimize cost per unit time, not cost per order interval.

So, we want to solve:

$$\min C^{\tau}(Q) = \frac{C(Q)}{\tau} = \frac{C(Q)}{Q/\lambda} = \frac{\lambda C(Q)}{Q}$$

$$= \frac{\lambda}{Q}\left(K + c^{p}Q + c^{h}\left(\frac{Q^2}{2\lambda}\right)\right)$$

$$= \frac{\lambda K}{Q} + \lambda c^{p} + c^{h}\frac{Q}{2}$$

Differentiating with respect to Q and setting to 0:

$$\frac{\mathrm{d}C^{\tau}(Q)}{dQ} = -\frac{\lambda K}{Q^2} + \frac{c^h}{2} = 0$$

The economic order quantity

Finally, solving for Q gives us:

$$Q = \sqrt{\frac{2K\lambda}{c^h}} = \text{ The Economic Order Quantity (EOQ)}$$

Also called the Economic Lot Size.

Properties of the optimal solution:

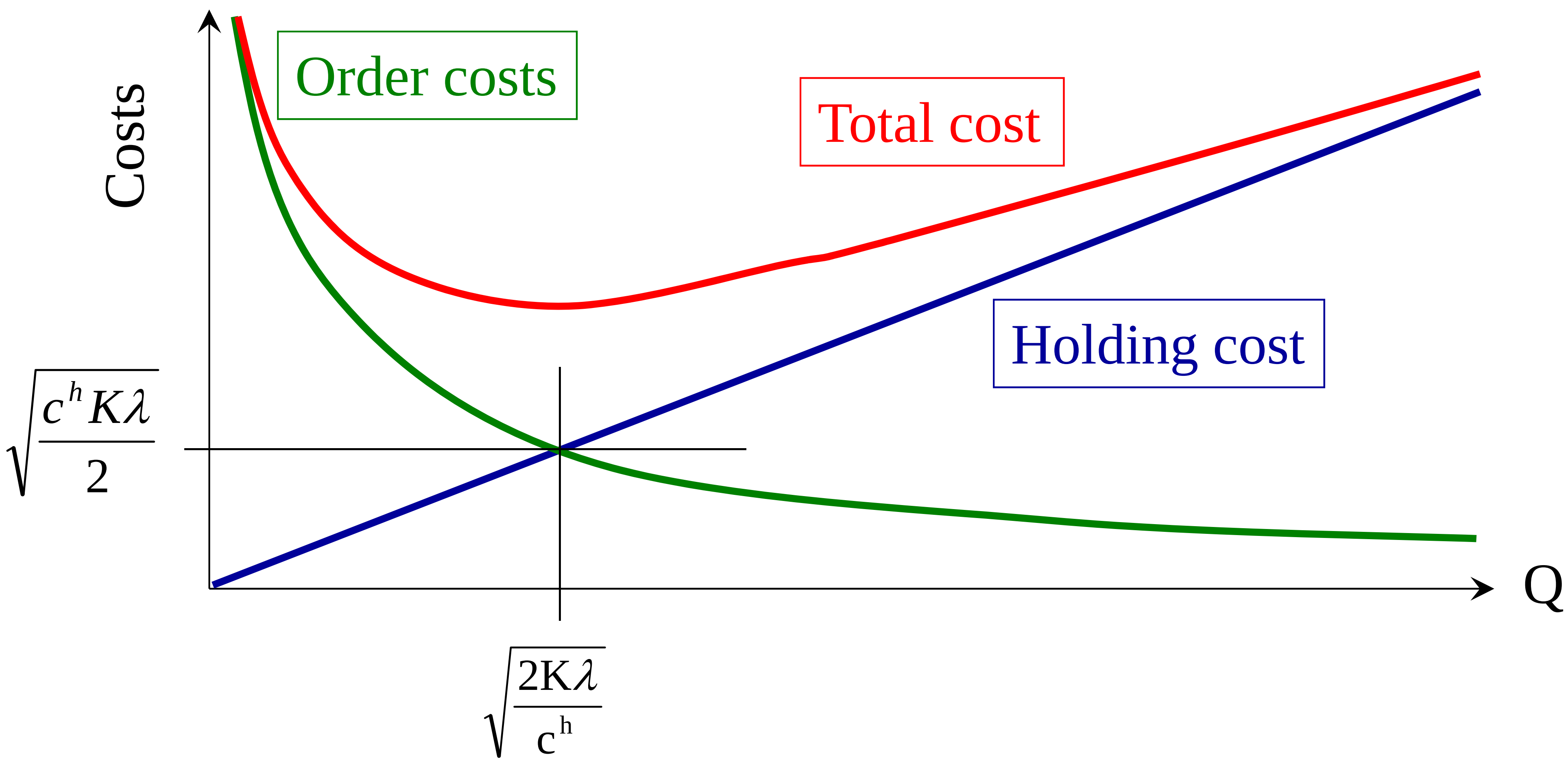
1. Purchase costs do not enter the equation (why?)
2. Order costs per unit time = holding costs per unit time:

$$\text{Order costs per unit time } = \frac{\lambda K}{Q} = \frac{\lambda K}{\sqrt{\frac{2K\lambda}{c^h}}} = \sqrt{\frac{c^h K \lambda}{2}}$$

$$\text{Holding costs per unit time } = c^h \frac{Q}{2} = \frac{c^h}{2}\sqrt{\frac{2K\lambda}{c^h}} = \sqrt{\frac{c^h K \lambda}{2}}$$

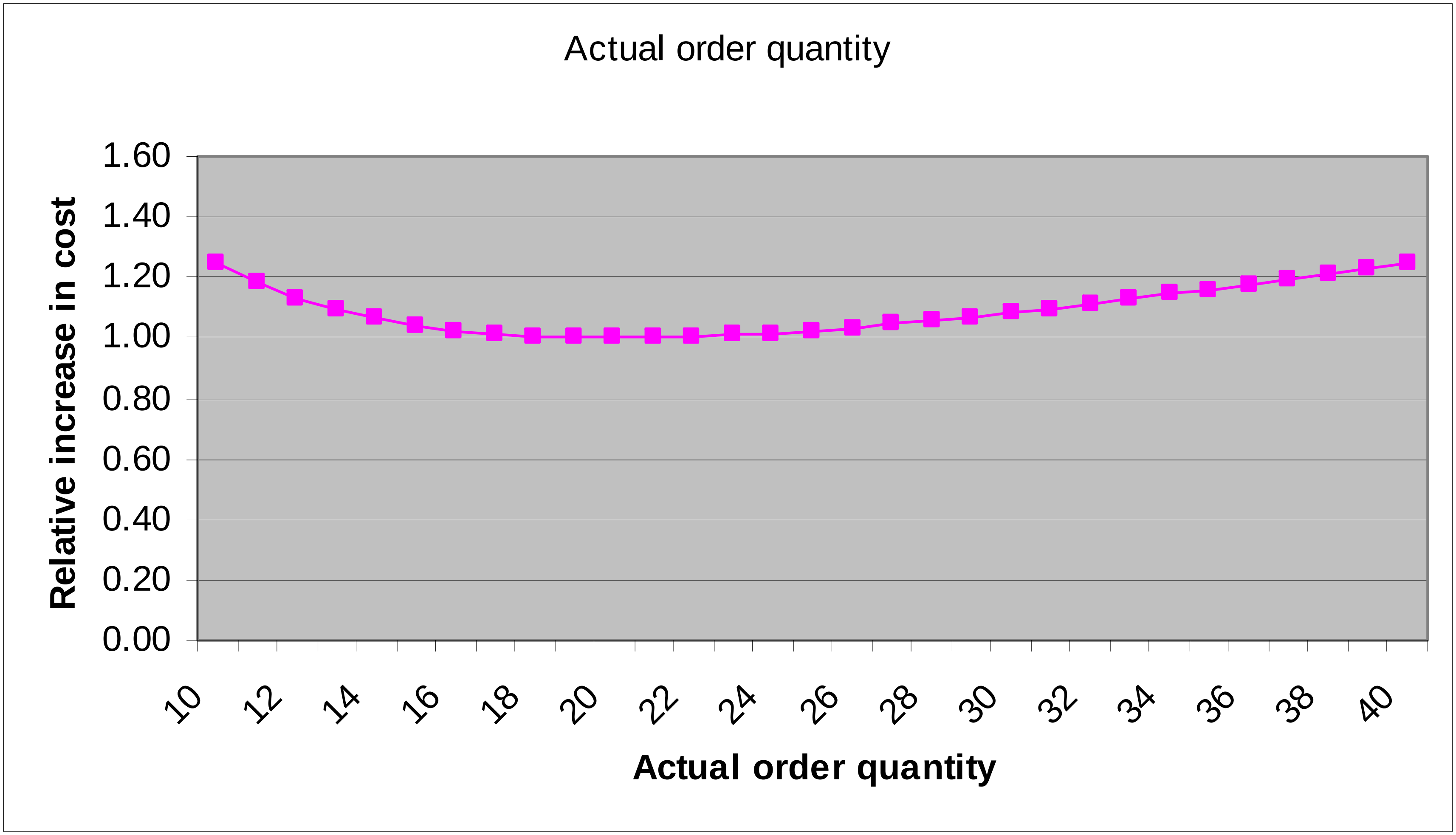
The economic order quantity

- The average cost function:



The economic order quantity

- Sensitivity analysis



Outline

- Basic inventory problems
- The economic order quantity
- An inventory game
- Multiperiod lot sizing
  - » Math programming formulation
  - » Heuristics
  - » Wagner-Whitin algorithm

An inventory game

- The basics:
  - » Random demand (uniform between 0 and 10)
  - » Demands are not revealed until an order is entered.
  - » Cost parameters:

Order cost	25
Purchase	10
Holding	1
Stockout	5

An inventory game Player: Joe

Time	Invento	Order	Demar	Ending	Sold	Lost	Order	Purcha	Holdin	Stocko	Total	Cum	
0	20	0	2	18	2	0	0	0	18	0	18	18	
1	18	0	1	17	1	0	0	0	17	0	17	35	
2	17	0	8	9	8	0	0	0	9	0	9	44	
3	9	0	3	6	3	0	0	0	6	0	6	50	
4	6	0	2	4	2	0	0	0	4	0	4	54	
5	4	4	6	2	6	0	25	40	2	0	67	121	
6	2	6	4	4	4	0	25	60	4	0	89	210	
7	4	4	5	3	5	0	25	40	3	0	68	278	
8	3	5	8	0	8	0	25	50	0	0	75	353	
9	0	8	6	2	6	0	25	80	2	0	107	460	
10	2	6	2	6	2	0	25	60	6	0	91	551	
11	6	2	3	5	3	0	25	20	5	0	50	601	
12	5	3	5	3	5	0	25	30	3	0	58	659	
13	3	5	5	3	5	0	25	50	3	0	78	737	
14	3	5	0	8	0	0	25	50	8	0	83	820	
15	8	0	2	6	2	0	0	0	6	0	6	826	
16	6	2	9	0	8	1	25	20	0	5	50	876	
17	0	8	1	7	1	0	25	80	7	0	112	988	
18	7	1	6	2	6	0	25	10	2	0	37	1025	
19	2	5	4	3	4	0	25	50	3	0	78	1103	
20	3	4	8	0	7	1	25	40	0	5	70	1173	
							375	680	108	10	1173		

An inventory game Player Jimmie:

Time	Invento	Order	Demar	Ending	Sold	Lost	Order	Purcha	Holdin	Stocko	Total	Cum	
0	20	0	2	18	2	0	0	0	18	0	18	18	
1	18	0	1	17	1	0	0	0	17	0	17	35	
2	17	0	8	9	8	0	0	0	9	0	9	44	
3	9	0	3	6	3	0	0	0	6	0	6	50	
4	6	0	2	4	2	0	0	0	4	0	4	54	
5	4	20	6	18	6	0	25	200	18	0	243	297	
6	18	0	4	14	4	0	0	0	14	0	14	311	
7	14	0	5	9	5	0	0	0	9	0	9	320	
8	9	0	8	1	8	0	0	0	1	0	1	321	
9	1	20	6	15	6	0	25	200	15	0	240	561	
10	15	0	2	13	2	0	0	0	13	0	13	574	
11	13	0	3	10	3	0	0	0	10	0	10	584	
12	10	0	5	5	5	0	0	0	5	0	5	589	
13	5	20	5	20	5	0	25	200	20	0	245	834	
14	20	0	0	20	0	0	0	0	20	0	20	854	
15	20	0	2	18	2	0	0	0	18	0	18	872	
16	18	0	9	9	9	0	0	0	9	0	9	881	
17	9	0	1	8	1	0	0	0	8	0	8	889	
18	8	0	6	2	6	0	0	0	2	0	2	891	
19	2	10	4	8	4	0	25	100	8	0	133	1024	
20	8	0	8	0	8	0	0	0	0	0	0	1024	
							100	700	224	0	1024		

An inventory game   Player:

Time	Invento	Order	Demar	Ending	Sold	Lost	Order	Purcha	Holdin	Stocko	Total	Cum	
0	20	-1	0	20	-1	0	0	0	20	0	20		
1	20	-1	0	20	-1	0	0	0	20	0	20		
2	20	-1	0	20	-1	0	0	0	20	0	20		
3	20	-1	0	20	-1	0	0	0	20	0	20		
4	20	-1	0	20	-1	0	0	0	20	0	20		
5	20	-1	0	20	-1	0	0	0	20	0	20		
6	20	-1	0	20	-1	0	0	0	20	0	20		
7	20	-1	0	20	-1	0	0	0	20	0	20		
8	20	-1	0	20	-1	0	0	0	20	0	20		
9	20	-1	0	20	-1	0	0	0	20	0	20		
10	20	-1	0	20	-1	0	0	0	20	0	20		
11	20	-1	0	20	-1	0	0	0	20	0	20		
12	20	-1	0	20	-1	0	0	0	20	0	20		
13	20	-1	0	20	-1	0	0	0	20	0	20		
14	20	-1	0	20	-1	0	0	0	20	0	20		
15	20	-1	0	20	-1	0	0	0	20	0	20		
16	20	-1	0	20	-1	0	0	0	20	0	20		
17	20	-1	0	20	-1	0	0	0	20	0	20		
18	20	-1	0	20	-1	0	0	0	20	0	20		
19	20	-1	0	20	-1	0	0	0	20	0	20		
20	20	-1	0	20	-1	0	0	0	20	0	20		
							0	0	420	0	420		

An inventory game   Player:

Time	Invento	Order	Demar	Ending	Sold	Lost	Order	Purcha	Holdin	Stocko	Total	Cum	
0	20	-1	0	20	-1	0	0	0	20	0	20		
1	20	-1	0	20	-1	0	0	0	20	0	20		
2	20	-1	0	20	-1	0	0	0	20	0	20		
3	20	-1	0	20	-1	0	0	0	20	0	20		
4	20	-1	0	20	-1	0	0	0	20	0	20		
5	20	-1	0	20	-1	0	0	0	20	0	20		
6	20	-1	0	20	-1	0	0	0	20	0	20		
7	20	-1	0	20	-1	0	0	0	20	0	20		
8	20	-1	0	20	-1	0	0	0	20	0	20		
9	20	-1	0	20	-1	0	0	0	20	0	20		
10	20	-1	0	20	-1	0	0	0	20	0	20		
11	20	-1	0	20	-1	0	0	0	20	0	20		
12	20	-1	0	20	-1	0	0	0	20	0	20		
13	20	-1	0	20	-1	0	0	0	20	0	20		
14	20	-1	0	20	-1	0	0	0	20	0	20		
15	20	-1	0	20	-1	0	0	0	20	0	20		
16	20	-1	0	20	-1	0	0	0	20	0	20		
17	20	-1	0	20	-1	0	0	0	20	0	20		
18	20	-1	0	20	-1	0	0	0	20	0	20		
19	20	-1	0	20	-1	0	0	0	20	0	20		
20	20	-1	0	20	-1	0	0	0	20	0	20		
							0	0	420	0	420		

Outline

- Basic inventory problems
- The economic order quantity
- An inventory game
- Multiperiod lot sizing
  - » Math programming formulation
  - » Heuristics
  - » Wagner-Whitin algorithm

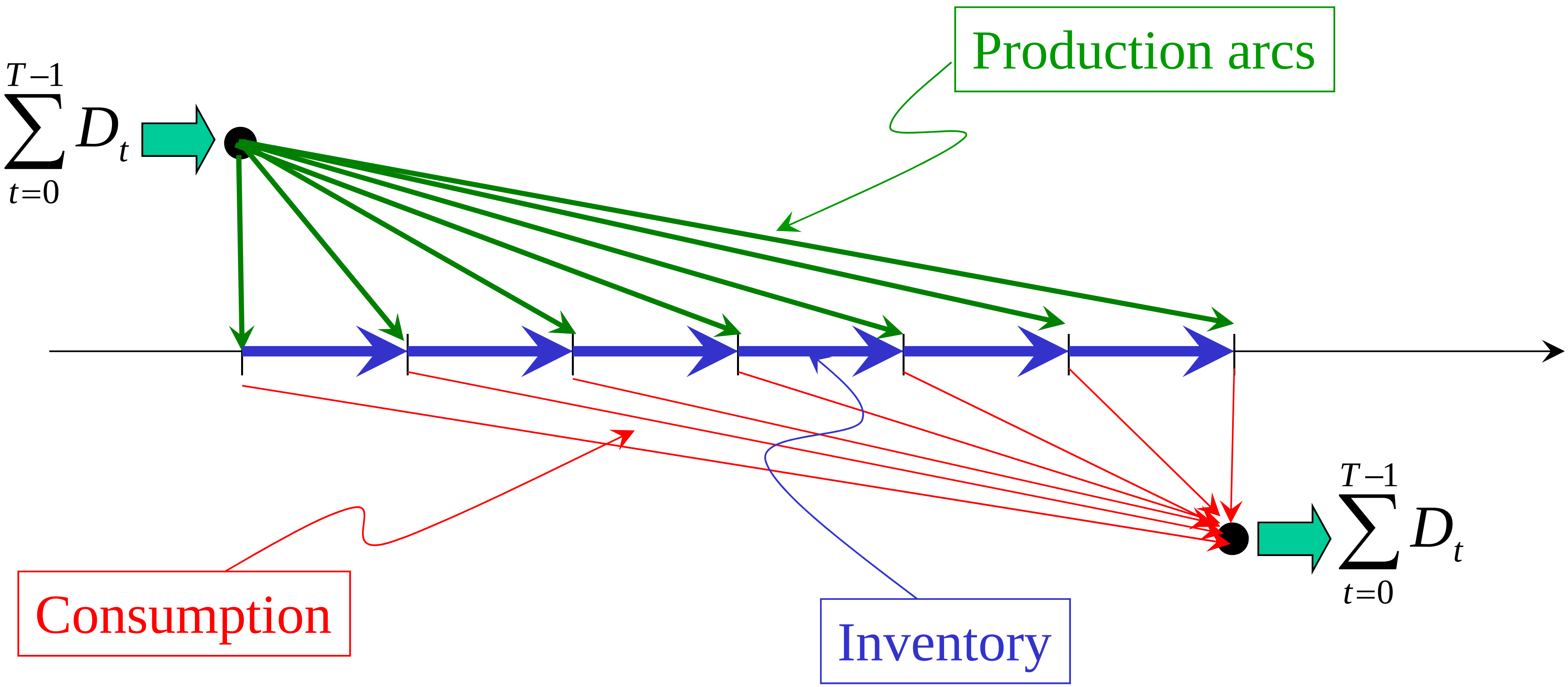
Math programming formulation

- What do we do when the demands are nonstationary?
  - » D_t= the *forecasted* demand for time period t, $0 \leq t < T$. We are going to use a *point* estimate of the demand, which produces a *deterministic* model.
  - » T = the *planning horizon.*
- Objective function:

$$\min_{x_t, y_t} \sum_{t=0}^{T-1} c_t \left(R_t, x_t, y_t \right)$$

Math programming formulation

- The optimization problem can be visualized as a network:



Math programming formulation

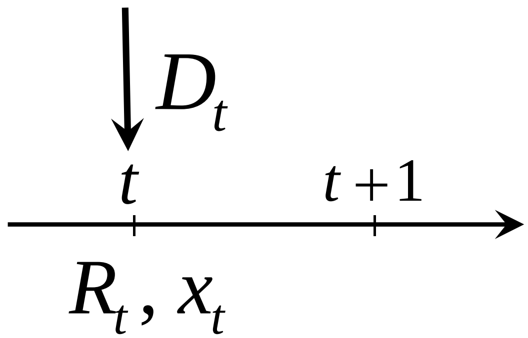
- Integer programming formulation:

Notation:

Activity variables:

$R_t =$ inventory at beginning of period t

$D_t =$ demand during period starting at t



Parameters:

$c^h =$ Unit holding cost per time period

$c^p =$ Unit purchase cost

$K =$ Fixed order cost

Decision variables:

$x_t =$ amount ordered in period t

$$y_t = \begin{cases} 1 & x_t > 0 \\ 0 & x_t = 0 \end{cases}$$

Math programming formulation

Objective function:

$$\min_{x,y} \sum_{t=0}^{T-1} K y_t + c_t^p x_t + c^h \left(R_t + x_t - D_t \right)$$

subject to:

$$R_{t+1} = R_t + x_t - D_t$$

$$x_t \leq M y_t \qquad \text{(M = big number)}$$

$$x_t \geq D_t - R_t$$

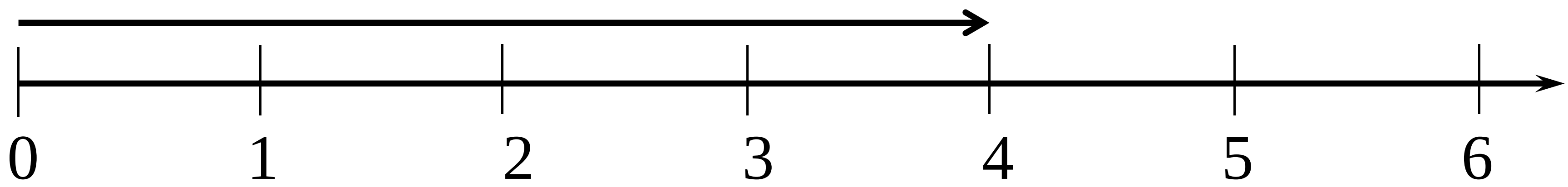
$$x_t \geq 0$$

$$y_t = (0,1)$$

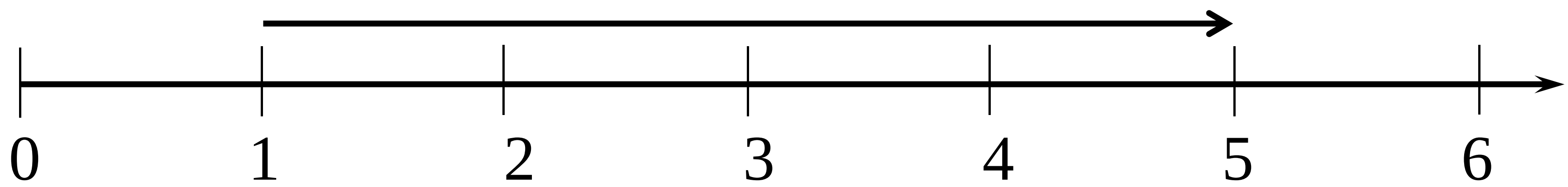
This is an integer programming problem, which can be solved using commercial solvers such as Cplex and Gurobi.

Math programming formulation

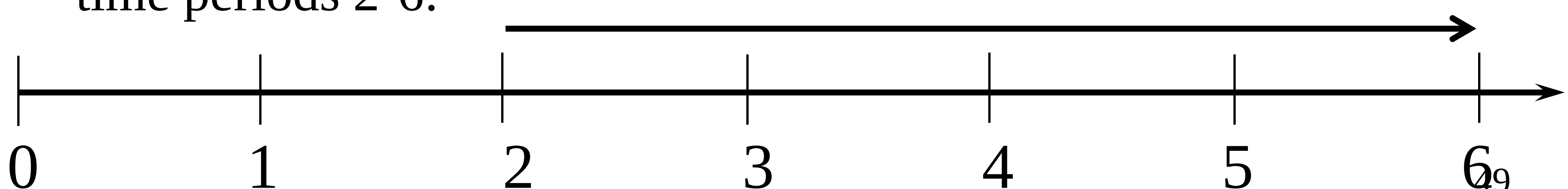
- We implement our math programming formulation as a *rolling horizon procedure*
  - » Optimize over 0-4, implement time 0



  - » Roll to time 1, see new information, solve updated problem for time periods 1-5:



  - » Roll to time 2, see new information, solve updated problem for time periods 2-6:



Math programming formulation

- Rolling horizon procedures
  - » These are *deterministic approximations* of the problem over a *planning horizon* H

Objective function:

$$\min_{x_t y_t} \sum_{t'=t}^{t+H} K y_{tt'} + c_{t'}^{p} x_{tt'} + c^{h}\left(R_{tt'} + x_{tt'} - D_{tt'}\right)$$

where $x_t = \left(x_{tt'}\right)_{t'=t,\ldots,t+H}$, $y_t = \left(y_{tt'}\right)_{t'=t,\ldots,t+H}$

subject to:

$$R_{t,t'+1} = R_{tt'} + x_{tt'} - D_{tt'}$$

$$x_{tt'} \le M y_{tt'} \qquad \text{(M = big number)}$$

$$x_{tt'} \ge D_{tt'} - R_{tt'}$$

$$x_{tt'} \ge 0$$

$$y_{tt'} = (0,1)$$

Heuristics

- Silver-Meal heuristic (Least average cost)

$$\text{Let } C(s) = \text{average cost per unit time if we order over the next } s \text{ time periods}$$

$$= \frac{1}{s}\left(K + c^h \sum_{t=0}^{s-1} tD_t\right)$$

Calculate C(1), C(2), ..., C(s). Stop when C(s + 1) > C(s).

Set T = s. Order enough for the next T time periods.

» ***One of the best known and most widely used heuristics in supply chain management.***

Heuristics

- Least unit cost

Let C(s) = average cost per unit produced if we order over the next s time periods

$$= \frac{\left(K + c^h \sum_{t=0}^{s-1} t D_t\right)}{\left(\sum_{t=0}^{s-1} D_t\right)}$$

Calculate C(1), C(2), ..., C(s). Stop when C(s + 1) > C(s).

Set T = s.

Least unit cost reflects the way managers are actually measured.

No - one is measured in terms of $/day (why not???).

Wagner-Whitin algorithm

- Example: Seasonal TV demand
  - » Parameters:

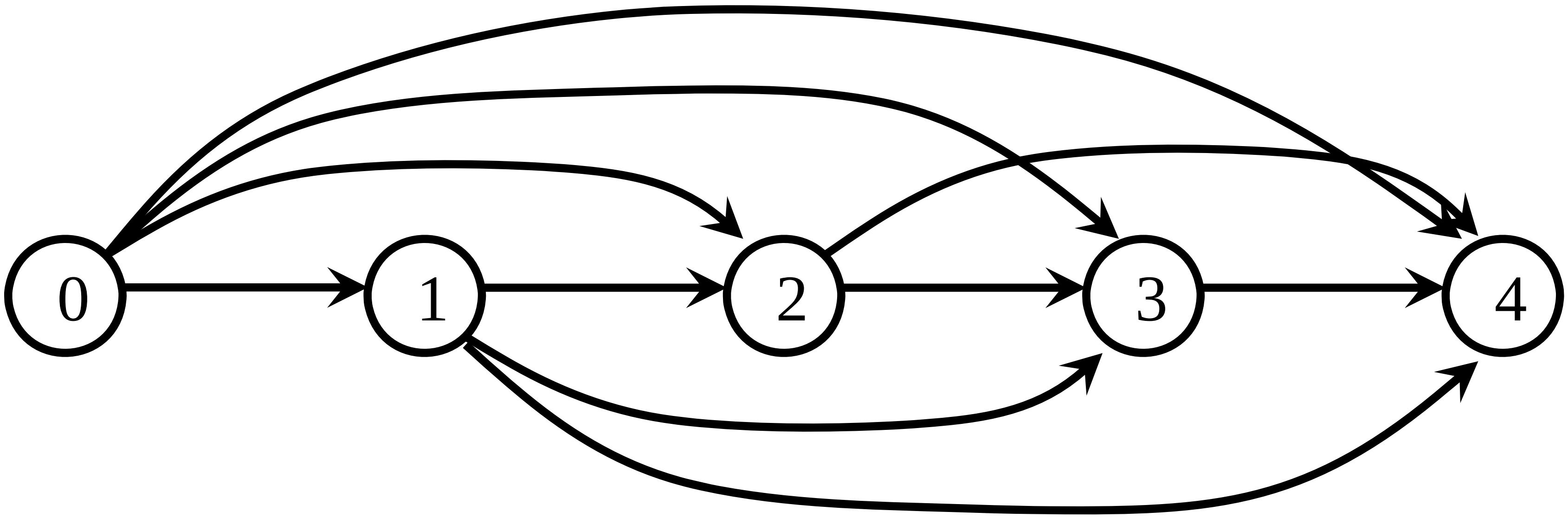
Setup cost	8
Order cost	1
Holding cost	1.3

  - » Demands (in 1000's):

Time	Demand
0	1
1	3
2	5
3	2

Wagner-Whitin algorithm

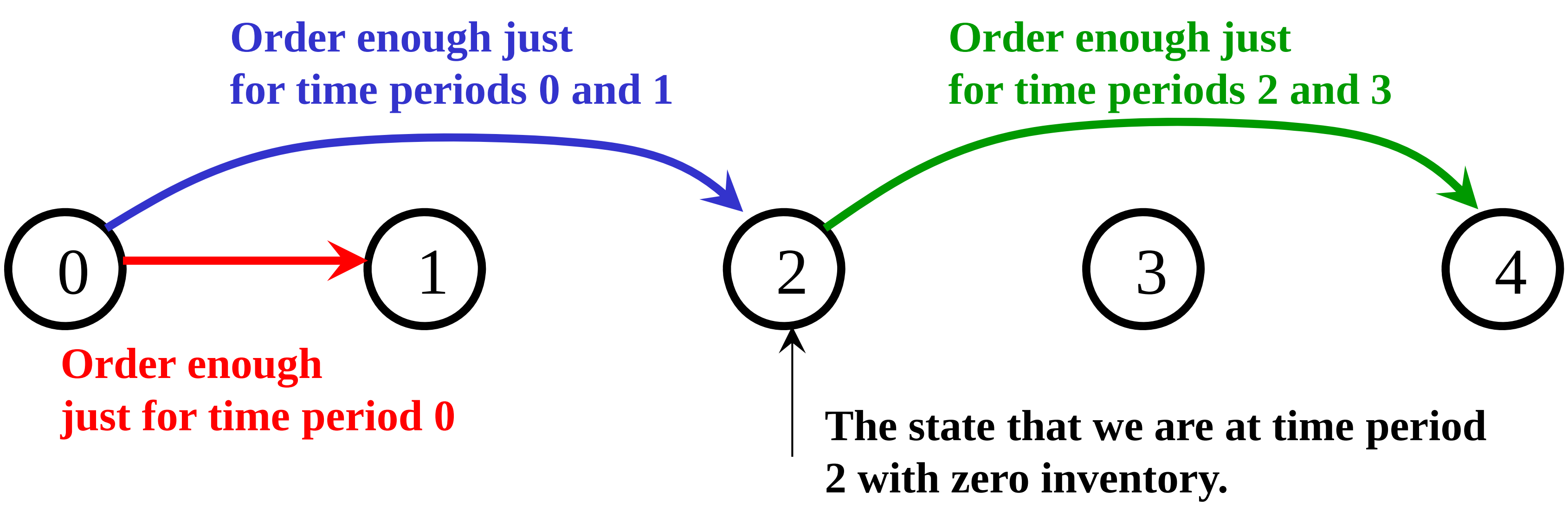
- A network representation:
  - » The set of decisions represents a shortest path problem over a specialized network:



  - » The cost on each arc is the cost of the decision, including order costs and all holding costs.
  - » What does the optimal solutionn look like?

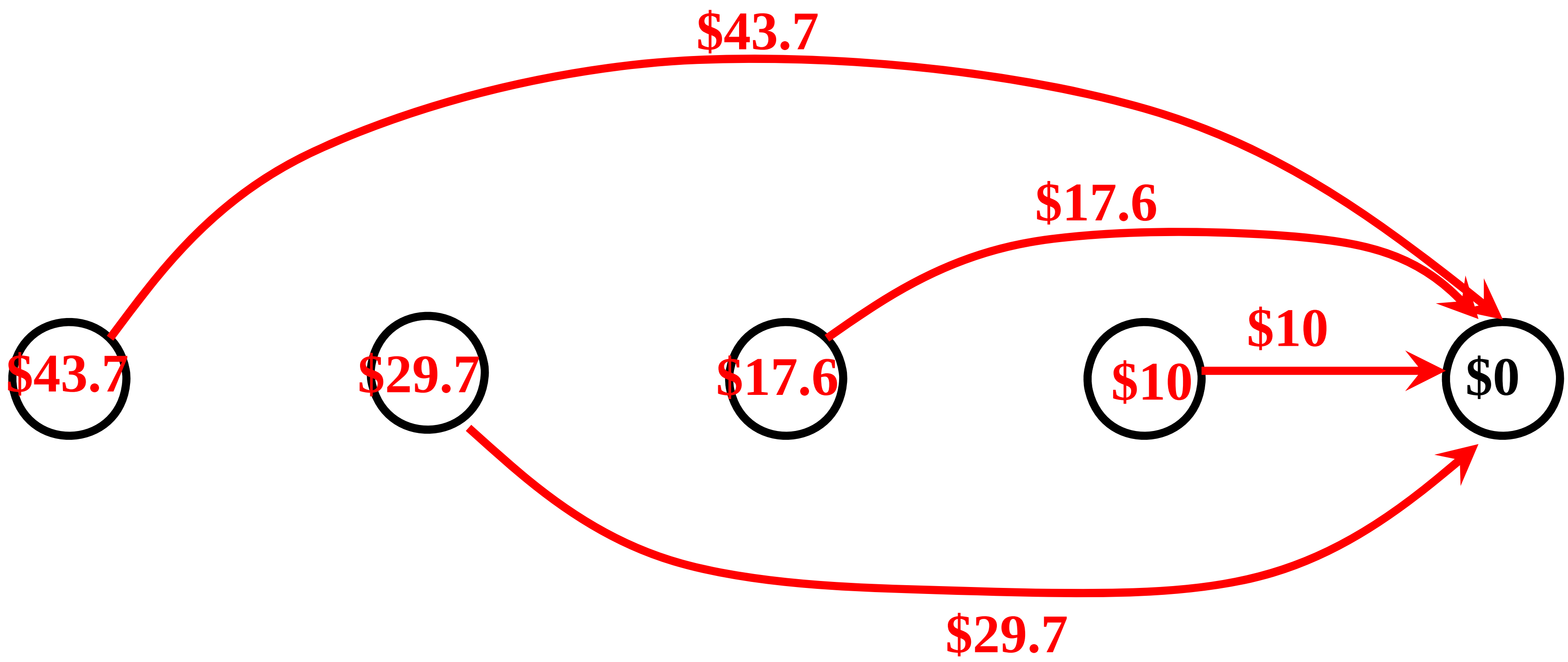
Wagner-Whitin algorithm

- Properties of optimal solution
  - » We only need to make decisions when the inventory is zero.
  - » This means our decision variable is not the quantity, but the number of time periods into the future that we need to cover.



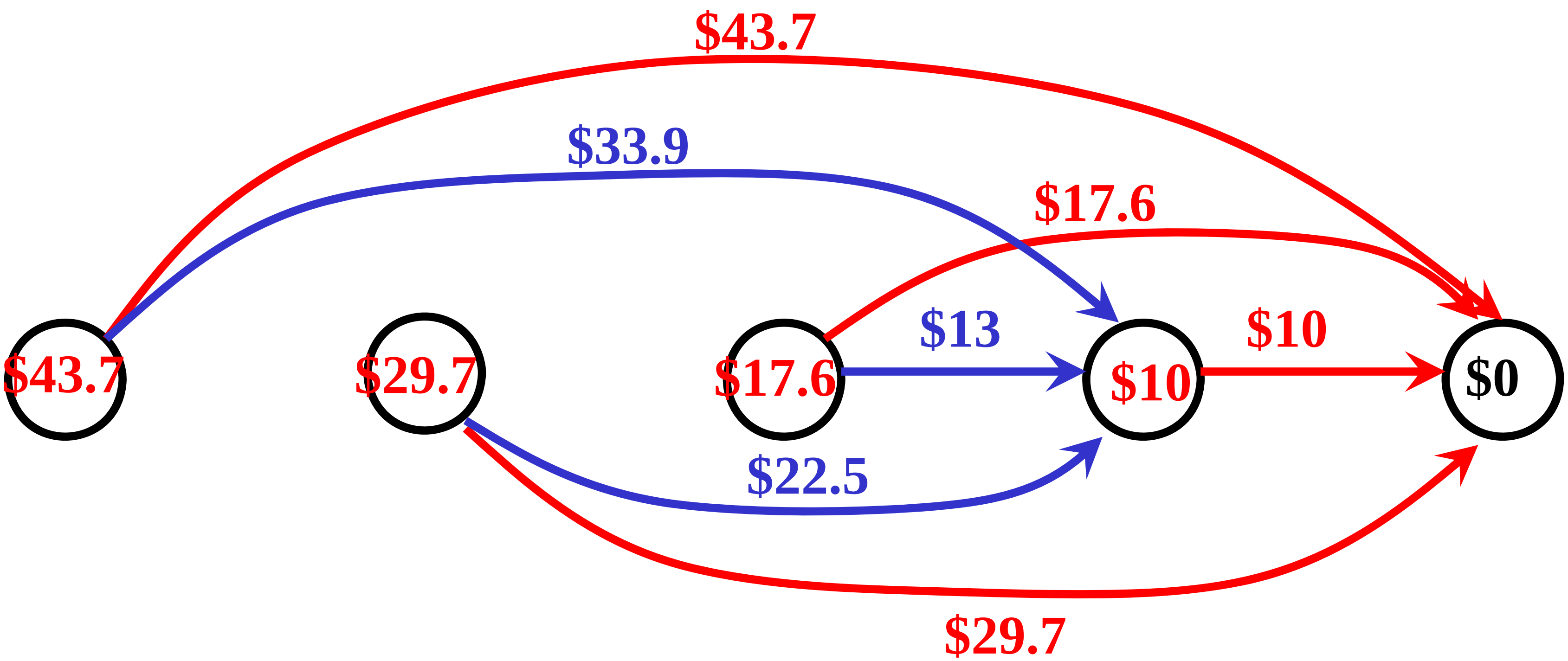
Wagner-Whitin algorithm

- Start with the final node:



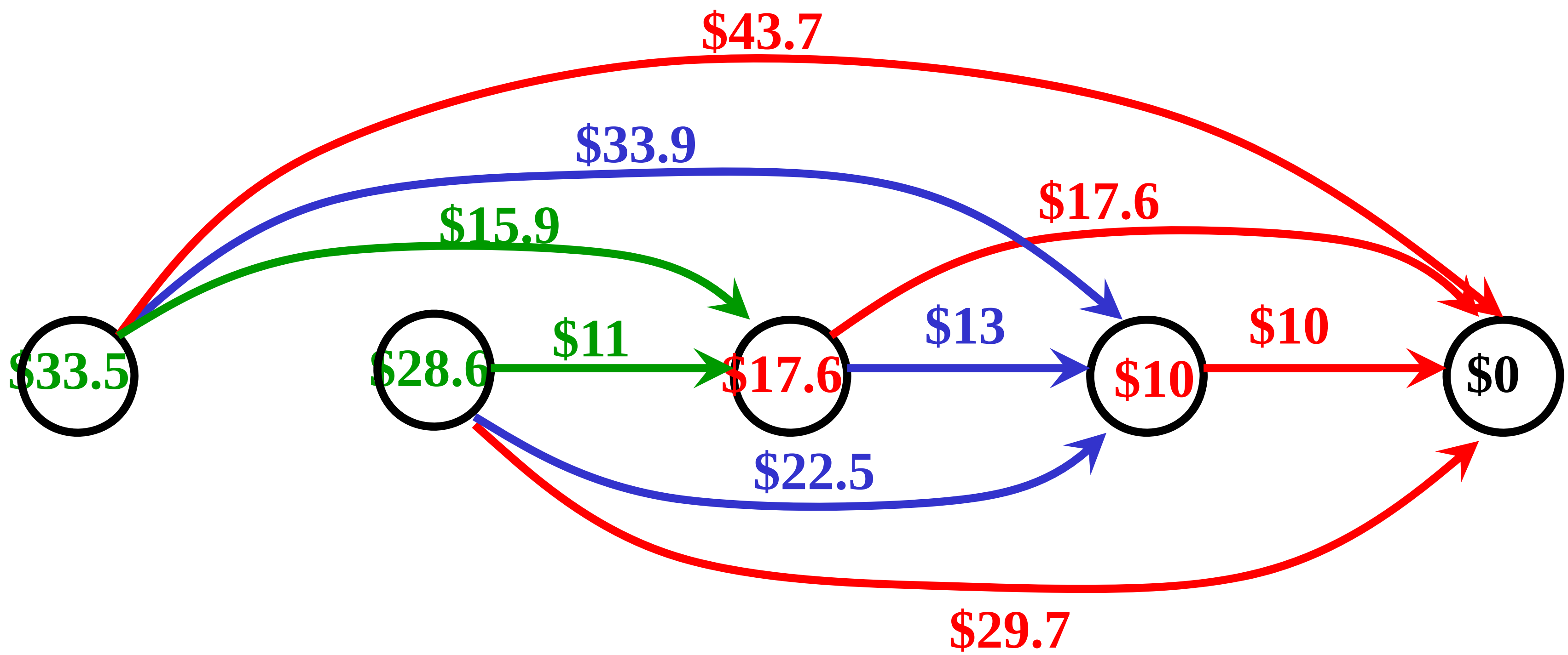
Wagner-Whitin algorithm

- Links into time period 3:



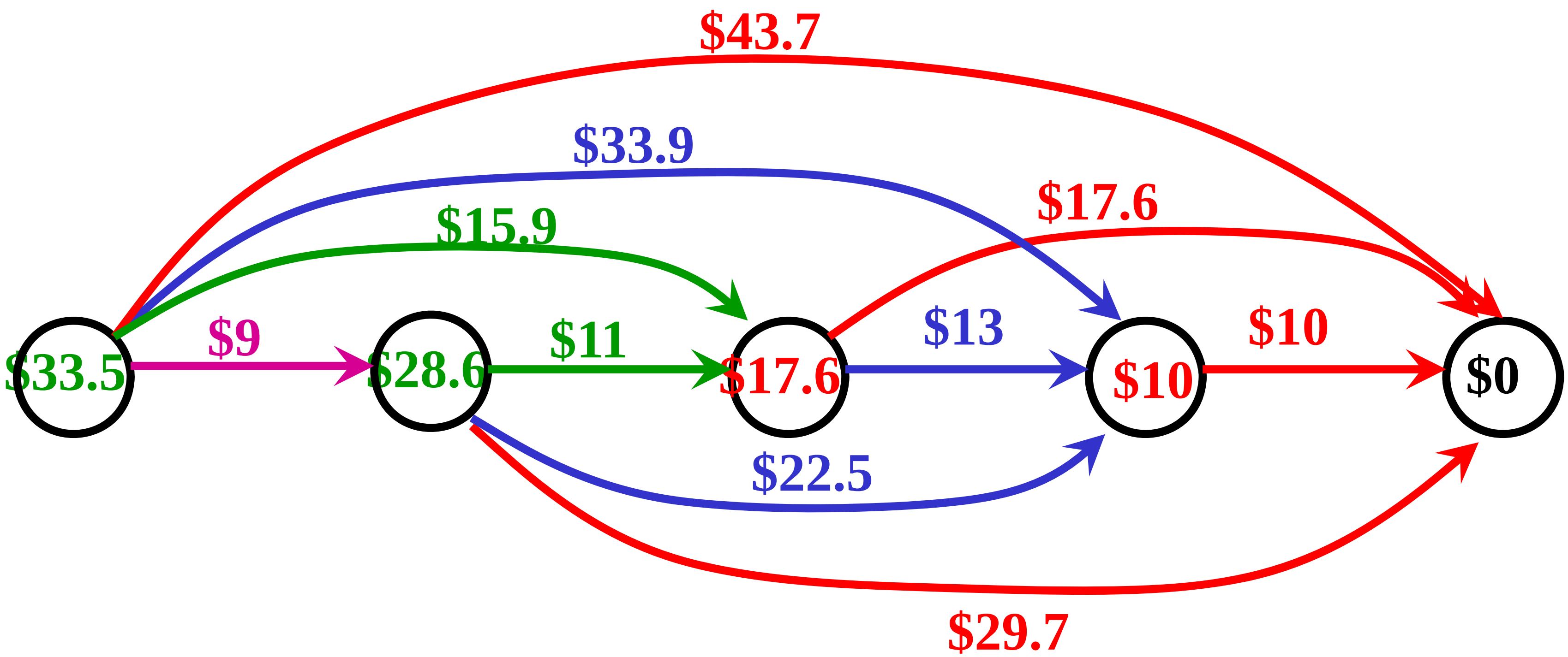
Wagner-Whitin algorithm

- Links into time period 2:



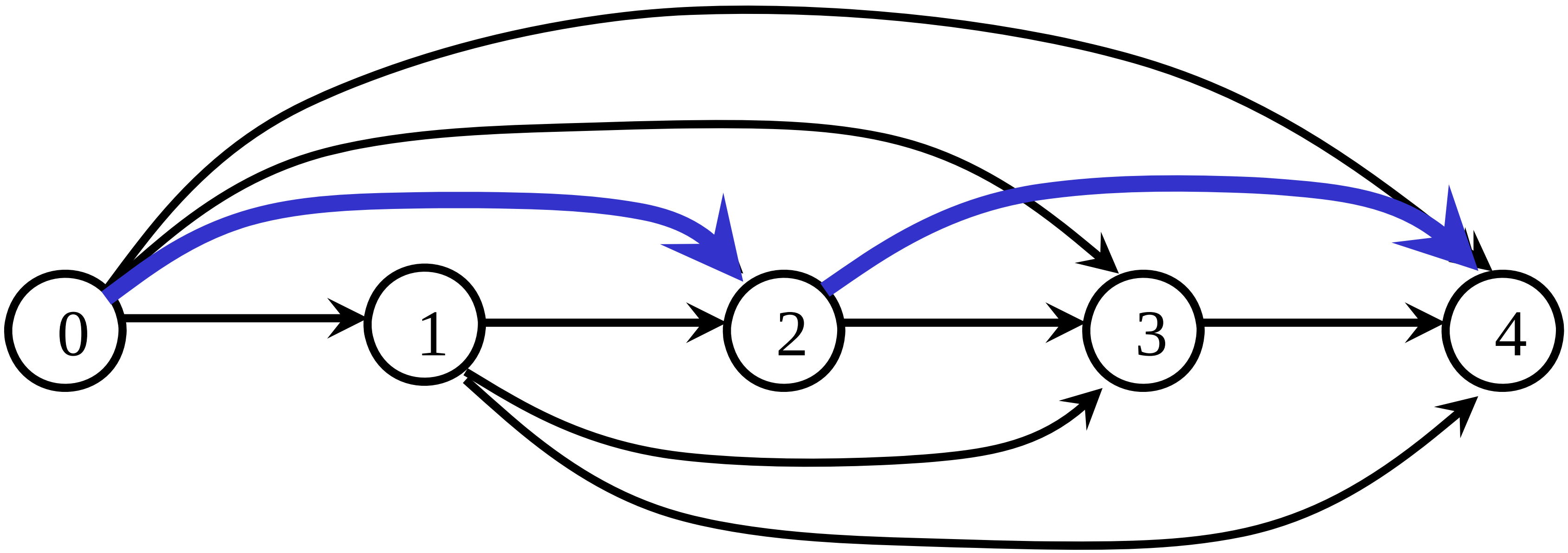
Wagner-Whitin algorithm

- Links into time period 1:



Wagner-Whitin algorithm

- Finally, we walk forward in time:



We use the values computed in the backward pass to walk forward and compute decisions.

Wagner-Whitin algorithm

- Strengths:
  - » Very fast
  - » Handles very general cost functions
    - You can use virtually any shape order cost function.
  - » Handles time-dependent data (e.g. seasonal data, day of week effects or hour of day patterns).
  - » Handles forecasts of the future (which is a form of time-dependency).

Wagner-Whitin algorithm

- Limitations of this model:
  - » Assumes demands are deterministic!!!
    - "Optimal" solution is not really optimal.
    - Drives inventories to zero, which will create stockouts.
    - Have to reoptimize as forecasted demands change.
  - » Limitations:
    - Computationally demanding when you have to solve 100,000 problems (Wal-Mart!).
    - Solutions are not "obviously" better than good heuristics under realistic conditions.
    - Gets complicated if you have multiple items and joint capacity constraints (need to use integer programming formulation)
  - » But:
    - Serves as a useful subproblem in the context of larger applications.
    - Highlights behavior of the problem.

Wagner-Whitin algorithm

- Important generalizations:
  - » Upper bounds on order quantities
    - What if we cannot order more than u_t in time period t?
  - » Upper bounds on production and multiple items:
    - This is the problem that actually arises in practice.
    - Called the “capacitated multi-item lot sizing problem.”
  - » Limit on total production time, in the presence of setup times.
    - The literature on setup times is very sparse.