Chapter 1

The Production Paradigm

Evolution of Production Systems

- Ancient Systems
  - **basic planning, organizations and control**
  - **specialization of labor**
- Feudal Systems
  - **hierachical system (delegation)**
  - **land and labor as production input**
- European System
  - **double entry bookkeeping, cost accounting**
  - **Industrial Revolution: specialization, mass markets, mass production**
- American System
  - **interchangeable parts**
  - **steam power**
  - **assembly lines**

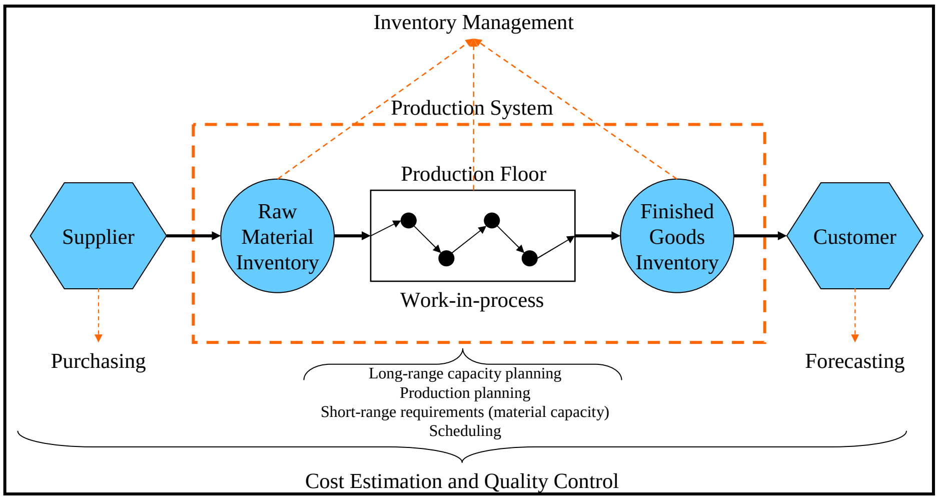
The Competitive Environment

- Status Quo of the American (and European) System(late 80s):
  - **production driven system**
  - **cost efficient production as the main goal**
  - **high quality standardized goods**
  - **Market is taken as given**
- Change towards a market-driven system
  - **more sophisticated consumers**
  - **short product life cycles**
  - **product variety increases**
  - **global competition and heterogeneous markets**

Production Systems

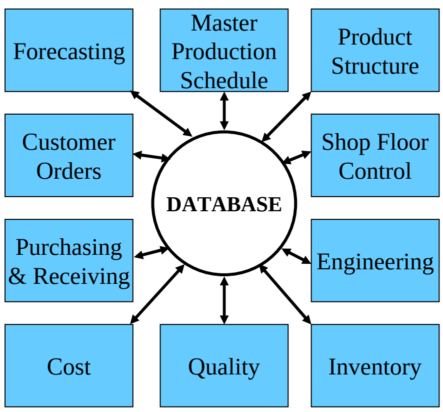
- Input ⟶ Output
- manufacturing firms
- service companies: Universities
- flow process in two parts:
  - **physical material**
  - **information**
- coordination also with suppliers and distributors: supply chain management: recent emphasis on bi-directional information flow

Production Systems



Production Information System

The PPC function integrates material flow using the information system. Integration is achieved through a common database.



Building Blocks

- Objectives:
  - **Quality**
  - **Cost**
  - **Time**

- **These might be seen as the** fundamental objectives **of the firm**
- **induced by these objectives one might observe various** subordinate objectives **at different levels and parts of the company**
  - **more variability, high inventory**
  - **low unit costs, low inventory**
  - **high throughput, less variability**
  - **short cycle times, high inventory**
- Important to understand effects of individual incentives!

Building Blocks

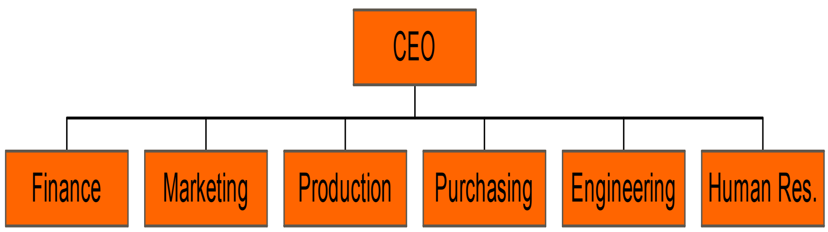
- **Physical Arrangement**
  - production volume and
  - product variety
  - determine **layout**
    - job shop (low-volume, high customized)
      - process or functional layout
    - flow shop (high-volume)
      - product layout

Building Blocks

- **Organizational Arrangements**
  - **Functional Structure:** input oriented
  - **Divisional Structure:** output oriented (projects, services, programs, locations) strategic business units
  - **Matrix Structure:** one person-two bosses (input & output oriented)

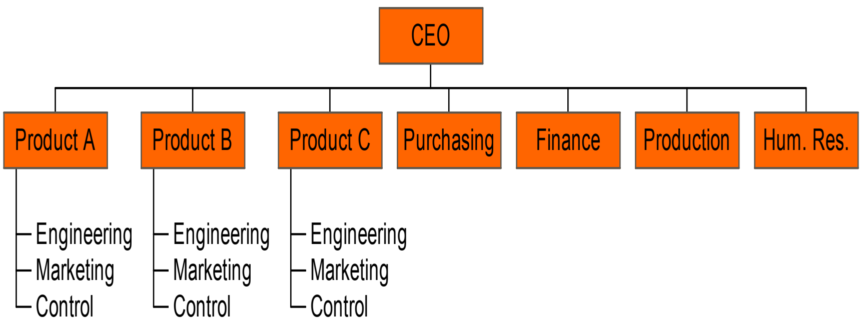
Organizational Arrangements

Functional Structure

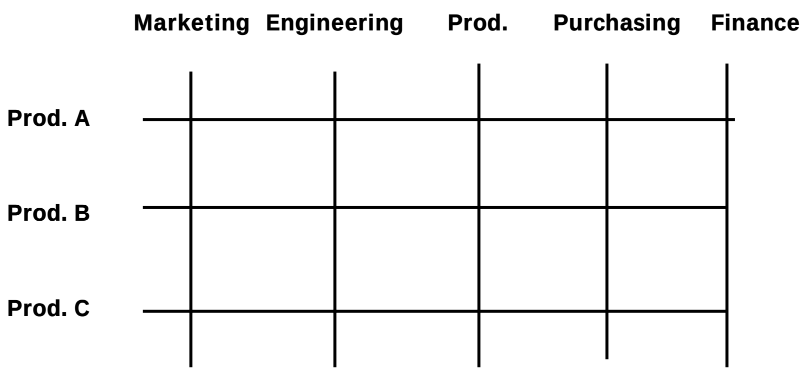


Organizational Arrangements

Divisional Structure



Organizational Arrangements: Matrix

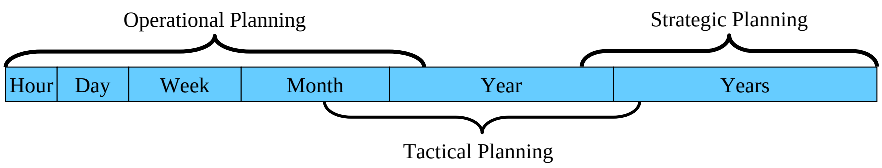


Production Planning and Control (PPC)

- Intergrated-material-flow-based information system
- based on a feedback loop (control theory)
- management of deviations
- art of selecting the appropriate mix of management technologies
- impact of organizational structure, life-cycle effects

Building Blocks

- **Planning horizons**



Building Blocks

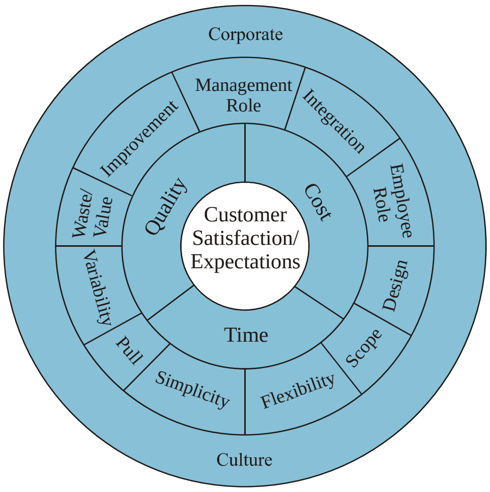
⌘ **Types of Decisions**

	Long (strategic) top management	**Intermediate (tactical) middle management**	**Short (operational) operational management**
Time	three to ten years	six months to three years	one week to six months
Unit	dollars; hours	dollars; hours; product line; product family	individual products; product family
Inputs	aggregate forecast; plant capacity	intermediate forecast; capacity and production levels taken from long range plan	short range forecast; work force levels, processes; inventory levels
Decisions	capacity; product; supplier needs; quality policy	work force levels; processes; production rates; inventory levels; contracts with suppliers; quality level; quality costs	allocation of jobs to machines; overtime; undertime; subcontracting; delivery dates for suppliers; product quality

Chapter 2

Market Driven Systems

Market driven systems
The Wheel of Competitiveness



The Wheel of competitiveness

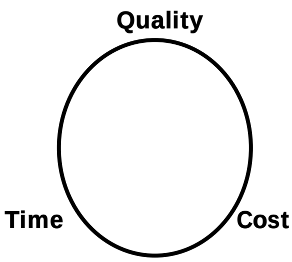
- **Hub: the customer**
  - individual rather than average customer
  - fast changing expectations
  - little loyalty
  - 'internal customers': any operation is the customer of the previous operation

The Wheel of competitiveness

- **The Delivery Cycle:**

Quality ⟶ **Time** ⟶ **Cost**

or

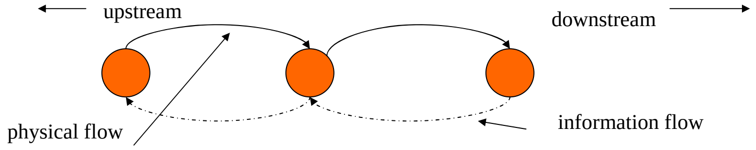


The Wheel of Competitiveness

- **The Support Circle**
  - Scope (Supplier - Producer - Relationship)
  - Integration
    - looking at the system rather than a component
    - product and process design
  - Flexibility
    - volume
    - process (setups)
  - Design
    - function, life, form and effective manufacture

The Wheel of Competitiveness

- **The Support Cycle:**
  - Simplicity (KISS)
  - Variability
    - deterministic manufacturing
    - Factory Physics (Hopp/Spearman)
  - Pull
    - physical flow
    - information flow
    - the essence of pull production is to do things upstream only when requested downstream



The Wheel of Competitiveness

- **The Support Cycle**
  - Waste/Value
    - "doing it right the first time"
    - value-adding activities
    - cost adding activities
  - Improvement
    - Integrated and Continues Improvement
    - Kaizen, ...
  - Management role
    - change process
    - commitment
    - participation
    - goals

The Wheel of Competitiveness

- **The Support Cycle**
  - Employee role
    - involvement
    - development

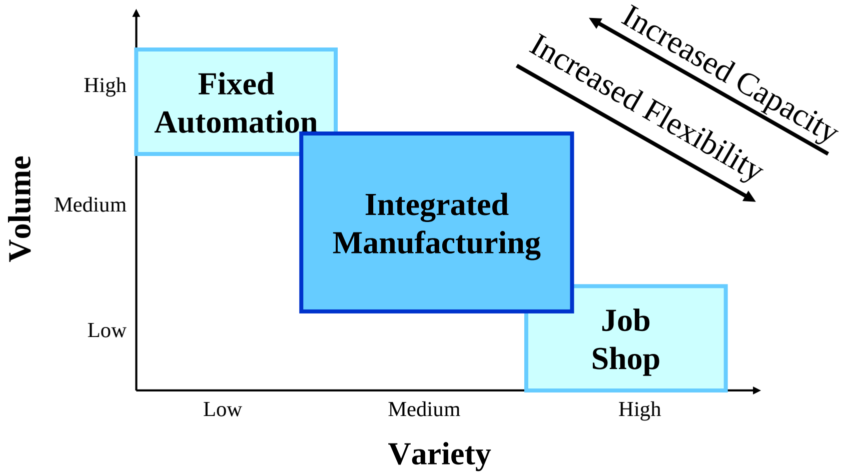
- **The impact circle**
  - Efficiency: make things right
    - local
    - ration of output to input
  - Effectiveness: requirements of the total system

Implementation

- **Integrated Production Systems**
  - best applied in the medium-variety, medium-volume range
  - information integration is key aspect

- **3 leading approaches**
  - Cellular Manufacturing Systems (CMS)
  - Flexible Manufacturing Systems (FMS)
  - Computer Integrated Manufacturing Systems (CIM)

Integrated Production Systems



Integrated Production Systems

- **Cellular Manufacturing Systems**
  - manned or unmanned cells
  - produce a family of parts that have similar processes
  - group technology (see Basic Course: OMA)
  - organized in a u-shaped layout in which multifunctional workers perform the required operations

Integrated Production Systems

- **Flexible Manufacturing Systems**
  - integration of
    - manufacturing or assembly processes
    - automated material flow systems
    - computer communication
    - control
  - computer control system does:
    - production control
    - scheduling
    - flow control
    - machine control
  - reaction to real time status data
  - automotive and electronics industry

Integrated Production Systems

- **Computer Integrated Manufacturing (CIM)**
  - broader scope than CMS
  - use information technology to coordinate business functions with product development, design and manufacturing
  - ‚bridges' between FMS islands

Market driven systems

- **Integration Process**
  - teamwork
  - concurrent engineering
    - life cycle engineering
    - product and process design are considered together
    - cross functional teams
  - TQM
  - World class manufacturing
  - Lean production(Toyota, production floor focus)
  - Agile manufacturing(enterprise view)

Chapter 3

Problem Solving

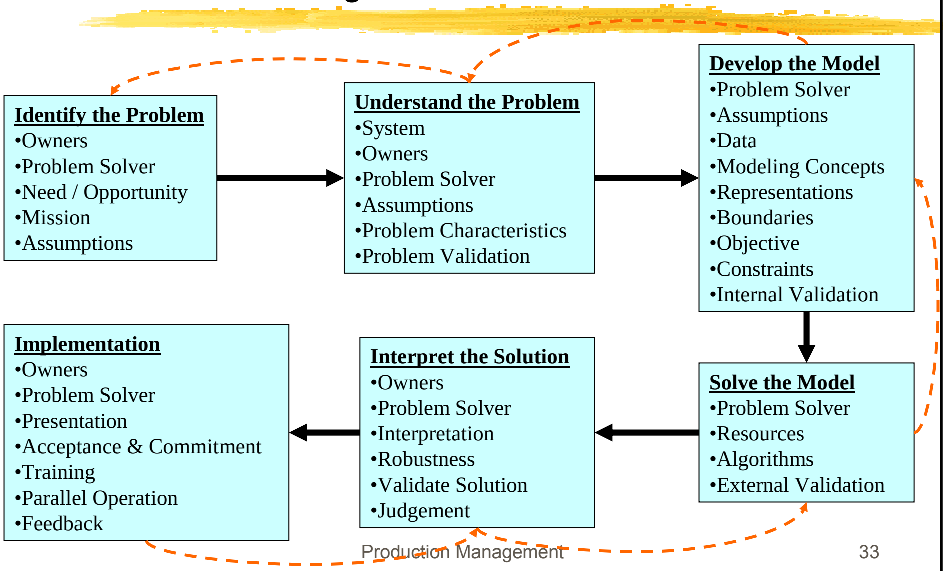
Problem Solving

- **Current state** ⟶ **goal state**
- **impact:** should be worth the resource
- ability to measure the gap
- ability to close the gap
- solve or dissolve

Problem Solving

- **Problem Identifikation**
  - Symptoms
  - Problem mission
  - mission will be translated into goals and objectives
  - problem owners: people who must live with the solution
  - Assumptions
  - Initial Problem Statement

Problem Solving



Problem Solving

- **Understand the problem**
  - The systems perspective
    - analysis
    - synthesis
  - Goals
  - Problem Characteristics
    - one-time - recurrent
    - level of detail
  - Validate Understanding

Problem Solving

- **Develop a model**
  - Model representation
    - iconic
    - analog
    - symbolic
  - Data
  - Modeling concepts
    - Boundaries
    - Objectives
    - Constraints
  - Relationships
  - Assumptions and Involvement
  - Internal validation

Problem Solving

- **Solve the Model**
  - External validation
  - Simplification
  - Solution Strategy
    - Exact
    - Heuristic
    - Simulation
- **Interpret the solution**
  - robustness
  - plausibility
- **Implementation**

Example: MaTell – Identifiy

- **MaTell produces telephones: desk phones, wall phones, answering machines**
- **All 3 products are made at a single plant**
- **Customers cannot buy the products because they are unavailable**

- **Is there a problem?**
- **What is the problem mission?**
- **Who are the owners of this problem?**
- **Assumptions?**

- **Initial problem statement:**
  - Current state: Some customers who want our product cannot get them.
  - Goal state: Deliver a product to all of our customers who want one.
  - Problem: How can we provide products to all out customers?

Example: MaTell - Understand

- **variety of ways to provide more products**
  - build a new plant
  - expand the existing plant
  - subcontracting
  - ...
- **actual production system**
  - fabrication department - assembly department
  - 15000 wall phones (W), 17000 desk phones (D), 5000 answering machines (A) per weak
  - plant works a three eight-hour shifts a day, seven days a week
  - fabrication: 135 hours per week
  - assembling: 163 hours per week
- **new problem owner: production department**
- **2 strategies:**
  - using capacity more effectively
  - reducing the time a product spends in assembly

Example: MaTell - Develop

- **data available: time it takes to make each product in the fabrication and assembly department**
  - 1000 desk phones: 2.5 hours fabrication, 3 hours assembly
  - 1000 wall phones: 4 hours fabrication, 3 hours assembly
  - 1000 answering machines: 6 hours fabrication, 14 hours assembly
- **objective:**

$$W+D+A$$

- **total fabrication time:**

$$4\,W + 2.5\,D + 6\,A$$

- **total assembly time:**

$$3\,W + 3\,D + 14\,A$$

- **marketing department: at most 30000 desk phones, 30000 wall phones and 12000 answering machines can be sold per week.**
- **assumptions:**
  - Demand will continue at the same levels or higher for some time
  - The number of products made is a good measure for increasing the throughput.
  - There is a linear relationships between products and fabrication (assembly) time.
  - Data are accurate.

Example: MaTell – Solve / Interprete

- **Solve using Excel spreadsheet / Solver**
- **Is the new mix more or less profitable ?**
- **margins: $2.20 (D), $2.00 (W), $7.00 (A)**
- **alternative objective:**

$$2.2\,D + 2\,W + 7\,A$$

- **add lower bounds: 10 (D), 10 (W), 4 (A)**

Example: MaTell – Implementation

- present the solution
  Though the spreadsheet was not used to get the solution, it would be a good way to introduce the LP solution
- acceptance relatively easy (owners were involved)
- commitment may be more difficult, but only few resources needed (LP package, training for the planner)
- check the system from time to time (conditions may change)

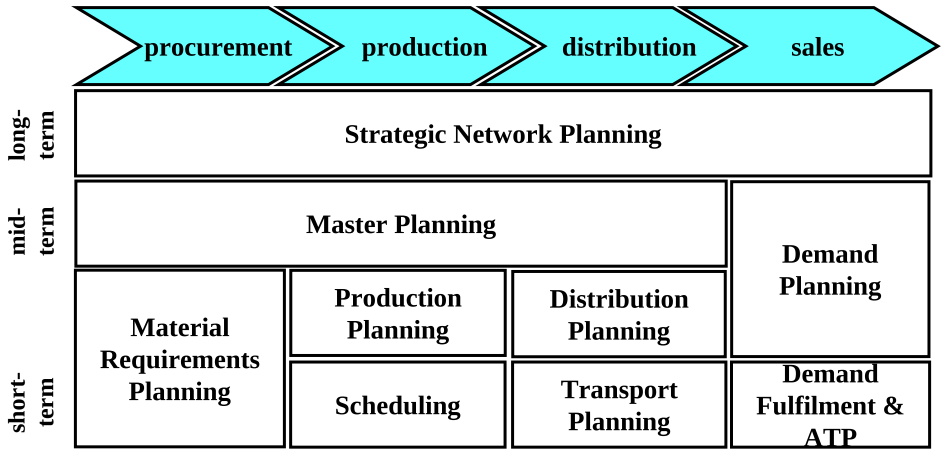
Problem Solving

- Work to do:
- Examples: 3.12 abcd, 3.19 ab, 3.30abc, 3.36abc, 3.41 abc, 3.46

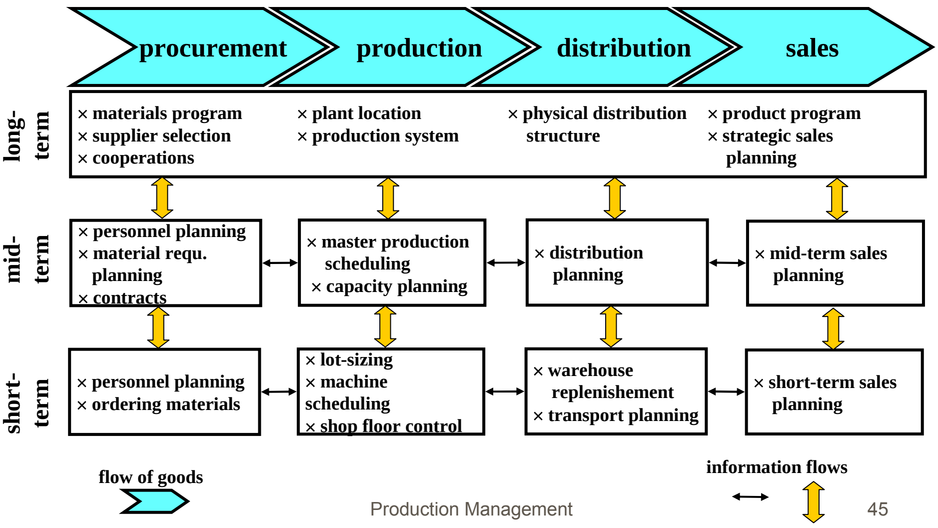
Chapter 5

Aggregate Planning

Supply Chain Planning Matrix



Supply Chain Planning Matrix



Aggregate Planning

- **Example:**
  - one product (plastic case)
  - two injection molding machines, 550 parts/hour
  - one worker, 55 parts/hour
  - steady sales 80.000 cases/month
  - 4 weeks/month, 5 days/week, 8h/day
  - how many workers?
- **in real life constant demand is rare**
  - change demand
  - produce a constant rate anyway
  - vary production

Aggregate Planning

- **Influencing demand**
  - do not satisfy demand
  - shift demand from peak periods to nonpeak periods
  - produce several products with peak demand in different period
- **Planning Production**
  - Production plan: how much and when to make each product
  - rolling planning horizon
  - long range plan
  - intermediate-range plan
    - units of measurements are aggregates
    - product family
    - plant department
    - changes in workforce, additional machines, subcontracting, overtime,...
  - Short-term plan

Aggregate Planning

- **Aspects of Aggregate Planning**
  - Capacity: how much a production system can make
  - Aggregate Units: products, workers,...
  - Costs
    - production costs (economic costs!)
    - inventory costs(holding and shortage)
    - capacity change costs

Aggregate Planning

- **Spreadsheet Methods**
- **Zero Inventory Plan**
  - Precision Transfer, Inc. Produces more than 300 different precision gears (the aggregation unit is a gear!).
  - Last year (=260 working days) Precision made 41.383 gears of various kinds with an average of 40 workers.
  - 41.383 gears per year
  - *40 x 260 worker-days/year = 3,98 -> 4 gears/ worker-day*

  - Aggregate demand forecast for precision gear:

Month	January	February	March	April	May	June	Total
Demand	2760	3320	3970	3540	3180	2900	19.670

Aggregate Planning

- holding costs: $5 per gear per month
- backlog costs: $15 per gear per month
- hiring costs: $450 per worker
- lay-off costs: $600 per worker
- wages: $15 per hour (all workers are paid for 8 hours per day)
- there are currently 35 workers at Precision
- currently no inventory

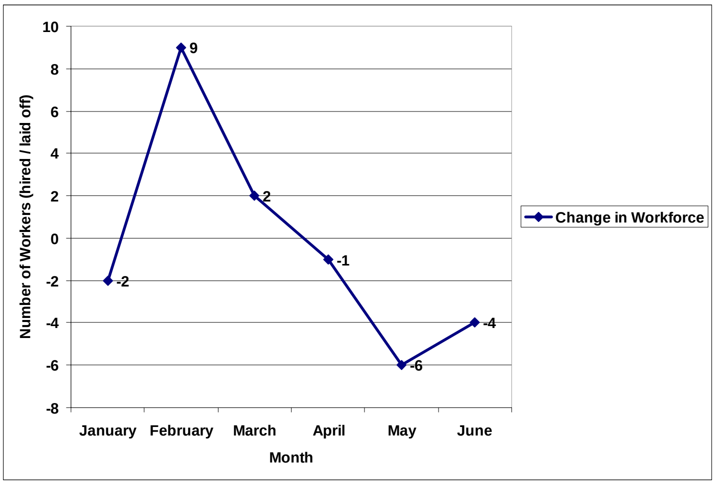
- Production plan?

Aggregate Planning

⌘ **Zero Inventory Plan**

- produce exactly amount needed per period
- adapt workforce

Aggregate Planning



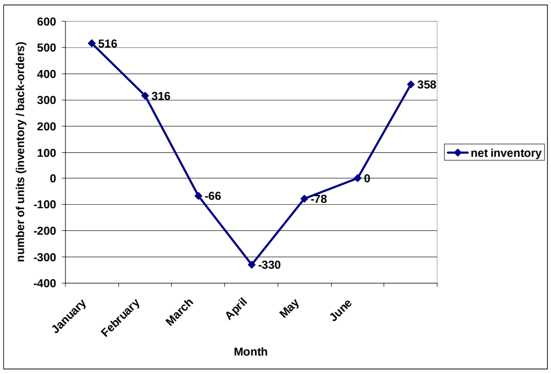
Aggregate Planning

- **Level Work Force Plan**
  - backorders allowed
  - constant numbers of workers
  - demand over the planning horizon
  - gears a worker can produce over the horizon

  - 19670/(4x129)=38,12 -> 39 workers are always needed

Aggregate Planning

- Inventory: January: 3276 - 2760 = 516
- February: 516 + 3120 – 3320
- March: 316 + 3588 – 3670 = -66! -Backorders: 66 x $15 = $990



Aggregate Planning

- **no backorders are allowed**
  - *workers= cumulative demand/(cumulative days x units/workers/day)*

  - January: 2760/(21 x 4) = 32,86 -> 33 workers
  - February: (2760+3320)/[(21+20) x 4] = 37,07 -> 38 workers.
  - March: 10.050/(64 x 4) =>40 workers
  - April: 13.590/(85 x 4) => 40 workers
  - May: 16.770/(107 x 4) => 40 workers
  - June: 19670/(129 x 4) => 39 workers

Aggregate Planning

- **Example Mixed Plan**
  - The number of workers used is an educated guess based on the zero inventory and level work force plans!

Spreadsheet Methods Summary

	Zero-Inv.	Level/BO	Level/No BO	Mixed
Hiring cost	4950	1800	2250	3150
Lay-off cost	7800	0	0	4200
Labor cost	59856	603720	619200	593520
Holding cost	0	4160	6350	3890
BO cost	0	7110	0	990
Total cost	611310	616790	627800	605180
Workers	33	39	40	35

Aggregate Planning

- **Linear Programming Approaches to Aggregate Planning**

  Parameters:

  T... Planning horizon length

  t ... Index of periods, t=1,2,..., T

  D_t …forcasted number of units demanded in period t

  n_t …number of units that can be made by one worker in period t

  C_t^P …cost to produce one unit in period t

  C_t^W …cost of one worker in period t

Aggregate Planning

C_t^H …cost to hire one worker in period t

C_t^L …cost to lay off one worker in period t

C_t^I …cost to hold one unit in inventory in period t

C_t^B …cost to backorder one unit in period t

Aggregate Planning

Decision Variables:

P_t …number of units produced in period t

W_t …number of workers available in period t

H_t …number of workers hired in period t

L_t …number of workers laid off in period t

I_t …number of units held in inventory in period t

B_t …number of units backordered in period t

Aggregate Planning

Constraints: work, Capacity, force, material

$P_t \le n_t W_t \qquad t = 1, 2, \ldots, T$

$W_t = W_{t-1} + H_t - L_t \qquad t = 1, 2, \ldots, T$

net inventory this period = net inventory last period + production this period - demand this period

$I_t - B_t = I_{t-1} - B_{t-1} + P_t - D_t$

Costs

$$\sum_{t=1}^{T} (C_t^P P_t + C_t^W W_t + C_t^H H_t + C_t^L L_t + C_t^I I_t + C_t^B B_t)$$

Aggregate Planning

- **Example: Precision Transfer**
  - Planning horizon: 6 months T= 6
  - Costs do not vary over time $C_t^P = 0$
  - d_t : days in month t
  - $C_t^W = \$120 d_t$
  - $C_t^H = \$450$
  - $C_t^L = \$600$
  - $C_t^I = \$5$

  - We assume that no backorders are allowed!
  - no production costs and no backorder costs are included!
  - Demand

January	February	March	April	May	June	Total
2760	3320	3970	3540	3180	2900	19.670

Linear Program Model for Precision Transfer

Minimize

$$\begin{aligned}&2520W_1 + 2400W_2 + 2760W_3 + 2520W_4 + 2640W_5 + 2640W_6\\&+ 450(H_1 + H_2 + H_3 + H_4 + H_5 + H_6)\\&+ 600(L_1 + L_2 + L_3 + L_4 + L_5 + L_6)\\&+ 5(I_1 + I_2 + I_3 + I_4 + I_5 + I_6)\end{aligned}$$

subject to
(Production-capacity constraints)

$$P_1 \le 84W_1, \quad P_2 \le 80W_2, \quad P_3 \le 92W_3, \quad P_4 \le 84W_4, \quad P_5 \le 88W_5, \quad P_6 \le 88W_6,$$

(Work-force constraints)

$$\begin{aligned}&W_1 = 35 + H_1 - L_1, \quad W_2 = W_1 + H_2 - L_2, \quad W_3 = W_2 + H_3 - L_3,\\&W_4 = W_3 + H_4 - L_4, \quad W_5 = W_4 + H_5 - L_5, \quad W_6 = W_5 + H_6 - L_6,\end{aligned}$$

(Inventory-balance constraints)

$$\begin{aligned}&I_1 = P_1 - 2760, \quad I_2 = I_1 + P_2 - 3320, \quad I_3 = I_2 + P_3 - 3970,\\&I_4 = I_3 + P_4 - 3540, \quad I_5 = I_4 + P_5 - 3180, \quad I_6 = I_5 + P_6 - 2900\,(=0),\end{aligned}$$

(Non-negativity constraints)

$$\begin{aligned}&P_1, P_2, P_3, P_4, P_5, P_6, W_1, W_2, W_3, W_4, W_5, W_6, H_1, H_2, H_3, H_4, H_5, H_6,\\&L_1, L_2, L_3, L_4, L_5, L_6, I_1, I_2, I_3, I_4, I_5, I_6 \ge 0\end{aligned}$$

Aggregate Planning

- **LP solution (total cost = $600 191,60)**

	Production	Inventory	Hired	Laid off	Workers
January	2940,00	180,00	0,00	0,00	35,00
February	3232,86	92,86	5,41	0,00	40,41
March	3877,14	0,00	1,73	0,00	42,14
April	3540,00	0,00	0,00	0,00	42,14
May	3180,00	0,00	0,00	6,01	36,14
June	2900,00	0,00	0,00	3,18	32,95

Aggregate Planning

Rounding LP solution

	January	February	March	April	May	June	Total
Days	21	20	23	21	22	22	129
Units/Worker	84	80	92	84	88	88	516
Demand	2760	3320	3970	3540	3180	2900	19670
Workers	**35**	**41**	**42**	**42**	**36**	**33**	229
Capacity	2940	3280	3864	3528	3168	2904	19684
Capacity - Demand	180	-40	-106	-12	-12	4	14
Cumulative Difference	180	140	34	22	10	14	400
Produced	2930	3280	3864	3528	3168	2900	19670
Net inventory	170	130	24	12	0	0	336
Hired	0	6	1	0	0	0	7
Laid Off	0	0	0	0	6	3	9
Costs	89050	101750	116490	105900	98640	88920	**600750**

Aggregate Planning

- **Practical Issues**
  - 100.000 variables and 40.000 constraints
  - LP/MIP Solvers: CPLEX, XPRESS-MP, ...
- **Extensions**
  - Bounds

$$I_t \le I_t^U$$

$$I_t^L \le I_t \le I_t^U$$

$$L_t \le 0.05 W_t$$

Training

$$W_t = W_{t-1} + H_{t-1} - L_t$$

Aggregate Planning

- **Transportation Models**
  - supply points: periods, initial inventory
  - demand points: periods, excess demand, final inventory

$n_t W_t$ = capacity during period t

D_t = forecasted number of units demanded in period t

C_t^P = the cost to produce one unit in period t

C_t^I = the cost to hold one unit in inventory in period t

Aggregate Planning

t	1	2	3
capacity $n_t W_t$	350	300	350
demand	200	300	400
production costs	10	11	12
holding costs	2	2	2

initial inventory: 50
final inventory: 75

Aggregate Planning

	1	2	3	Ending inventory	Excess capacity	Available capacity
Beginning inventory	0 **50**	2	4	6	0	50
Period 1	10 **150**	12	14 **50**	16 **75**	0 **75**	350
Period 2	-	11 **300**	13	15	0	300
Period 3	-	-	12 **350**	14	0	350
Demand	200	300	400	75	75	1050

Aggregate Planning

- Extension:

t	1	2	3
capacity $n_t W_t$	350	350	300
demand	400	300	400
production costs	10	11	12
holding costs	2	2	2

  - **overtime**: overtime capacity is **90, 90** and **75** in period 1, 2 and 3;
  - overtime costs are **$16, $18** and **$ 20** for the three periods respectively;
  - **backorders**:units can be backordered at a cost of **$5** per unit-month; production in period 2 can be used to satisfy demand in period 1

Aggregate Planning

		1	2	3	Ending inventory	Excess capacity	Available capacity
Beginning inventory		(0)	(2) **25**	(4) **25**	(6)	(0)	50
Period 1	Regular time	(10) **350**	(12)	(14)	(16)	(0)	350
	Overtime	(16) **50**	(18)	(20)	(22)	(0) **40**	90
Period 2	Regular time	(16)	(11) **275**	(13)	(15) **75**	(0)	350
	Overtime	(23)	(18)	(20)	(22)	(0) **90**	90
Period 3	Regular time	(22)	(17)	(12) **300**	(14)	(0)	300
	Overtime	(30)	(25)	(20) **75**	(22)	(0)	75
Demand		400	300	400	75	130	1305

Aggregate Planning

- **Disaggregating Plans**
  - aggregate units are not actually produced, so the plan should consider individual products
  - disaggregation
  - master production schedule
- **Questions:**
  - In which order should individual products be produced?
    - e.g.: shortest run-out time $R_i = I_i / D_i$
  - How much of each product should be produced?
    - e.g.: balance run-out time

Aggregate Planning

- **Advanced Production Planning Models**
  - Multiple Products
  - same notation as before
  - add subscript i for product i

  - Objective function

$$\min \sum_{t=1}^{T} \left(C_t^W W_t + C_t^H H_t + C_t^L L_t + \sum_{i=1}^{N} C_{it}^P P_{it} + C_{it}^I I_{it} \right)$$

Aggregate Planning

subject to

$$\sum_{i=1}^{N} \left(\frac{1}{n_{it}} \right) P_{it} \le W_t \qquad t = 1, 2, \ldots, T$$

$$W_t = W_{t-1} + H_t - L_t \qquad \text{t=1,2,...,T}$$

$$I_{it} = I_{it-1} + P_{it} - D_{it} \qquad \text{t=1,2,...,T; i=1,2,...,N}$$

$$P_{it}, W_t, H_t, L_t, I_{it} \ge 0 \qquad \text{t=1,2,...,T; i=1,2,...,N}$$

Aggregate Planning

- **Computational Effort:**
  - 10 products, 12 periods: 276 variables, 144 constraints
  - 100 products, 12 periods: 2436 variables, 1224 constaints

Aggregate Planning

- Example: Carolina Hardwood Product Mix
  - Carolina Hardwood produces 3 types of dining tables;
  - There are currently 50 workers employed who can be hired and laid off at any time;
  - Initial inventory is 100 units for table1, 120 units for table 2 and 80 units for table 3;

t	1	2	3	4
costs of hiring	420	410	420	405
costs of lay off	800	790	790	800
costs per worker	600	620	620	610

Aggregate Planning

- The number of units that can be made by one worker per period:

t	Table 1	Table 2	Table 3
1	200	300	260
2	220	310	255
3	210	300	250
4	200	290	265

- Forecasted demand, unit cost and holding cost per unit are:

	Demand			Unit costs			Holding costs		
t	Table 1	Table 2	Table 3	Table 1	Table 2	Table 3	Table 1	Table 2	Table 3
1	3500	5400	4500	120	150	200	10	12	12
2	3100	5000	4200	125	150	210	9	11	12
3	3000	5100	4100	120	145	205	10	12	11

Aggregate Planning

Minimize

$$600W_1 + 620W_2 + 620W_3 + 610W_4 + 420H_1 + 410H_2 + 420H_3 + 405H_4 + 800L_1 + 790L_2 + 790L_3 + 800L_4$$
$$+ 120P_{11} + 150P_{21} + 200P_{31} + 125P_{12} + 150P_{22} + 210P_{32} + 120P_{13} + 145P_{23} + 205P_{33} + 125P_{14} + 148P_{24} + 205P_{34}$$
$$+ 10I_{11} + 12I_{21} + 12I_{31} + 9I_{12} + 11I_{22} + 12I_{32} + 10I_{13} + 12I_{23} + 11I_{33} + 10I_{14} + 11I_{24} + 11I_{34}$$

subject to

$$\frac{P_{11}}{200} + \frac{P_{21}}{300} + \frac{P_{31}}{260} \leq W_1, \quad \frac{P_{12}}{220} + \frac{P_{22}}{310} + \frac{P_{32}}{255} \leq W_2,$$
$$\frac{P_{13}}{210} + \frac{P_{23}}{300} + \frac{P_{33}}{250} \leq W_3, \quad \frac{P_{14}}{200} + \frac{P_{24}}{290} + \frac{P_{34}}{265} \leq W_4,$$

$$W_1 = 50 + H_1 - L_1, \quad W_2 = W_1 + H_2 - L_2, \quad W_3 = W_2 + H_3 - L_3, \quad W_4 = W_3 + H_4 - L_4,$$

$$I_{11} = 100 + P_{11} - 3500, \quad I_{21} = 120 + P_{21} - 5400, \quad I_{31} = 80 + P_{31} - 4500,$$
$$I_{12} = I_{11} + P_{12} - 3100, \quad I_{22} = I_{21} + P_{22} - 5000, \quad I_{32} = I_{31} + P_{32} - 4200,$$
$$I_{13} = I_{12} + P_{13} - 3000, \quad I_{23} = I_{22} + P_{23} - 5100, \quad I_{33} = I_{32} + P_{33} - 4100,$$
$$I_{14} = I_{13} + P_{14} - 3400, \quad I_{24} = I_{23} + P_{24} - 5500, \quad I_{34} = I_{33} + P_{34} - 4600,$$

$$P_{it}, I_{it}, W_t, H_t, L_t, I_{it} \geq 0$$

Aggregate Planning

⌘ **Multiple Products and Processes**

T = horizon length, in periods
N = number of products
K = number of resource types
t = index of periods, $t = 1, 2, \ldots, T$
i = index of products, $i = 1, 2, \ldots, N$
k = index of resource types, $k = 1, 2, \ldots, K$
D_{it} = forecasted number of units demanded for product i in period t
m_i = number of different processes available to make product i
A_{kt} = amount of resource k available in period t
a_{ijk} = amount of resource k required by one unit of product i if produced by process j
C^P_{ijt} = cost to produce one unit of product i using process j in period t
C^I_{it} = cost to hold one unit of product i in inventory for period t

The decision variables are

P_{ijt} = number of units of product i produced by process j in period t
I_{it} = number of units of product i held in inventory at the end of period t

Aggregate Planning

The linear programming formulation is

$$\text{Minimize} \sum_{t=1}^{T} \sum_{i=1}^{N} \sum_{j=1}^{m_i} (C^P_{ijt} P_{ijt} + C^I_{it} I_{it})$$

subject to

$$\sum_{i=1}^{N} \sum_{j=1}^{m_i} a_{ijk} P_{ijt} \le A_{kt} \qquad t = 1, 2, \ldots, T; \; k = 1, 2, \ldots, K$$

$$I_{it} = I_{it-1} + \sum_{j=1}^{m_i} P_{ijt} - D_{it} \qquad t = 1, 2, \ldots, T; \; i = 1, 2, \ldots, N$$

$$P_{ijt}, I_{it} \ge 0 \qquad t = 1, 2, \ldots, T; \; i = 1, 2, \ldots, N; \; j = 1, 2, \ldots, m_i$$

Aggregate Planning

- Example: Cactus Cycles process plan
- CC produces 2 types of bicycles, street and road;
- Estimated demand and current inventory:

t	initial inventory	1	2	3
street b.	100	1000	1050	1100
road b.	50	500	600	550

- available capacity(hours) and holding costs per bike:

	Capacity(hours)		Holding	
t	Machine	Worker	Street	Road
1	8600	17000	5	6
2	8500	16600	6	7
3	8800	17200	5	7

Aggregate Planning

- process costs (process1, process2) and resource requirement per unit:

	Process1		Process2	
t	Street	Road	Street	Road
1	72	85	80	90
2	74	88	78	95
3	75	84	78	92
Machine hours required	5	8	4	6
Worker hours required	10	12	8	9

Aggregate Planning

Minimize

$$72P_{111} + 80P_{121} + 85P_{211} + 90P_{221}$$
$$+ 74P_{112} + 78P_{122} + 88P_{212} + 95P_{222}$$
$$+ 75P_{113} + 78P_{123} + 84P_{213} + 92P_{223}$$
$$+ 5I_{11} + 6I_{12} + 5I_{13} + 6I_{21} + 7I_{22} + 7I_{23}$$

subject to

$$5P_{111} + 4P_{121} + 8P_{211} + 6P_{221} \le 8600, \quad 10P_{111} + 8P_{121} + 12P_{211} + 9P_{221} \le 17000,$$
$$5P_{112} + 4P_{122} + 8P_{212} + 6P_{222} \le 8500, \quad 10P_{112} + 8P_{122} + 12P_{212} + 9P_{222} \le 16600,$$
$$5P_{113} + 4P_{123} + 8P_{213} + 6P_{223} \le 8800, \quad 10P_{113} + 8P_{123} + 12P_{213} + 9P_{223} \le 17200,$$

$$I_{11} = 100 + P_{111} + P_{121} - 1000, \quad I_{21} = 50 + P_{211} + P_{221} - 500,$$
$$I_{12} = I_{11} + P_{112} + P_{122} - 1050, \quad I_{22} = I_{21} + P_{212} + P_{222} - 600,$$
$$I_{13} = I_{12} + P_{113} + P_{123} - 1100, \quad I_{23} = I_{22} + P_{213} + P_{223} - 550,$$

$$P_{ijt}, I_{it} \ge 0 \qquad t = 1,2,3; \quad i = 1,2; \quad j = 1,2$$

Aggregate Planning

- **solution: Objective Function value = $8.534.166**

	Street Bicycle			Road Bicycle		
	Process			Process		
t	1	2	Inventory	1	2	Inventory
1	900	0	0	118,75	525	193,75
2	1050	0	0	406,25	0	0
3	0	1100	0	550	0	0

Aggregate Planning - Extensions

- **Hopp/Spearman, S. 522-540**
- **Notation:**

X_{it} ... amount of product i produced in period t

r_i ... net profit from one unit of product i

S_{it} ... amount of product i sold in period t

a_{ij} ... time required on workstation j to produce one unit of product i

c_{jt} ... capacity of workstation j in period t in units (consistent with a_{ij})

Aggregate Planning - Extensions

- **Backorders**

$$\max \sum_{t=1}^{\bar{t}} \sum_{i=1}^{m} r_i S_{it} - h_i I_{it}^{+} - \pi_i I_t^{-}$$

subject to

$$\underline{d}_{it} \le S_{it} \le \bar{d}_{it} \qquad \text{for all } i,t$$

$$\sum_{i=1}^{m} a_{ij} X_{it} \le c_{jt} \qquad \text{for all } j,t$$

$$I_{it} = I_{it-1} + X_{it} - S_{it} \qquad \text{for all } i,t$$

$$I_{it} = I_{it}^{+} - I_{it}^{-} \qquad \text{for all } i,t$$

$$X_{it}, S_{it}, I_{it}^{+}, I_{it}^{-} \ge 0 \qquad \text{for all } i,t$$

Aggregate Planning - Extensions

⌘ **Overtime**

l'_j = cost of one hour of overtime at workstation j

O_{jt} = overtime at workstation j in period t in hours

$$\max \sum_{t=1}^{\bar{t}} \left\{ \sum_{i=1}^{m} (r_i S_{it} - h_i I_{it}^{+} - \pi_i I_{it}^{-}) - \sum_{j=1}^{n} l' O_{jt} \right\}$$

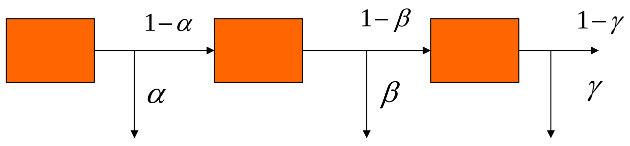
subject to

$$\sum_{i=1}^{m} a_{ij} X_{it} \leq c_{jt} + O_{jt} \qquad \text{for all } i,t$$

$$X_{it}, S_{it}, I_{it}^{+}, I_{it}^{-}, O_{jt} \geq 0 \qquad \text{for all } i,t$$

Aggregate Planning - Extensions

⌘ **Yield loss**



$\alpha, \beta, \gamma \ldots$ fraction of output that is lost

$y_{ij} \ldots$ cumulative yield from station j onward

(including station j) for product i

we must release $\frac{d}{y_{ij}}$ units of i into station j

Aggregate Planning - Extensions

- **Basic model + Yield loss extension (no backorders)**

$$\max \sum_{t=1}^{\bar{t}} \sum_{i=1}^{m} (r_i S_{it} - h_i I_{it})$$

subject to

$$\underline{d}_{it} \le S_{it} \le \bar{d}_{it} \qquad \text{for all } i,t$$

$$\sum_{i=1}^{m} \frac{a_{ij} X_{it}}{y_{ij}} \le c_{jt} \qquad \text{for all } j,t$$

$$I_{it} = I_{it-1} + X_{it} - S_{it} \qquad \text{for all } i,t$$

$$X_{it}, S_{it}, I_{it} \ge 0 \qquad \text{for all } i,t$$

Aggregate Planning - WorkforcePlanning

- **Single product, workforce resizing, overtime allocation**
- **Notation**

b = number of man-hours required to produce one unit of product

l = cost of regular time in dollars/man-hour

l' = cost of overtime in dollars/man-hour

e = cost to increase workforce by one man-hour per period

e′ = cost to decrease workforce by one man-hour per period

W_t = workforce in period t in man-hours of regular time

H_t = increase in workforce from period t-1 to t in man-hours

F_t = decrease in workforce from period t-1 to t in man-hours

O_t = overtime in period t in hours

Aggregate Planning - WorkforcePlanning

- **LP formulation: maximize net profit, including labor, overtime, holding, and hiring/firing costs**
- **subject to constraints on sales, capacity,...**

$$\max \sum_{t=1}^{\bar{t}} \left\{ rS_t - hI_t - lW_t - l'O_t - eH_t - e'F_t \right\}$$

subject to

$$\underline{d}_t \le S_t \le \overline{d}_t \qquad \text{for all } t$$

$$a_j X_t \le c_{jt} \qquad \text{for all j}, t$$

$$I_t = I_{t-1} + X_t - S_t \qquad \text{for all } t$$

$$W_t = W_{t-1} + H_t - F_t \qquad \text{for all } t$$

$$bX_t \le W_t + O_t \qquad \text{for all } t$$

$$X_t, S_t, I_t, O_t, W_t, F_t, H_t \ge 0 \qquad \text{for all } t$$

AP-WP Example

- **Revenue: 1000$**
- **worker capacity: 168h/month**
- **initially 15 workers**
- **no initial inventory**
- **holding costs: 10$/unit/month**
- **regular labor costs: 35$/hour**
- **overtime: 150% of regular**
- **hiring costs: 2500$ (2500/168 ~ 15$ per man-hour)**
- **lay-off costs: 1500$ (1500/168 ~ 9$ per man-hour)**
- **no backordering**
- **demands over 12 months:**

200, 220, 230, 300, 400, 450, 320, 180, 170,170, 160, 180

- **demands must be met! (S=D)**

AP-WP Example(cont.)

- **Determine over a 12 month horizon:**
  - **Number of workers: W**
  - **Output: X**
  - **Overtime use: O**
  - **Inventory: I**
  - **(H, F are additional choice variables in the model)**

Aggregate Planning - WorkforcePlanning

Parameters:													
r	1000												
h	10												
l	35												
l'	52,5												
e	15												
e'	9												
b	12												
I_0	0												
W_0	2520												
t	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	total
	200	220	230	300	400	450	320	180	170	170	160	180	2980
Desicion Variables:													
t	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	
Xt													
Wt													
Ht													
Ft													
It													
Ot													
Objective:													
Profit	2980000	$											

Aggregate Planning - WorkforcePlanning

Constraints:				
I1-I0-X1	0,00	=	-200	d_1
I2-I1-X2	0,00	=	-220	d_2
I3-I2-X3	0,00	=	-230	d_3
I4-I3-X4	0,00	=	-300	d_4
I5-I4-X5	0,00	=	-400	d_5
I6-I5-X6	0,00	=	-450	d_6
I7-I6-X7	0,00	=	-320	d_7
I8-I7-X8	0,00	=	-180	d_8
I9-I8-X9	0,00	=	-170	d_9
I10-I9-X10	0,00	=	-170	d_10
I11-I10-X11	0,00	=	-160	d_11
I12-I11-X12	0,00	=	-180	d_12
W1-W0-H1+F1	-2520,00	=	0	
W2-W1-H2+F2	0,00	=	0	
W3-W2-H3+F3	0,00	=	0	
W4-W3-H4+F4	0,00	=	0	
W5-W4-H5+F5	0,00	=	0	
W6-W5-H6+F6	0,00	=	0	
W7-W6-H7+F7	0,00	=	0	
W8-W7-H8+F8	0,00	=	0	
W9-W8-H9+F9	0,00	=	0	
W10-W9-H10+F10	0,00	=	0	
W11-W10-H11+F11	0,00	=	0	
W12-W11-H12+F12	0,00	=	0	

bX1-W1-O1	0,00	<=	0
bX2-W2-O2	0,00	<=	0
bX3-W3-O3	0,00	<=	0
bX4-W4-O4	0,00	<=	0
bX5-W5-O5	0,00	<=	0
bX6-W6-O6	0,00	<=	0
bX7-W7-O7	0,00	<=	0
bX8-W8-O8	0,00	<=	0
bX9-W9-O9	0,00	<=	0
bX10-W10-O10	0,00	<=	0
bX11-W11-O11	0,00	<=	0
BX12-W12-O12	0,00	<=	0

Aggregate Planning - WorkforcePlanning

Parameters:													
r	1000												
h	10												
l	35												
l'	52,5												
e	15												
e'	9												
b	12												
I_0	0												
W_0	2520												
t	1	2	3	4	5	6	7	8	9	10	11	12	total
	200	220	230	300	400	450	320	180	170	170	160	180	2980
Desicion Variables:													
t	1	2	3	4	5	6	7	8	9	10	11	12	
Xt	302,86	302,86	302,86	302,86	302,86	302,86	302,86	180,00	170,00	170,00	170,00	170,00	2980,00
Wt	3634,29	3634,29	3634,29	3634,29	3634,29	3634,29	3634,29	2160,00	2040,00	2040,00	2040,00	2040,00	35760,00
Ht	1114,29	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1114,29
Ft	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1474,29	120,00	0,00	0,00	0,00	1594,29
It	102,86	185,71	258,57	261,43	164,29	17,14	0,00	0,00	0,00	0,00	10,00	0,00	1000,00
Ot	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Objective:													
Profit	1687337	$											

Aggregate Planning-Summary

The following scenarios have been discussed:

- **single product, single resource, single process**
  find: workforce, output, inventory (w. or w/o backorders)
- **multiple products, single resource, single process**
  find: workforce, all outputs, all inventories (w. or w/o backorders)
- **multiple products, multiple resources, multiple processes (workforce given)**
  find: all outputs, all inventories, use of processes

Aggregate Planning-Summary

The following scenarios have been discussed:

- **multiple products, multiple workstations (workstation capcities given)**
  find: all sales, all outputs, all inventories (w. or w/o backorders)
- **multiple products, multiple workstations**
  find: all sales, all outputs, all inventories (w. or w/o backorders), OT
- **single product, multiple workstations, one resource**
  find: workforce, all sales, all outputs, all inventories (w. or w/o backorders), OT

Aggregate Planning

- **Work to do:**
- **Examples: 5.7, 5.8abcdef, 5.9abcd, 5.10abcd, 5.16abcd, 5.21, 5.22, 5.29, 5.30**

  Replace capacity columns of table in problem 5.29 with

Month	Machine	Worker
1	1350	19000
2	1270	19000
3	1350	19500

- **Minicase BF SWING II**

Chapter 7

Production, Capacity and Material Planning

Production, Capacity and Material Planning

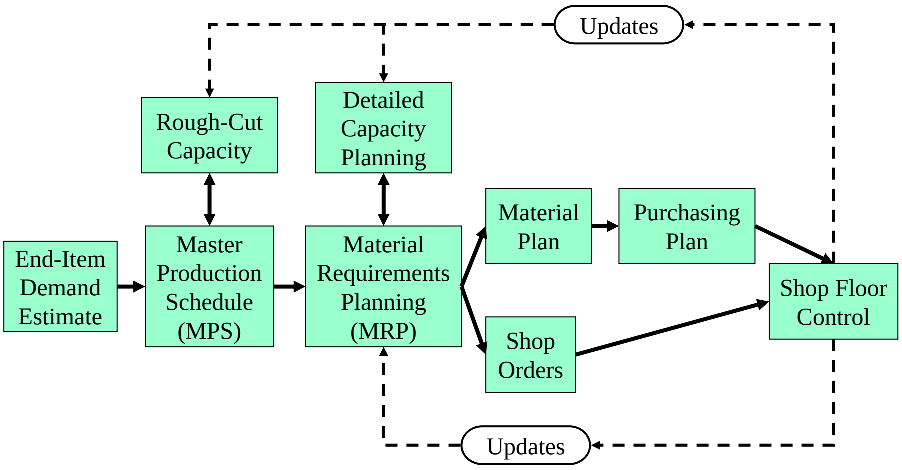
- **Production plan**
  - quantities of final product, subassemblies, parts needed at distinct points in time
- **To generate the Production plan we need:**
  - end-product demand forecasts
  - Master production schedule
- **Master production schedule (MPS)**
  - delivery plan for the manufacturing organization
  - exact amounts and delivery timings for each end product
  - accounts for manufacturing constraints and final goods inventory

Production, Capacity and Material Planning

Based on the MPS:

- **rough-cut capacity planning**
- **Material requirements planning**
  - determines material requirements and timings for each phase of production
  - *detailed capacity planning*

Production, Capacity and Material Planning



Master Production Scheduling

- **Aggregate plan**
- **demand estimates for individual end-items**
- **demand estimates vs. MPS**
  - inventory
  - capacity constraints
  - availability of material
  - production lead time
  - ...
- **Market environments**
  - make-to-stock (MTS)
  - make-to-order (MTO)
  - assemble-to-order (ATO)

Master Production Scheduling

- **MTS**
  - produces in batches
  - minimizes customer delivery times at the expense of holding finished-goods inventory
  - MPS is performed at the end-item level
  - production starts before demand is known precisely
  - small number of end-items, large number of raw-material items
- **MTO**
  - no finished-goods inventory
  - customer orders are backlogged
  - MPS is order driven, consisits of firm delivery dates

Master Production Scheduling

- **ATO**
  - large number of end-items are assembled from a relatively small set of standard subassemblies, or modules
  - automobile industry
  - MPS governs production of modules (forecast driven)
  - Final Assembly Schedule (FAS) at the end-item level (order driven)
  - 2 lead times, for consumer orders only FAS lead time relevant

Master Production Scheduling

- MPS- SIBUL manufactures phones
  - **three desktop models A, B, C**
  - **one wall telephone D**
  - **MPS is equal to the demand forecast for each model**

WEEKLY MPS (= FORECAST)		Jan				Feb			
		Week				Week			
Product		**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Model A		1000	1000	1000	1000	2000	2000	2000	2000
Model B			500	500		350			350
Model C		1500	1500	1500	1500	1000		1000	1000
Model D		600		600			300	200	
weekly total		3100	3000	3600	2500	3350	2300	3200	3350
monthly total		12200				12200			

Master Production Scheduling

- **MPS Planning - Example**
  - **MPS plan for model A of the previous example:**
  - **Make-to-stock environment**
  - **No safety-stock for end-items**
    - $I_t = I_{t-1} + Q_t - \max\{F_t, O_t\}$
    - I_t = end-item inventory at the end of week t
    - Q_t = manufactured quantity to be completed in week t
    - F_t = forecast for week t
    - O_t= customer orders to be delivered in week t

INITIAL DATA Model A		Jan				Feb			
Current Inventory = 1600		Week				Week			
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
forecast Ft		1000	1000	1000	1000	2000	2000	2000	2000
orders Ot		1200	800	300	200	100			

Master Production Scheduling

- Batch production: batch size = 2500
  - I_t = max{0, I_{t-1} } – max{F_t, O_t}

$$Q_t = \begin{cases} 0, \text{if } I_t > 0 \\ 2500, \text{otherwise} \end{cases}$$

  - I_1 = max{0, 1600} – max{1000, 1200} = 400 >0
  - I_2 = max{0, 400} – max{1000, 800} = -600 <0 => Q2 = 2500
  - I_2 = 2500 + 400 – max{1000, 800} = 1900, etc.

MPS		Jan				Feb			
Current Inventory = 1600		Week				Week			
		1	2	3	4	5	6	7	8
forecast Ft		1000	1000	1000	1000	2000	2000	2000	2000
orders Ot		1200	800	300	200	100			
Inventory It	1600	400	1900	900	2400	400	900	1400	1900
MPS Qt			2500		2500		2500	2500	2500
ATP		400	1400		2200		2500	2500	2500

Master Production Scheduling

- Available to Promise (ATP)
  - ATP_1 = 1600 + 0 – 1200 = 400
  - ATP_2 = 2500 –(800 + 300) = 1400, etc.
  - Whenever a new order comes in, ATP must be updated
- Lot-for-Lot production

MPS		Jan				Feb			
Current Inventory = 1600		Week				Week			
		1	2	3	4	5	6	7	8
forecast Ft		1000	1000	1000	1000	2000	2000	2000	2000
orders Ot		1200	800	300	200	100			
Inventory It	1600	400	0	0	0	0	0	0	0
MPS Qt		0	600	1000	1000	2000	2000	2000	2000
ATP		400	0	700	800	1900	2000	2000	2000

Master Production Scheduling

- **MPS Modeling**
  - differs between MTS-ATO and MTO
  - find final assembly lot sizes
  - additional complexity because of joint capacity constraints
  - cannot be solved for each product independently

Master Production Scheduling

- **Make-To-Stock-Modeling**

Q_{it} = production quantity of product i in period t
I_{it} = Inventory of product i at end of period t
D_{it} = demand (requirements) for product i in time period t
a_i = production hours per unit of product i
h_i = inventory holding cost per unit of product i per time period
A_i = set-up cost for product i
G_t = production hours available in period t

$y_{it} = 1$, if set-up for product i occurs in period t ($Q_{it} > 0$)

Master Production Scheduling

- **Make-To-Stock-Modeling**

$$\min \sum_{i=1}^{n} \sum_{t=1}^{T} (A_i y_{it} + h_i I_{it})$$

$$I_{i,t\text{-}1} + Q_{it} - I_{it} = D_{it} \qquad \text{for all (i,t)}$$

$$\sum_{i=1}^{n} \mathrm{a}_i Q_{it} \le G_t \qquad \text{for all t}$$

$$\mathrm{Q}_{\mathrm{it}} - y_{it} \sum_{k=1}^{T} D_{ik} \le 0 \qquad \text{for all (i,t)}$$

$$\mathrm{Q}_{\mathrm{it}} \ge 0; I_{it} \ge 0; y_{it} \in \{0,1\}$$

Master Production Scheduling

- **Assemble-To-Order Modeling**
- **two master schedules**
  - MPS: forecast-driven
  - FAS: order driven
- **overage costs**
  - holding costs for modules and assembled products
- **shortage costs**
  - final product assemply based on available modules
    → no explicit but implicit shortage costs for modules
  - final products: lost sales, backorders

Master Production Scheduling

- *m* module types and *n* product types
- Q_{kt} = quantity of module k produced in period t
- g_{kj} = number of modules of type k required to assemble order j

- **Decision Variables:**
- I_{kt} = inventory of module k at the end of period t
- y_{jt} = 1, if order j is assembled and delivered in period t; 0, otherwise
- h_k = holding cost
- π_{jt} = penalty costs, if order j is satisfied in period t and order j is due in period t′ (t′<t); holding costs if t′ > t

Master Production Scheduling

- **Assemble-To-Order Modeling**

$$\min \sum_{k=1}^{m}\sum_{t=1}^{L} h_k I_{kt} + \sum_{j=1}^{n}\sum_{t=1}^{L} \pi_{jt} y_{jt}$$

subject to

$$I_{kt} = I_{k,t-1} + Q_{kt} - \sum_{j=1}^{n} g_{kj} y_{jt} \qquad \text{for all (k, t)}$$

$$\sum_{j=1}^{n} a_j y_{jt} \le G_t \qquad \text{for all t}$$

$$\sum_{t=1}^{L} y_{jt} = 1 \qquad \text{for all j}$$

$$I_{kt} \ge 0; \quad y_{jt} \in \{0,1\} \qquad \text{for all (j, k, t)}$$

Master Production Scheduling

- **Capacity Planning**
  - Bottleneck in production facilities
  - Rough-Cut Capacity Planning (RCCP) at MPS level
  - feasibility
  - detailed capacity planning (CRP) at MRP level
  - both RCCP and CRP are only providing information

Master Production Scheduling

MPS:	January			
	Week			
Product	1	2	3	4
A	1000	1000	1000	1000
B	-	500	500	-
C	1500	1500	1500	1500
D	600	-	600	-

	Bill of capacity (min)	
	Assembly	Inspection
A	20	2
B	24	2.5
C	22	2
D	25	2.4

	Capacity requires (hr)				
	Week				
	1	2	3	4	Available capacity per week
Assembly	1133	1083	1333!!	883	1200
Inspection	107	104	128!!	83	110

- **weekly capacity requirements?**
- **Assembly: 1000*20 + 1500*22 + 600*25 = 68000 min = 1133,33 hr**
- **Inspection: 1000*2 + 1500*2 + 600*2,4 = 6440 min = 107,33 hr etc.**
- **available capacity per week is 1200 hr for the assembly work center and 110 hours for the inspection station;**

Master Production Scheduling

- **Infinite capacity planning (information providing)**
- **finding a feasible cost optimal solution is a NP-hard problem**

- **if no detailed bill of capacity is available: capacity planning using overall factors (globale Belastungsfaktoren)**
  - required input:
  - MPS
  - standard hours of machines or direct labor required
  - historical data on individual shop workloads (%)

- Example from Günther/Tempelmeier
  - C133.3: overall factors

Master Production Scheduling

capacity planning using overall factors

	week					
product	**1**	**2**	**3**	**4**	**5**	**6**
A	100	80	120	100	120	60
B	40	-	60	-	40	-

product	**work on critical machine**	**work on non-critical machine**	**Total**
A	1	2	3
B	4	2	6

historic capacity requirements on critical machines:
40 % on machine a
60 % on machine b

Master Production Scheduling

in total 500 working units are available per week, 80 on machine a and 120 on machine b;

Solution:
overall factor = time per unit x historic capacity needs

product A:
machine a: 1 x 0,4 = 0,4
machine b: 1 x 0,6 = 0,6

product B:
machine a: 4 x 0,4 = 1,6
machine b: 4 x 0,6 = 2,4

Master Production Scheduling

capacity requirements: product A

machine	week					
	1	2	3	4	5	6
a	40	32	48	40	48	24
b	60	48	72	60	72	36
other	200	160	240	200	240	120

capacity requirements: product B

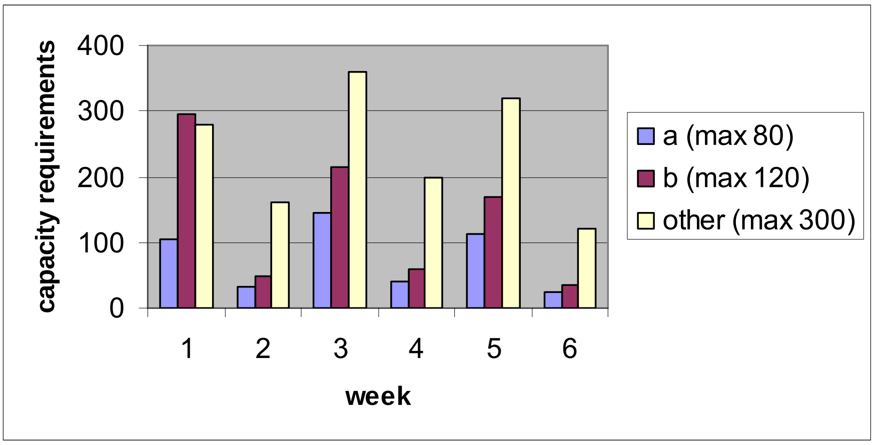
machine	week					
	1	2	3	4	5	6
a	64	-	96	-	64	-
b	96	-	144	-	96	-
other	80	-	120	-	80	-

Master Production Scheduling

total capacity requirements

machine	**week**					
	1	**2**	**3**	**4**	**5**	**6**
a	104	32	144	40	112	24
b	156	48	216	60	168	36
other	280	160	360	200	320	120

Master Production Scheduling



Master Production Scheduling

- **Capacity Modeling**
  - heuristic approach for finite-capacity-planning
  - based on input/output analysis
  - relationship between capacity and lead time

  - G= work center capacity
  - R_t= work released to the center in period t
  - Q_t= production (output) from the work center in period t
  - W_t= work in process in period t
  - U_t= queue at the work center measured at the beginning of period t, prior to the release of work
  - L_t= lead time at the work center in period t

Master Production Scheduling

$$Q_t = \min\{G, U_{t-1} + R_t\}$$
$$U_t = U_{t-1} + R_t - Q_t$$
$$W_t = U_{t-1} + R_t = U_t + Q_t$$
$$L_t = \frac{W_t}{G}$$

- **Lead time is not constant**
- **assumptions:**
  - constant production rate
  - any order released in this period is completed in this period

Master Production Scheduling

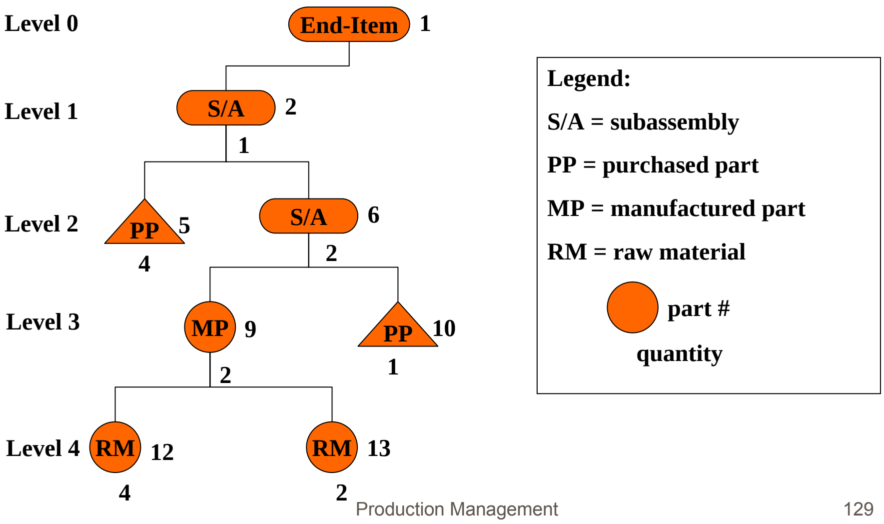
- **Example**

	Period						
	0	1	2	3	4	5	6
G (hr/week)		36	36	36	36	36	36
R_t (hours)		20	30	60	20	40	40
Q_t (hours)		*30*	*30*	*36*	*36*	*36*	*36*
U_t (hours)	10	*0*	*0*	*24*	*8*	*12*	*16*
W_t (hours)		*30*	*30*	*60*	*44*	*48*	*52*
L_t(weeks)		*0,83*	*0,83*	*1,67*	*1,22*	*1,33*	*1,44*

Material Requirements Planning

- **Inputs**
  - master production schedule
  - inventory status record
  - bill of material (BOM)
- **Outputs**
  - planned order releases
    - purchase orders(supply lead time)
    - workorders(manufacturing lead time)

Material Requirements Planning

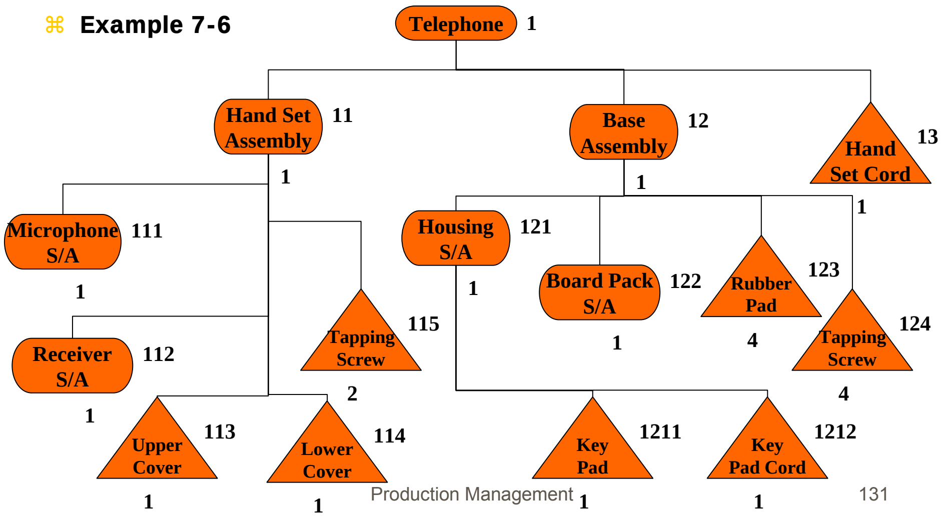


Material Requirements Planning

- **MRP Process**
  - goal is to find net requirements (trigger purchase and work orders)
  - explosion
    - Example:
    - MPS, 100 end items
    - yields gross requirements
  - netting
    - Net requirements = Gross requirements - on hand inventory - quantity on order
    - done at each level prior to further explosion
  - offsetting
    - the timing of order release is determined
  - lotsizing
    - batch size is determined

Material Requirements Planning

⌘ **Example 7-6**



Material Requirements Planning

PART 11 (gross requirements given)
net requirements?
Planned order release?
Net requ.(week 2) = 600 – (1600 + 700) = -1700 =>Net requ.(week2) = **0**
Net requ.(week 3) = 1000 – (1700 + 200) = -900 =>Net requ.(week3) = **0**
Net requ.(week 4) = 1000 – 900 = **100** etc.

	week								
	current	1	2	3	4	5	6	7	8
gross requirements			600	1000	1000	2000	2000	2000	2000
scheduled receipts		400	700	200					
projected inventory balance	1200	1600	1700	900	0	0	0	0	0
net requirements					100	2000	2000	2000	2000
planned receipts									
planned order release									

Material Requirements Planning

Assumptions:
lot size: 3000
lead time: 2 weeks

	week								
	current	1	2	3	4	5	6	7	8
gross requirements			600	1000	1000	2000	2000	2000	2000
scheduled receipts		400	700	200					
projected inventory balance	1200	1600	1700	900	**2900**	**900**	**1900**	**2900**	**900**
net requirements					100	2000	2000	2000	2000
planned receipts					3000		3000	3000	
planned order release			3000		3000	3000			

Material Requirements Planning

- **Multilevel explosion**

part number	description	Qty
12	base assembly	1
121	housing S/A	1
123	rubber pad	4
1211	key pad	1

- lead time is one week
- lot for lot for parts 121, 123, 1211
- part 12: fixed lot size of 3000

Part 12	current	1	2	3	4	5	6	7	8
gross requirements			600	1000	1000	2000	2000	2000	2000
scheduled receipts		400	400	400					
projected inventory balance	800	1200	1000	400	2400	400	1400	2400	400
net requirements		0	0	0	0	0	0	0	0
planned receipts		0	0	0	3000	0	3000	3000	0
planned order release	0	0	0	3000	0	3000	3000	0	0

Part 121	current	1	2	3	4	5	6	7	8
gross requirements	0	0	0	3000	0	3000	3000	0	0
scheduled receipts									
projected inventory balance	500	500	500	0	0	0	0	0	0
net requirements		0	0	0	0	0	0	0	0
planned receipts		0	0	2500	0	3000	3000	0	0
planned order release		0	2500	0	3000	3000	0	0	0

Part 123	current	1	2	3	4	5	6	7	8
gross requirements	0	0	0	12000	0	12000	12000	0	0
scheduled receipts			10000						
projected inventory balance	15000	15000	25000	13000	13000	1000	0	0	0
net requirements		0	0	0	0	0	0	0	0
planned receipts		0	0	0	0	0	11000	0	0
planned order release		0	0	0	0	11000	0	0	0

Part 1211	current	1	2	3	4	5	6	7	8
gross requirements	0	0	2500	0	3000	3000	0	0	0
scheduled receipts		1500							
projected inventory balance	1200	2700	200	200	0	0	0	0	0
net requirements		0	0	0	0	0	0	0	0
planned receipts		0	0	0	2800	3000	0	0	0
planned order release		0	0	2800	3000	0	0	0	0

Material Requirements Planning

- **MRP Updating Methods**
  - MRP systems operate in a dynamic environment
  - regeneration method: the entire plan is recalculated
  - net change method: recalculates requirements only for those items affected by change

	February				Updated MPS for February			
	Week				Week			
Product	5	6	7	8	5	6	7	8
A	2000	2000	2000	2000	2000	2000	2300	1900
B	350	-	-	350	500	-	200	150
C	1000	-	1000	1000	1000	-	800	1000
D	-	300	200	-	-	300	200	-

	Net Change for February			
	Week			
Product	5	6	7	8
A			300	-100
B	150		200	-200
C			-200	
D				

Material Requirements Planning

- **Additional Netting procedures**
  - implosion:
    - opposite of explosion
    - finds common item
  - combining requirements:
    - process of obtaining the gross requirements of a common item
  - pegging:
    - identify the item's end product
    - useful when item shortages occur

Material Requirements Planning

- **Lot Sizing in MRP**
  - minimize set-up and holding costs
  - can be formulated as MIP
  - a variety of heuristic approaches are available
  - simplest approach: use independent demand procedures (e.g. EOQ) at every level

Material Requirements Planning

- **MIP Formulation**
  - Indices:

$i = 1 \ldots P$	label of each item in BOM (assumed that all labels are sorted with respect to the production level starting from the end-items)
$t = 1 \ldots T$	period t
$m = 1 \ldots M$	resource m

  - Parameters:

$\Gamma(i)$	set of immediate successors of item i
$\Gamma^{-1}(i)$	set of immediate predeccessors of item i
s_i	setup cost for item i
c_{ij}	quantity of itme i required to produce item j
h_i	holding cost for one unit of item i
a_{mi}	capacity needed on resource m for one unit of item i
b_{mi}	capacity needed on resource m for the setup process of item i
L_{mt}	available capacity of resource m in period t
oc_m	overtime cost of resource m
G	large number, but as small as possible (e.g. sum of demands)
D_{it}	external demand of item i in period t

Material Requirements Planning

- Decision variables:

x_{it}	deliverd quantity of item i in period t
I_{it}	inventory level of item i at the end of period t
O_{mt}	overtime hours required for machine m in period t
y_{it}	binary variable indicating if item i is produced in period t (=1) or not (=0)

- Equations:

$$\min \sum_{i=1}^{P} \sum_{t=1}^{T} (s_i y_{it} + h_i I_{it}) + \sum_{t=1}^{T} \sum_{m=1}^{M} oc_m O_{mt}$$

$$I_{i,t} = I_{i,t-1} + x_{i,t} - \sum_{j \in \Gamma(i)} c_{ij} x_{jt} - D_{it} \quad \forall i,t \qquad x_{it} - G y_{it} \le 0 \quad \forall i,t$$

$$\sum_{i=1}^{P} (a_{mi} x_{it} + b_{mi} y_{it}) \le L_{mt} + O_{mt} \quad \forall m,t \qquad x_{it}, I_{it}, O_{mt} \ge 0, \quad y_{it} \in \{0,1\} \quad \forall i,m,t$$

Material Requirements Planning

- **Multi-Echelon Systems**
  - Multi-echelon inventory
  - each level is referred as an **echelon**
  - "total inventory in the system varies with the number of stocking points"
  - Modell (Freeland 1985):
    - demand is insensitive to the number of stocking points
    - demand is normally distributed and divided evenly among the stocking points,
    - demands at the stocking points are independent of one another
    - a (Q,R) inventory policy is used
    - β-Service level (fill rate) is applied
    - Q is determined from the EOQ formula

Material Requirements Planning

- **Reorder point in (Q,R) policies:**
  - i: total annual inventory costs (%)
  - c: unit costs
  - A: ordering costs
  - τ :lead time
  - σ_τ: variance of demand in lead time
- **given a fill rate β choose $z(\beta)$ such that:**

$$L(z) = \int_z^\infty (y - z)\ \phi(y)dy = \frac{(1-\beta)Q}{\sigma_\tau}$$

ϕ: density of ***N(0,1)*** distribution; $L(z)$: standard loss function

Unit Normal Linear Loss Integral L(Z)

Z	.00	.02	.04	.06	.08
0.00	.3989	.3890	.3793	.3697	.3602
0.10	.3509	.3418	.3329	.3240	.3154
0.20	.3069	.2986	.2904	.2824	.2745
0.30	.2668	.2592	.2518	.2445	.2374
0.40	.2304	.2236	.2170	.2104	.2040
0.50	.1978	.1917	.1857	.1799	.1742
0.60	.1687	.1633	.1580	.1528	.1478
0.70	.1429	.1381	.1335	.1289	.1245
0.80	.1202	.1160	.1120	.1080	.1042
0.90	.1004	.0968	.0933	.0899	.0866
1.00	.0833	.0802	.0772	.0742	.0714
1.10	.0686	.0660	.0634	.0609	.0585
1.20	.0561	.0539	.0517	.0496	.0475
1.30	.0456	.0437	.0418	.0401	.0383
1.40	.0367	.0351	.0336	.0321	.0307
1.50	.0293	.0280	.0268	.0256	.0244
1.60	.0233	.0222	.0212	.0202	.0192
1.70	.0183	.0174	.0166	.0158	.0150
1.80	.0143	0..0136	.0129	.0122	.0116
1.90	.0110	.0104	.0099	.0094	.0089
2.00	.0084	.0080	.0075	.0071	.0067
2.10	.0063	.0060	.0056	.0053	.0050
2.20	.0047	.0044	.0042	.0039	.0037
2.30	.0036	.0034	.0032	.0030	.0028
2.40	.0027	.0026	.0024	.0023	.0022
2.50	.0021	.0018	.0017	.0016	.0016

Material Requirements Planning

- **Safety stock:** $s = z \cdot \sigma_\tau$
- **Reorder point:** $R = \overline{D}_\tau + z \cdot \sigma_\tau$
- **Order quantity:** $Q = EOQ = \sqrt{\dfrac{2A\overline{D}}{ic}}$
- **Average inventory:** $\bar{I}(1) = \dfrac{Q}{2} + s$

$\bar{I}(n)$ = average inventory for n stocking points

$$\bar{I}(1) = \frac{1}{2}\sqrt{\frac{2A\overline{D}}{ic}} + z\sigma_\tau$$

Material Requirements Planning

for two stocking points :

demand at each point : D/2

variance of lead - time demand : $\sigma_\tau^2 / 2$

standard deviation is : $\sigma_\tau / \sqrt{2}$

average inventory at each stocking point is :

$$\frac{1}{2}\sqrt{\frac{2\mathrm{A}\bar{\mathrm{D}}/2}{\mathrm{ic}}} + \frac{z\sigma_\tau}{\sqrt{2}} = \frac{1}{\sqrt{2}}(Q/2+s)$$

Material Requirements Planning

the average inventory for two stocking point is :

$$\bar{I}(2) = 2\left[\frac{1}{\sqrt{2}}(Q/2+s)\right] = \sqrt{2}\left(Q/2+s\right) = \sqrt{2}\cdot\bar{I}(1)$$

$$\bar{I}(n) = \sqrt{n}\cdot\bar{I}(1)$$

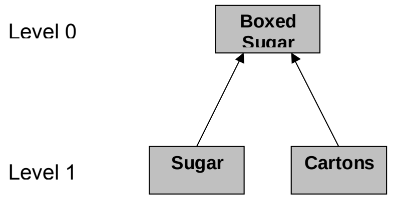
for each level the safety stock is : s/$\sqrt{n}$

the total safety stock is $\sqrt{n}\cdot s$

Material Requirements Planning

Example: At the packaging department of a sugar refinery:

A very-high-grade powdered sugar:



Sugar-refining lead time is five days;
Production lead time (filling time) is negligible;
Annual demand: D = 800 tons and σ= 2,5
Lead-time demand is normally distributed with $D\tau$ = 16 tons and $\sigma\tau$ = 3,54 tons
Fill rate = 95%
A = \$50, c = \$4000, i = 20%

Material Requirements Planning

Inventory at level 0 and 1? Safety stock?

$$Q = \sqrt{\frac{2AD}{ic}} = \sqrt{\frac{2x50\ x800}{800}} = 10\ tons$$

ß = 0,95 => z = 0,71
s = $z\sigma\tau$ = 0,71x3,54 = 2,51 tons

Suppose we keep inventory in level 0 only, i.e., n = 1:

$$I(1) = \frac{Q}{2} + s = \frac{10}{2} + 2{,}51 = 7{,}51 tons$$

Suppose inventory is maintained at both level 0 and level 1, i.e., n = 2:

$$I(2) = \sqrt{2} I(1) = 10{,}62\ tons$$

The safety stock in each level is going to be:

$$\frac{s}{\sqrt{2}} = \frac{2{,}51}{\sqrt{2}} = 1{,}77 tons$$

Material Requirements Planning

- **MRP as Multi-Echelon Inventory Control**
  - continuous-review type policy (Q,R)
  - hierarchy of stocking points (installation)
  - installation stock policy
  - echelon stock (policy): installation inventory position plus all downstream stock
  - MRP:
    - rolling horizon
    - level by level approach
    - bases ordering decisions on projected future installation inventory level

Material Requirements Planning

- All demands and orders occur at the beginning of the time period
- orders are initiated immediately after the demands, first for the final items and then successively for the components
- all demands and orders are for an integer number of units
- T= planning horizon
- τ_i= lead time for item i
- s_i= safety stock for item I
- R_i= reorder point for item I
- Q_i=Fixed order quantity of item i
- D_{it}= external requirements of item i in period t

Material Requirements Planning

- **Installation stock policies (Q,R^i) for MRP:**
  - a production order is triggered if the installation stock minus safety stock is insufficient to cover the requirements over the next τ_i periods
  - an order may consist of more than one order quantity Q
  - if lead time τ_i = 0, the MRP is equal to an installation stock policy.
  - safety stock = reorder point

Material Requirements Planning

- **Echelon stock policies (Q,R^e) for MRP:**
  - Consider a serial assembly system
  - Installation 1 is the downstream installation (final product)
  - the output of installation i is the input when producing one unit of item i-1 at the immediate downstream installation
  - w_i = installation inventory position at installation i
  - I_i = echelon inventory position at installation i (at the same moment)

$$I_i = w_i + w_{i-1} + \dots w_1$$

  - a multi-echelon (Q,R) policy is denoted by (Q_i, R_i^e)
  - R_i^e gives the reorder point for echelon inventory at i

Material Requirements Planning

$R_1^e = s_1 + D\tau_1$
$R_i^e = s_i + D\tau_i + R_{i-1}^e + Q_{i-1}$

- **Example:**
- **Two-level system, 6 periods**
  $I_1^0 = 18,\ I_2^0 = 38,\ R_1^e = 20,\ R_2^e = 34,\ Q_1 = 10,\ Q_2 = 30$
- **D = 2 (Item 1), $\tau_1 = 1$, $\tau_2 = 2$**

Material Requirements Planning

	Period	1	2	3	4	5	6
	Demand	2	2	2	2	2	2
Item 1	Level w1	18	26	24	22	20	28
	Production	10	0	0	0	10	0
Item 2	Level w2	10	10	10	10	30	10
	Production	0	0	30	0	0	0

Suppose now that five units were demanded in period 2:

	Period	2	3	4	5	6	7
	Demand	5	2	2	2	2	2
Item 1	Level w1	23	21	19	27	25	23
	Production	0	0	10	0	10	0
Item 2	Level w2	10	10	30	30	30	30
	Production	30	0	0	0	0	0

Material Requirements Planning

- **Lot Size and Lead Time**
  - lead time is affected by capacity constraints
  - lot size affects lead time
- **batching effect**
  - an increase in lot size should increase lead time
- **saturation effect**
  - when lot size decreases, and set-up is not reduced, lead time will increase

- **expected lead time can be calculated using models from queueing theory (M/G/1)**

Material Requirements Planning

$$L = \text{lead time}$$

$$L = \frac{(\lambda/\mu)^2 + \lambda^2\sigma^2}{2\lambda(1-\lambda/\mu)} + \frac{1}{\mu}$$

$$\lambda = \text{mean arrival rate}$$

$$\mu = \text{mean service rate}$$

$$\sigma^2 = \text{service time variance}$$

Material Requirements Planning

D_j = demand per period for product j

t_j = unit - production time for product j

S_j = set - up time for product j

Q_j = lotsize for product j

mean arrival rate of batches :

$$\lambda = \sum_{j=1}^{n} \lambda_j = \sum_{j=1}^{n} \frac{D_j}{Q_j}$$

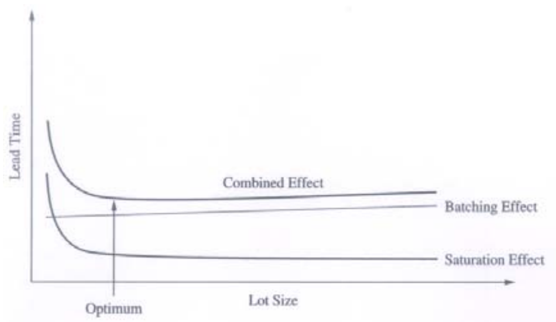
mean service time :

$$\frac{1}{\mu} = \frac{\sum_{j=1}^{n} \lambda_j (S_j + t_j Q_j)}{\sum_{j=1}^{n} \lambda_j}$$

service - time variance :

$$\sigma^2 = \frac{\sum_{j=1}^{n} \lambda_j (S_j + t_j Q_j)^2}{\sum_{j=1}^{n} \lambda_j} - \left(\frac{1}{\mu}\right)^2$$

Material Requirements Planning



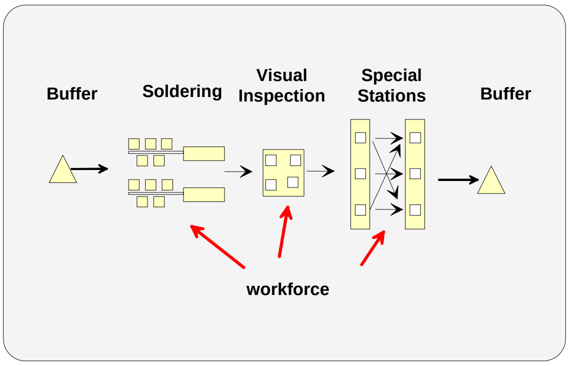
Material Planning

- Work to do: 7.7ab, 7.8, 7.10, 7.11, 7.14 (additional information: available hours: 225 (Paint), 130 (Mast), 100 (Rope)), 7.15, 7.16, 7.17, 7.31-7.34

Chapter 8

Operations Scheduling

Operations Scheduling



Operations Scheduling

- **Scheduling is**
  - the process of organizing, choosing and timing resource usage to carry out all the activities necessary to produce the desired outputs at the desired times, while satisfying a large number of time and relationship constraints among the activities and the resources (Morton and Pentico, 1993).
- **Schedule specifies**
  - the time each job starts and completes on each machine, as well as any additional resources needed.
- **A Sequence is**
  - a simple ordering of the jobs.

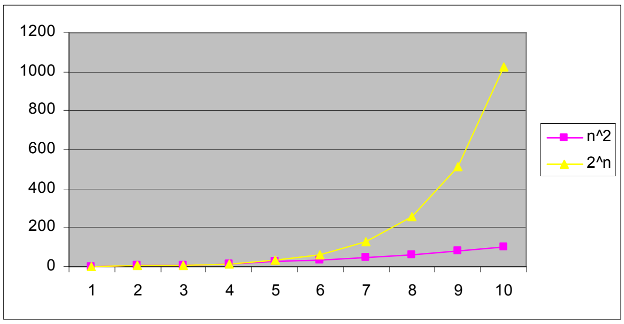
Operations Scheduling

- Determining a **best** sequence
- 32 jobs on a single machine
- 32! Possible sequences approx. 2.6×10^{35}
  - suppose a computer could examine one billion sequences per second
  - it would take 8.4×10^{15} centuries
- real life problems are much more complicated
- Scheduling theory helps to
  - classify the problems
  - identify appropriate measures
  - develop solution procedures

Operations Scheduling

- **Algorithmic complexity**
  - an efficient algorithm is one whose effort of any problem instance is bounded by a polynomial in the problem size, e.g. # of jobs
  - minimal spanning tree can be solved in at most n^2 iterations
  - n: number of edges
  - $O(n^2)$
  - if effort is exponential $O(2^n)$ the algorithm is not efficient
  - branch and bound algorithm for 0/1 variables
  - NP-hard problems: no exact algorithm in polynomial time is known. e.g. Traveling salesman problem
  - Heuristics are usually polynomial algorithms tailored to the specific problem structure

Operations Scheduling



Operations Scheduling

- **Scheduling Theory (Background)**
- **Jobs are**
  - activities to be done
  - processing time known
  - in general continously processed until finished (preemption not allowed)
  - due date
  - release date
  - precedence constraints
  - sequence dependent setup time
  - processed by at most one-machine at the same time

Operations Scheduling

- **Machines (resources)**
  - single machine, parallel machines
  - flow shop:
    - each job must be processed by each machine exactly once
    - all jobs have the same routing
    - a job cannot begin processing on the second machine until it has completed processing on the first
    - assembly line
  - job shop:
    - each job may have a unique routing
  - open shops:
    - job shops in which jobs have no specific routing
    - re-manufacturing and repair

Operations Scheduling

- **Measures**
  - profit, costs
  - it is difficult to relate a schedule to profit and cost
  - **regular measure** is a function of completion time
    - function only increases if at least one completion time in schedule increases

  - n= number of jobs to be processed
  - m= number of machines
  - p_{ik}= time to process job i on machine k
  - r_i = release date of job i
  - d_i = due date of job i
  - w_i = weight of job i relative to the other jobs

Operations Scheduling

- C_i = the completion time
- $F_i = C_i - r_i$, the flowtime
- $L_i = C_i - d_i$, lateness of job i
- $T_i = \max\{0, L_i\}$, tardiness of job i
- $E_i = \max\{0, -L_i\}$, earliness of job i

- $\delta_i = 1$, if job i is tardy ($T_i > 0$)
- $\delta_i = 0$, if job i is on time ($T_i = 0$)

$$C_{max} = \max_{i=1,n}\{C_i\}, \text{ makespan}$$
$$L_{max} = \max_{i=1,n}\{L_i\}, \text{ maximum lateness}$$
$$T_{max} = \max_{i=1,n}\{T_i\}, \text{ maximum tardiness}$$

Operations Scheduling

- **Common proxy objectives**
  - total flowtime
  - total tardiness
  - makespan
  - maximum tardiness
  - number of tardy jobs
  - if not all jobs are equally important weights should be introduced

  - minimizing total completion time is equivalent to minimizing total flowtime or minimizing total tardiness

Operations Scheduling

- **Algorithms:**
  - exact algorithms often based on (worst case scenario) enumeration (e.g. Branch and Bound, Dynamic Programming)
  - heuristic algorithm judged by quality (difference to the optimal solution) and efficacy (computational effort)
  - worst-case bounds are desirable to motivate use of a certain heuristic

Operations Scheduling

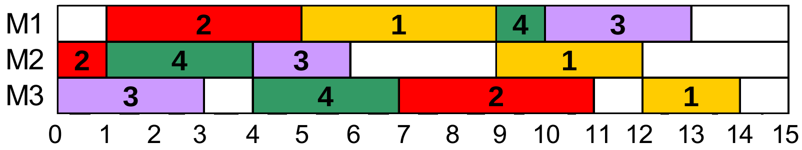
Consider the following four-job, three-machine job-shop scheduling problem:

	Processing time/machine number				
Job	**Op.1**	**Op.2**	**Op.3**	**Release Date**	**Due date**
1	4/1	3/2	2/3	0	16
2	1/2	4/1	4/3	0	14
3	3/3	2/2	3/1	0	10
4	3/2	3/3	1/1	0	8

- **Assume the following sequences:**
  - 2-1-4-3 on M1
  - 2-4-3-1 on M2
  - 3-4-2-1 on M3

Operations Scheduling

Gantt Chart (machine oriented)



Operations Scheduling

$$C_1 = 14, C_2 = 11, C_3 = 13, C_4 = 10$$

The makespan is

$$C_{\max} = \max\{C_1, C_2, C_3, C_4\} = \max\{14, 11, 13, 10) = 14$$

The total flowtime is

$$\sum_i F_i = 14 + 11 + 13 + 10 = 48$$

Operations Scheduling

The lateness and the tardiness of a job:

$$L_1 = 14 - 16 = -2 \qquad T_1 = \max\{0,-2\} = 0$$
$$L_2 = 11 - 14 = -3 \qquad T_2 = \max\{0,-3\} = 0$$
$$L_3 = 13 - 10 = 3 \qquad T_3 = \max\{0,3\} = 3$$
$$L_4 = 10 - 8 = 2 \qquad T_4 = \max\{0,2\} = 2$$

The total lateness is

$$\sum_i L_i = (-2) + (-3) + 3 + 2 = 0$$

The total tardiness is

$$\sum_i T_i = 0 + 0 + 3 + 2 = 5$$

The maximum tardiness is

$$T_{\max} = \max\{0,0,3,2\} = 3$$

Tardy jobs have δi =1, so

The number of tardy jobs is

$$T_1 = 0 \Rightarrow \delta_1 = 0$$
$$T_2 = 0 \Rightarrow \delta_2 = 0$$
$$T_3 > 0 \Rightarrow \delta_3 = 1$$
$$T_4 > 0 \Rightarrow \delta_4 = 1$$

$$N_T = 2$$

Operations Scheduling

- **Single Machine Scheduling**
- **Minimizing Flowtime**
  - Problem data

Job *i*	1	2	3	4	5
p *i*	4	2	3	2	4

  - Sequence: 1-2-3-4-5
  - Total Flowtime=?
  - $F=p_1 + (p_1+p_2) + (p_1+p_2+p_3)+...+(p_1+p_2+...+p_n)$
  - $F= np_1 + (n-1)p_2 +...+p_n$

Operations Scheduling

- Theorem. SPT sequencing minimizes total flowtime on a single machine with zero release times.
- Proof. We assume an optimal schedule is not an SPT sequence.
  - $p_i > p_j$
  -
  - TF(S) = TF(B) + $(t+p_i)$ + $(t+p_i+p_j)$ + TF(A)
  - TF(S') = TF(B) + $(t+ p_j)$ + $(t+ p_j +p_i)$ + TF(A)
  - TF(S)-TF(S')= $p_i - p_j > 0$

Operations Scheduling

SPT-rule ⇒ sequence: 2-4-3-1-5

$C_1 = 11$
$C_2 = 2$
$C_3 = 7$
$C_4 = 4$
$C_5 = 15$

Total flowtime = total completion time =39

- **SPT rule also minimizes**
  - total waiting time
  - mean # of jobs waiting (mean work in progress)
  - total lateness
- **Why?**

Operations Scheduling

- **Minimize weighted Flow-time:** $\sum_{i=1}^{n} w_i F_i$
- weighted SPT (WSPT): order ratios $\frac{p_i}{w_i}$ (nondecreasing**)**
- exact algorithm for weighted flow-time with zero release time (completion time)

Operations Scheduling

Weighted Flowtime

WSPT scheduling

$w_1 = 1, w_2 = 4, w_3 = 3, w_4 = 1, w_5 = 3$

the processing-time-to-weight ratio gives: 4; 0,5; 1; 2; 1,33

the WSPT sequence is the following: 2-3-5-4-1

$$C_1 = 15$$
$$C_2 = 2$$
$$C_3 = 5$$
$$C_4 = 11$$
$$C_5 = 9$$

the value of weighted flowtime is

$$\sum_{i=1}^{5} w_i F_i = 76$$

Operations Scheduling

- **Maximal Tardiness and Maximal Lateness**
  - due date oriented measure
  - earliest due date sequence (EDD)
  - EDD minimizes
    - Maximal Tardiness and
    - Maximal Lateness

Job i	1	2	3	4	5
Due date	16	10	7	7	5
Proc. Time	4	2	3	2	4

- EDD-sequence: 5-3-4-2-1
- Tardiness of the jobs is (0, 0, 2, 1, 0)

Operations Scheduling

- Number of Tardy Jobs
  - Hodgson's algorithm
  - **Step1.** Compute the tardiness for each job in the EDD sequence. Set N_T=0, and let k be the first position containing a tardy job. If no job is tardy go to step 4.
  - **Step 2.** Find the job with the largest processing time in positions 1 to k.

$$\text{Let } p_{[j]} = \max_{i=1,k} p_{[i]} \text{ then } j^* = [j]$$

  - **Step 3.** Remove job j* from the sequence, set N_T=N_T+1, and repeat Step1.
  - **Step 4**. Place the removed N_T jobs in any order at the end of the sequence.
  - **This sequence minimizes the number of tardy jobs**

Operations Scheduling

- **Consider the previous example:**
- **EDD-sequence: 5-3-4-2-1**
- Step1: The tardiness is (0, 0, 2, 1, 0) ⇒ Job 4 in the third position is the first tardy job;
- Step2: The processing times for jobs 5, 3 and 4 are 4, 3, 2, respectively;
  ⇒ largest processing time for job 5
- Step 3: Remove job 5, goto step 1
- Step 1: EDD-sequence is 3-4-2-1; completion times (3, 5, 7, 11) and tardiness (0, 0, 0, 0) ⇒ Go to step 4
- Step 4: schedule that minimizes the number of tardy jobs is 3-4-2-1-5 and has only one tardy job: Job 5

Operations Scheduling

- **Minimize the weighted number of tardy jobs!**
- **NP-hard Problem**
- Heuristic approach: processing-time-to-weight ratio (not exact!)
- Consider the previous example with the following weights:

  $w_1 = 1, w_2 = 4, w_3 = 3, w_4 = 1, w_5 = 3$

- EDD-sequence was 5-3-4-2-1
- **Step 1** first tardy job is job 4
- **Step 2** the processing-time-weight-ratio for jobs 5, 3 and 4 are 4/3, 3/3 and 2/1
- **Step 3** Remove job 4
- **Step 1** EDD-sequence is 5-3-2-1 with no tardiness
- **Step 4** new schedule 5-3-2-1-4 has one tardy job: job 4 with weight 1

Operations Scheduling

- **Minimize Flowtime with no tardy jobs**
  - for all jobs to be on time, the last job must be on time
  - schedulable set of jobs contain all jobs with due dates greater than or equal to the sum of all processing times
  - Start from the end and choose the job with the **largest** proc time among the schedulable jobs, schedule this job last, remove from the list and continue
  - **Optimal algorithm ! (**corresponding alg. For weighted flowtime is only heuristic)
  - **Problem data**

Job *i*	**1**	**2**	**3**	**4**	**5**
p *i*	**4**	**2**	**3**	**2**	**4**
due date	**16**	**11**	**10**	**9**	**12**

Operations Scheduling

- Step 1: Sum of the processing time is 15
  Job 1 has a due-date greater to 15 ⇒ schedule x-x-x-x-1
- Step 2: Sum of the remaining processing-times is 11
  Job 5 has a larger processing time ⇒ schedule x-x-x-5-1
- Step 3: remaining processing time is 7
  All remaining jobs have due dates at least that big
  ⇒ choose the one with the largest processing time ⇒ x-x-3-5-1
- Step 4: Continue ⇒2-4-3-5-1

Operations Scheduling

- **Minimizing total Tardiness**
- general single-machine tardiness problem is NP-hard

- **Heuristic approach for the weighted problem(Rachamadugu/Morton)**
- if all jobs are tardy, minimizing weighted tardiness is equivalent to minimizing weighted completion time, which is accomplished by the WSPT sequence.

- Weight-to-processing-time ratio is used

- Slack of job i, $S_i = d_i - (p_i + t)$ where t is the current time

Operations Scheduling

- A job should not get full WTPTR „credit" if its slack is positive

$$S_i^+ = \max\{0, S_i\}$$

- Average processing time of the jobs:

$$p_{av} = 1/n \sum_{i=1}^{n} p_i$$

- Ratio of the slack to the average processing time of jobs:

$$S_i^+ / p_{av}$$

which is the number of average job lengths until job j is tardy

- Weight of a job is discounted by an exponential function: $\exp(-S_i^+ / \kappa p_{av})$

Operations Scheduling

⌘ Define the priority of job i by

$$\gamma_i = \left(\frac{w_i}{p_i}\right) e^{-[S_i^+ / \kappa \cdot p_{av}]}$$

κ is a parmeter of the heuristic to be chosen by the user (e.g. $\kappa = 2$)

⌘ Sequence jobs in descending order of priorities.

Operations Scheduling

⌘ Rachamadugu and Morton (1982) R&M Heuristics:

- **The owner of** Pensacola Boat Construction **has currently** 10 boats **to construct;**
- If PBC delivers a boat after the delivery date, a penalty proportional to both the value of the boat and the tardiness must be paid.

Job	1	2	3	4	5	6	7	8	9	10
p(i)	8	12	6	10	3	11	9	11	13	7
w(i)	4	1	6	5	1	4	5	9	8	1
d(i)	26	28	32	35	38	48	50	51	53	64

- How should PBC schedule the work to minimize the penalty paid?

Operations Scheduling

- **Penalty is weighted tardiness where weights measure the value of the boat.**
- **κ = 2**
- **Calculate:** $p_{av} = 9$
- **Job1:**

$$\gamma_1 = \left(\frac{w_1}{p_1}\right) e^{-[S_1^+/(\kappa\ p_{av})]} = \left(\frac{4}{8}\right) e^{-[(26-8)/(2x9)]} = 0{,}5e^{-1} = 0{,}18$$

Operations Scheduling

Job	1	2	3	4	5	6	7	8	9	10
p(i)	8	12	6	10	3	11	9	11	13	7
w(i)	4	1	6	5	1	4	5	9	8	1
d(i)	26	28	32	35	38	48	50	51	53	64
w(i)/p(i)	0,5	0,08	1	0,5	0,33	0,36	0,56	0,82	0,62	0,14
S^+(i)	18	16	26	25	35	37	41	40	40	57
S^+(i)/kp_{av}	1	0,89	1,44	1,39	1,94	2,06	2,28	2,22	2,22	3,17
priority	**0,18**	**0,03**	**0,24**	**0,12**	**0,05**	**0,05**	**0,06**	**0,09**	**0,07**	**0,01**

Jobs	3	1	4	8	9	7	5	6	2	10	Sum
gamma_i	0,24	0,18	0,125	0,09	0,07	0,06	0,05	0,047	0,03	0,01	
p_i	6	8	10	11	13	9	3	11	12	7	90
C_i	6	14	24	35	48	57	60	71	83	90	
d_i	32	26	35	51	53	50	38	48	28	64	
T_i	0	0	0	0	0	7	22	23	55	26	133
w_i	6	4	5	9	8	5	1	4	1	1	
w_i T_i	0	0	0	0	0	35	22	92	55	26	230

Operations Scheduling

- Minimizing Earliness and Tardiness with a Common Due-Date

$$Z = \sum_{i=1}^{n} (E_i + T_i)$$

- this is not a regular measure
- assume common due date: $d_j = D$
- Number jobs in LPT sequence: $p_1 \geq p_2 \geq \cdots \geq p_n$
- choose j* = n/2 or n/2+0.5
- if $p_1 + p_3 + \cdots + p_{j^*} \leq D$ then the following sequence is

optimal: 1 - 3 - 5 - 7 - . . . - n - . . .- 6 - 4 - 2

Operations Scheduling

- **Example: 10 Jobs with common due-date 80**

Jobs	A	B	C	D	E	F	G	H	I	J
proc Time	8	18	11	4	15	5	23	25	10	17

Operations Scheduling

- if $p_1 + p_3 + \cdots + p_{j^*} > D$ then apply a heuristic (by Sundararaghavan & Ahmed, 1984)
- **Step 0:** Set $B = D; A = \sum_{i=1}^{n} p_i - D; k = b = 1;$ a = n; use the LPT sequence
- **Step 1:** If B>A:

  assign job k to position b
  b:=b+1
  B:=B-p_k

  else

  assign job k to position a
  a:=a-1
  A:=A-p_k
- **Step 2:** k:=k-1; if k<=n go to step 1.

Operations Scheduling

Problems with non-zero release time

- Non-zero release times typically makes scheduling problems much harder, e.g. SPT does in general not minimize total flowtime
- **Heuristic Approach:**
- At each time t determine the set of **schedulable jobs:** jobs that have been released but not yet processed.
- Choose from the schedulable jobs according to some rule (e.g. SPT for minimizing flowtime)

Operations Scheduling

- **Preemption allowed:**

	j	1	2	3	4	5	6
	r	**12**	**2**	**0**	**11**	**4**	**10**
	p	**8**	**4**	**3**	**6**	**2**	**2**
t=0	rp			**3**			
t=2	rp		4	**1**			
t=3	rp		**4**	C			
t=4	rp		3			**2**	
t=6	rp		**3**			C	
t=9	rp		C				
t=10	rp						**2**
t=11	rp				6		**1**
t=12	rp	8			**6**		C
t=18	rp	**8**			C		
t=24	rp	C					

Operations Scheduling

- **Minimizing makespan with non-zero release time and tails**
- Given n jobs with release times r_i, procssing times p_i, and tails n_i
- **Schrage Heuristics:**
  - Start at t=0
  - 1. Determine schedulable jobs
  - 2. If there are schedulable jobs select the job j* among them with the largest tails, otherwise t=t+1 goto 1.
  - 3. Schedule j* at t
  - 4. If all jobs have been scheduled stop, otherwise set $t = t + p_{j*}$, goto 1.

Operations Scheduling

- **Schrage Heuristics Example: 6 jobs with release times and tails**

j	1	2	3	4	5	6
r_j	12	2	0	11	9	10
p_j	8	4	3	6	2	2
n_j	21	9	2	6	7	10

- **Minimize makespan!**

Operations Scheduling

- Denote by SJ the set of schedulable jobs and by S the scheduled sequence

- Step 1. t = 0, SJ = {3}, S = <3>, t = 3, C_{max} = 5
- Step 2. t=3, SJ = {2}, S = <3-2>, t = 7, C_{ma} = 16
- Step 3. t = 9, SJ = {5}, S = <3-2-5>, t = 11, C_{ma} = 18
- Step 4. t=11, SJ = {4, 6}, S = <3- 2- 5- 6>, t = 13, C_{ma} = 23
- Step 5. t=13, SJ = {1, 4}, S = <3- 2- 5- 6-1>, t = 21, C_{ma} = 42
- Step 6. T=21 SJ = {4}, S = <3- 2- 5- 6- 1- 4>, t = 27, C_{ma} = 42

- **Schrage heuristic is in general not optimal, e.g. B&B model can be used as an exact algorithm**

Operations Scheduling

- **Minimizing Set-Up Times**
  - sequence-dependent set-up times
  - the time to change from one product to another may be significant and may depend on the previous part produced
  - p_{ij} = time to process job j if it immediately follows job i
  - Examples:
    - electronics industry
    - paint shops
    - injection molding
  - minimizes makespan
  - problem is equivalent to the traveling salesman problem (TSP), which is NP-hard.

Operations Scheduling

SST(=shortest set-up time) heuristic

A metal products manufacturer has contracted to ship metal braces euch day fo four customers. Each brace requires a different set-up on the rolling mill:

Rolling mill set-up times

Job	A	B	C	D
A	∞	3	4	5
B	3	∞	4	6
C	1	6	∞	2
D	5	4	∞*	∞

*Job C cannot follow job D, because of quality problems

SST-heuristic:

Step 1 starting arbitrarily by choosing one Job: A

Step 2 B has the smallest set-up time following A; ⇒ A-B

Step 3 C has the smallest set-up time of all the remaining jobs following B; ⇒ A-B-C

Step 4 D is the last remaining job; ⇒ A-B-C-D-A with a makespan of 3 + 4 + 2 + 5 =14

Operations Scheduling

- **A regret based Algorithm**
  - makespan must be at least as big as the n smallest elements
  - reduced matrix
    - row reduction
    - column reduction
    - sum of reduced costs = lower bound for TSP
  - **find reduced matrix!**

Job	A	B	C	D
A	∞	3	4	5
B	3	∞	4	6
C	1	6	∞	2
D	5	4	∞*	∞

Operations Scheduling

- **The reduced matrix has a zero in every row and column**
- **what happens if we do not choose j to follow i**
- **regret: lower bound on not choosing j to follow i**

Job	A	B	C	D
A	∞	0	0	2
B	0	∞	0	3
C	0	5	∞	0
D	1	0	∞*	∞

Operations Scheduling

Regret heuristic

Find the cycle sequence that minimizes the set-up time.

Set-up times

Job	1	2	3	4	5
1	∞	18	3	3	6
2	19	∞	9	10	5
3	9	18	∞	13	20
4	6	6	1	∞	2
5	17	1	13	17	∞

Solution: TSP model – regret heuristic

Step 0 C(max) = 0 and L = 1
Step 1 Reduce the matrix:

Reduced matrix

Job	1	2	3	4	5	Min
1	∞	15	0	0	6	3
2	14	∞	4	5	0	5
3	0	9	∞	4	11	9
4	5	5	0	∞	1	1
5	16	0	12	16	∞	1
						19

Operations Scheduling

Step 2 Calculate the regret

Job	1	2	3	4	5	Min
1	∞	15	0(0)	0(4)	6	3
2	14	∞	4	5	0(5)	5
3	0(9)	9	∞	4	11	9
4	5	5	0(1)	∞	1	1
5	16	0(17)	12	16	∞	1
						19

New matrix

Job	1	3	4	5
1	∞	0	0	3
2	14	4	5	∞
3	0	∞	4	11
4	5	0	∞	1

Step 3 Choose the largest regret : 17
Step 4 Assign a job pair: Job 2 immediately follows job 5 (5-2)
L = 1+1;
We prohibit 2-5

Operations Scheduling

Step1 Reduce the matrix

C_{max}=19+4+1=24

Reduced Matrix

Job	1	3	4	5
1	∞	0	0	2
2	10	0	1	∞
3	0	∞	4	10
4	5	0	∞	0

Step 2 Calculate the regret

Job	1	3	4	5
1	∞	0(0)	0(1)	2
2	10	0(1)	1	∞
3	0(9)	∞	4	10
4	5	0(0)	∞	0(2)

Step 3 Choose the largest regret: 9
Step 4 Assign a job pair: 3-1
Prohibit 1-3

Step 1 Reduce the matrix: not possible

Matrix

Job	3	4	5
1	∞	0	2
2	0	1	∞
4	0	∞	0

Operations Scheduling

Step 2 Calculate regret

Job	3	4	5
1	∞	0(3)	2
2	0(1)	1	∞
4	0(0)	∞	0(2)

Step 3 Choose the largest regret: 3
Step 4 Assign job pair : 1-4; partial sequence: 5-2, 3-1-4
Prohibit 4-1 and 4-3 (to keep 3-1-4-3 from being chosen)

Final Matrix

Job	3	5
2	0	∞
4	∞	0

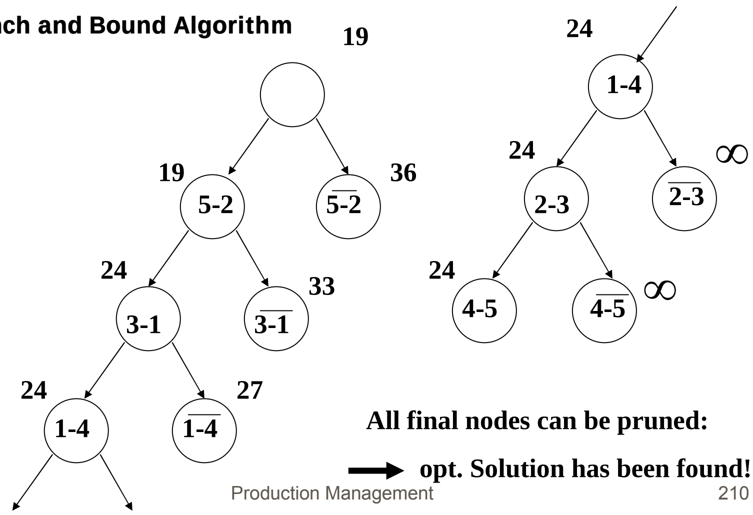
choose 2-3 and 4-5
-> sequence 3-1-4-5-2
the total set-up time is 24

Operations Scheduling

- **Branch and Bound Algorithm**
  - 1. Using the regret heuristic construct a (sub-)tree where each node represents the decision to let j follow i ($i - j$) or to prohibit that j follows i ($\overline{i-j}$)
  - 2. For each node a lower bound for the makespan is inferred from the regret heuristic
  - 3. Once a solution is obtained from the regret heuristic this is an upper bound for the optimal makespan. All nodes where the lower bound is above that level are pruned.
  - 4. If all but one final node are pruned (or no non-pruned node can be further branched) this final node gives the optimal solution.
  - 5. If 4. does not hold start again with 1. at one of nodes which are not pruned and can still be branched.

Operations Scheduling

- **Branch and Bound Algorithm**



All final nodes can be pruned:

→ **opt. Solution has been found!**

Operations Scheduling

- **Single-Machine Search Methods**
  - Neighborhood Search
  - Simulated Annealing
  - Ant System
  - Tabu Search
  - ...
- **Neighborhood Search**
  - seed
  - Neighborhood
  - any heuristic can be used to produce an initial sequence

Operations Scheduling

- adjacent pairwise interchange (API):
  - n-1 neighbors
  - 1-2-3-4-*5-6*-7-8-9
  - 1-2-3-4-*6-5*-7-8-9
- Pairwise interchange (PI):
  - n(n-1)/2 neighbors
  - 1-2-8-4-5-6-7-3-9
- Insertion (INS)
  - $(n-1)^2$ neighbors
  - 1-2-3-7-4-5-6-8-9
- Evaluation function
- Update function

Operations Scheduling

Neighborhood search

Consider the following single- machine tardiness problem;
Use the EDD sequence as the initial seed with an API neighborhood;

Data for neighborhood search

Job	1	2	3	4	5	6
Processing time	10	3	16	8	4	10
Due-date	15	16	24	30	35	37

- Step 1 Construct the EDD sequence and evaluate its total tardiness
  Set i = 1 and j = 2

- The EDD sequence S*: 1-2-3-4-5-6; tardiness-vector (0, 0 ,5 , 7 , 6, 14)

Operations Scheduling

- Step 2 Swap the jobs in the *i*-th and *j*-th position in S*; the sequence is S' with tardiness T'. If T' < T, go to step4

- Step 3 j = j +1: If j >n: go to step 5. Otherwise, i = j-1and go to step 2;

- Step 4 Replace S* with S'; i = 1, j = 2; go to step 2

- Step 5 Stop; S* is a local optimal sequence.

Operations Scheduling

Neighborhood search solution

Jobs i	j	Schedule 1	2	3	4	5	6	Tardiness
i	j	1	2	3	4	5	6	32
1	2	2	1	3	4	5	6	32
2	3	1	3	2	4	5	6	42
3	4	1	2	4	3	5	6	33
4	5	1	2	3	5	4	6	30
1	2	2	1	3	5	4	6	30
2	3	1	3	2	5	4	6	40
3	5	1	2	5	3	4	6	34
5	4	1	2	3	4	5	6	32
4	6	1	2	3	5	6	4	32

Operations Scheduling

- **Single machine results**
  - Flowtime - SPT (E)
  - Lateness - SPT (E)
  - Weighted Flowtime -WSPT (E)
  - Maximal Tardiness (Lateness) - EDD (E)
  - Nb. Of tardy jobs - Hodgson (E)
  - weighted nb. Of tardy jobs - modified Hodgson (H)
  - No jobs tardy/flowtime - modified SPT (E)
  - Tardiness - R&M (H)
  - weighted Tardiness - R&M (H)
  - makespan with non-zero release time and tails - Schrage (H)
  - Sequence dependent - SST (H), regret (H), B&B (E)

Operations Scheduling

- **Parallel Machines**
  - Scheduling decisions:
    - which machine processes the job
    - in what order
  - **List Schedule**
    - to create a schedule, assign the job on the list to the machine with the smallest amount of work assigned.
    - **Step 0**. Let $H_i=0$, $i=1,2,\dots,m$ be the assigned workload on machine i, $L=([1],[2],\dots,[n])$ the ordered list sequence, $C_j=0$, $j=1,2,\dots,n$, and $k=1$
    - **Step 1**. Let $j^*=L_k$ and $H_{i^*}=\min_{i=1,m}\{Hi\}$;
      Assign job j* to be processed on machine i*, $C_{j^*}=H_{i^*}+p_{j^*}$, $H_{i^*}=H_{i^*}+p_{j^*}$
    - **Step 2**. Set k=k+1, if k>n,stop. Otherwise go to step 1.

Operations Scheduling

Minimizing flowtime on parallel processors

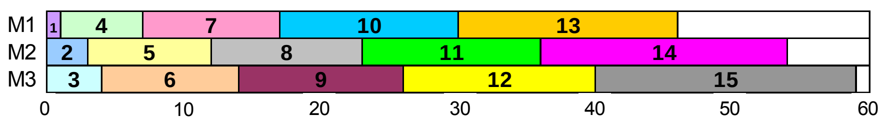
Consider a facility with 3 identical machines and 15 jobs that need to be done as soon as possible;
Processing times(after SPT):

Job	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Time	1	2	4	6	9	10	10	11	12	13	13	14	16	18	19

Optimal schedule:

Machine 1			Machine 2			Machine 3		
j	p(j)	C(j)	j	p(j)	C(j)	j	p(j)	C(j)
1	1	1	2	3	3	3	4	4
4	6	7	5	9	12	6	10	14
7	10	17	8	11	23	9	12	26
10	13	30	11	13	36	12	14	40
13	16	46	14	18	54	15	19	59

Total flowtime = 372



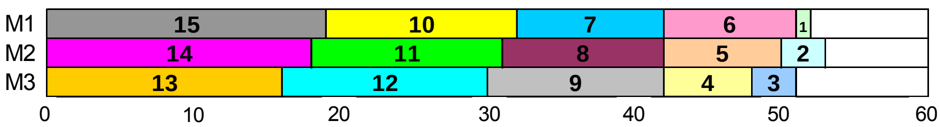
Operations Scheduling

Minimize the makespan

Use a longest processing time (LPT) first list;
Assign the next job on the list to the machine with the least total processing time assigned.

Optimal schedule:

Machine 1			Machine 2			Machine 3		
j	p(j)	C(j)	j	p(j)	C(j)	j	p(j)	C(j)
15	19	19	14	18	18	13	16	16
10	13	32	11	13	31	12	14	30
7	10	42	8	11	42	9	12	42
6	10	52	5	9	51	4	6	48
1	1	53	2	3	54	3	4	52



Operations Scheduling

- **Flow shops**
  - all jobs are processed in the same order
  - two machine makespan model: Johnson's Algorithm
  - Bound on makespan:

$$C^*_{\max} \geq \max\left\{\left(\min_{i=1,n} p_{i2} + \sum_{i=1}^{n} p_{i1}\right), \left(\min_{i=1,n} p_{i1} + \sum_{i=1}^{n} p_{i2}\right)\right\}$$

  - Formulate Johnson's Algorithm
  - For 2-machine Flow shops the optimal schedule is a **Permutation Schedule,** i.e. the job sequence is the same on every machine

Operations Scheduling

- **Makespan with more than two machines**
  - Johnson's algorithm will work in special cases, e.g. three machine problem where the second machine is dominated:

$$p_{i2} \leq \max(\min p_{i1}, \min p_{i3})$$

  - Formulate an artificial two machine problem with

$$p'_{i1} = p_{i1} + p_{i2} \quad \text{and} \quad p'_{i2} = p_{i2} + p_{i3}$$

and solve it using the Johnson algorithm gives the optimal solution for the three machine problem

Operations Scheduling

- **Heuristics for the m-machine problem**
  - Cambell, Dudek and Smith (1970)
  - convert a m-machine problem into a two machine problem
  - how?

$$p'_{i1} = \sum_{j=1}^{k} p_{ij} \quad \text{and} \quad p'_{i2} = \sum_{j=l}^{m} p_{ij}$$

  - Start with:k=1 and l=m; then k=2 and l=m-1; until: k=m-1 and l=2
  - m-1 schedules are generated
  - Use the best of these m-1 schedules

Operations Scheduling

Flow-shop heuristics

Processing data:

Job	1	2	3	4	5
M1	1	10	17	12	11
M2	13	12	9	17	3
M3	6	18	13	2	5
M4	2	18	4	6	16

Use the CDS heuristic to solve this problem.

(1) i.) Use the Johnson's algorithm only for M1 and M4:

Job	1	2	3	4	5
M1	1	10	17	12	11
M4	2	18	4	6	16
[j]	1	2	5	4	3

1-2-5-4-3, C_{max} = 88

Next combine M1 and M2 to pseudomachine 1 and M3 and M4 to pseudomachine 2.

Job	1	2	3	4	5
PM1	14	22	26	29	14
PM2	8	36	17	8	21
[j]	4	2	3	5	1

5-2-3-1-4, C_{max} = 85

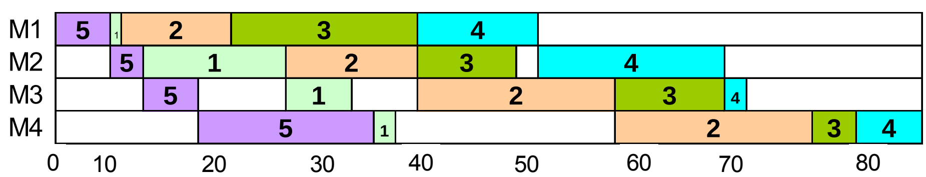
Finally combine M1, M2 and M3 to pseudomachine 1 and M2, M3 and M4 to pseudomachine 2.

Job	1	2	3	4	5
PM1	20	40	39	31	19
PM2	21	48	26	25	24
[j]	2	3	4	5	1

5-1-2-3-4, C_{max} = 85

Operations Scheduling

⌘ **Gantt Chart for the CDS schedule**



Operations Scheduling

- **Gupta – Heuristic**
  - Gupta (1972)
  - exact for 2-machine problem and 3-machine problem, where the 2nd machine is dominated

$$e_i = \begin{cases} 1 & \text{if } p_{i1} < p_{im} \\ -1 & \text{if } p_{i1} \geq p_{im} \end{cases} \qquad s_i = \frac{e_i}{\min\limits_{k=1,\ldots,m-1} \{p_{i,k} + p_{i,k+1}\}}$$

  - Sorting jobs with nonincreasing s_i ($s_{[1]} \geq s_{[2]} \geq \ldots \geq s_{[n]}$)

Job	p1+p2	p2+p3	p3+p4	min	ei	si	[i]
1	14	19	8	8	1	0.12	1
2	22	30	36	22	1	0.05	3
3	26	22	17	17	-1	-0.06	4
4	29	19	8	8	-1	-0.12	5
5	14	8	21	8	1	0.12	2

Operations Scheduling

- **Branch and Bound Approaches**
  - machine based bounds
  - job based bounds
  - three machines
    - H_j=current completion time of the last job scheduled on machine j
    - U=set of unscheduled jobs
    - makespan on machine 1 must be at least:

$$C^*_{\max} \geq H_1 + \sum_{i \in U} p_{i1} + \min_{i \in U}\{p_{i2} + p_{i3}\}$$

    - machine2:

$$C^*_{\max} \geq \max\left\{\left(H_1 + \min_{i \in U}\{p_{i1}\}\right), H_2\right\} + \sum_{i \in U} p_{i2} + \min_{i \in U}\{p_{i3}\}$$

Operations Scheduling

- Machine 3:

$$C^*_{\max} \geq \max\left\{\left(H_1 + \min_{i\in U}\{p_{i1} + p_{i2}\}\right), \left[H_2 + \min_{i\in U}\{p_{i2}\}\right], H_3\right\} + \sum_{i\in U} p_{i3}$$

- job oriented bounds:

$$C_{\max} \geq H_1 + \max_{i\in U}\left\{\sum_{j=1}^{m} p_{ij} + \sum_{k\in U, k\neq i} \min\{p_{k1}, p_{k3}\}\right\}$$

Operations Scheduling

- **B&B algorithm for minimizing makespan in multi-machine Flow Shops**
- **1.** Create an initial incumbent solution, e.g. CDS heuristic
  ⟶ upper bound
- 2. Starting at t=0 with a root node; branch the tree by generating a node for each schedulable jobs.
- 3. In each node calculate the lower bounds and prune the node if at least one exceeds the upper bound.
- 4. If a non-pruned final node exists at the lowest level take the
- corresponding solution as new incumbent, update the upper bound and do the corresponding pruning.
- 5. If all final nodes are pruned current incumbent is the optimal solution, otherwise branch at the node with the lowest lower bound and goto 3.

Operations Scheduling

Makespan permutation schedule for a three-machine flow-shop

Processing data:

Machine j	Job i 1	2	3	4	5
1	1	10	17	12	11
2	13	12	9	17	3
3	6	18	13	2	5

Solution:
Start with CDS algorithm: sequence: 1-2-3-4-5, $C_{max} = 65$

Initial lower bound:
M1: $C_{max}^* >= H_1 + (p_{11} + p_{21} + p_{31} + p_{41} + p_{51})$
$+ \min\{ p_{12} + p_{13}, p_{22} + p_{23}, p_{32} + p_{33}, p_{42} + p_{43}, p_{52} + p_{53} \}$
$= 0 + (1 + 10 + 17 + 12 + 11) + \min\{19, 30, 22, 19, 8\} = 51 + 8 = 59$

M2: $C_{max}^* >= \max\{[H_1 + \min\{p_{11} + p_{21} + p_{31} + p_{41} + p_{51}\}], H_2\}$
$+ (p_{12} + p_{22} + p_{32} + p_{42} + p_{52}) + \min \{ p_{13}, p_{23}, p_{33}, p_{43}, p_{53}\}$
$= \max\{[0 + \min\{1, 10, 17, 12, 11\}], 0\}$
$+ (13 + 12 + 9 + 17 + 3) + \min\{6, 18, 13, 2, 5\}$
$= 1 + 54 + 2 = 57$

Operations Scheduling

M3: $C_{max}^* >= \max\{[H_1 + \min\{ p_{11} + p_{12}, p_{21} + p_{22}, p_{31} + p_{32}, p_{41} + p_{42}, p_{51} + p_{52} \}],$
$[H_2 + \min\{p_{12} + p_{22} + p_{32} + p_{42} + p_{52}\}], H_3\} + (p_{13} + p_{23} + p_{33} + p_{43} + p_{53})$
$= \max\{[0 + \min\{14, 22, 26, 29, 14\}],$
$[0 + \min\{13, 12, 9, 17, 3\}], 0\} + (6 + 18 + 13 + 2 + 5)$
$= \max\{14, 3, 0\} + 44 = 58$

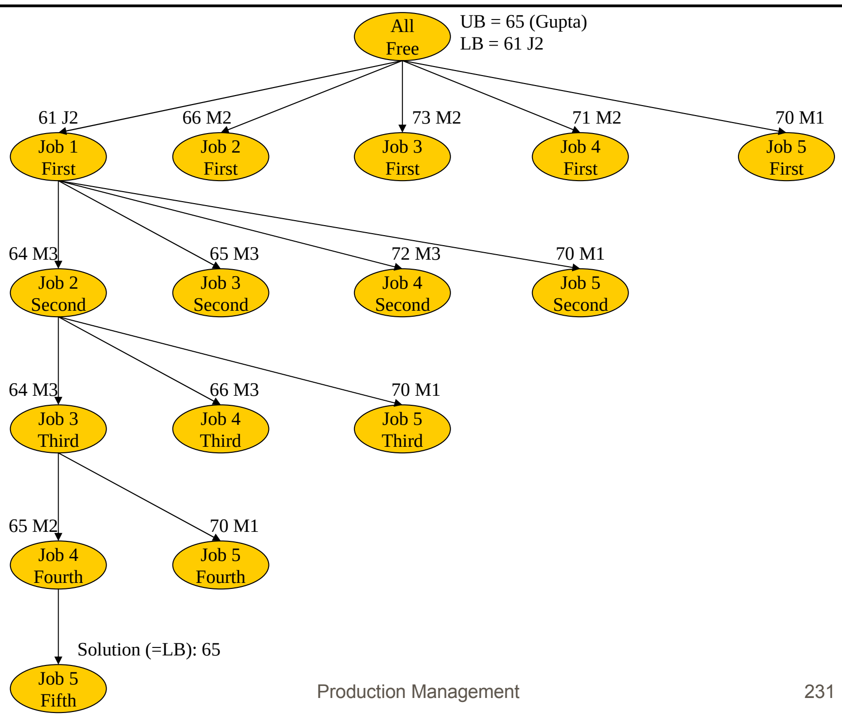
Job-based bounds are the following:

$$C_{\max} \geq H_1 + \sum_{j=1}^{3} p_{1j} + \sum_{k \in \{2,3,4,5\}} \min\{p_{k1}, p_{k3}\}$$

J1: $C_{max}^* >= H_1 + (p_{11} + p_{12} + p_{13})$
$+(\min\{ p_{21}, p_{23} \} + \min\{p_{31}, p_{33} \} + \min \{p_{41}, p_{43} \} + \min\{ p_{51}, p_{53} \})$
$= 0 + (1 + 13 + 6) + (\min\{10, 18\} + \min\{17, 13\} + \min\{12,2\} + \min\{11,5\})$
$= 0 + 20 + (10 + 13 + 2 + 5) = 50$
Similarly, we have

J2: $C_{max}^* >= 61$, J3: $C_{max}^* >= 57$, J4: $C_{max}^* >= 60$, J5: $C_{max}^* >= 45$

LB: 61, UB: 65



Operations Scheduling

1st level:J2 at first place: $H_1 = 10$, $H_2 = 22$, $H_3 = 40$

$U = \{1, 3, 4, 5\}$

M1: $C_{max}^* >= 59$

M2: $C_{max}^* >= 66$, which is greater than the upper bound; thus we fathom the node;

J3, J4 and J5 at first place: we can fathom all of them;

2nd level: Consider Job 3: $H_1 = 18$, $H_2 = 27$, $H_3 = 40$, $U = \{2, 4, 5\}$

M1: $C_{max}^* >= 59$

M2: $C_{max}^* >= 62$

M3: $C_{max}^* >= 65$, so we fathom the job; only job 2 remains unfathomed;

3rd level: Job 3: $H_1 = 28$, $H_2 = 37$, $H_3 = 57$, $U = \{4, 5\}$

M1: $C_{max}^* >= 59$

M2: $C_{max}^* >= 61$

M3: $C_{max}^* >= 64$

Machine-bounds did not fathom the node; so we have to calculate job-based bounds:

J4: $C_{max}^* >= 64$

J5: $C_{max}^* >= 49$

→ best bound = 64; thus create nodes for J4 and J5

4th level: nodes J4 and J5 of level 3 will be fathomed; thus the algorithm is complete:
1-2-3-4-5 with a makespan of 65;

Operations Scheduling

- **Job Shops**
  - different routings for different jobs
  - precedence constraints
  - $(n!)^m$ possible schedules

Operations Scheduling

- **Two machine job shops**
  - Jackson (1956)
  - minimize makespan
    - Machine A: {AB}, {A}, {BA}
    - Machine B: {BA}, {B}, {A,B}

Jackson's algorithm

Job	1	2	3	4	5	6	7	8	9	10
Route	BA	AB	BA	B	A	AB	B	BA	BA	AB
p(i)1	3	1	11	0	3	9	0	8	13	2
p(i)2	8	10	13	1	0	8	6	10	6	6

Find a schedule that would finish all jobs as soon as possible!

Solution:
{A} = {5}, {B} = {4,7}, {AB} = {2, 6, 10} and {BA} = {1, 3, 8, 9}

Operations Scheduling

Johnson's algorithm for {AB}:

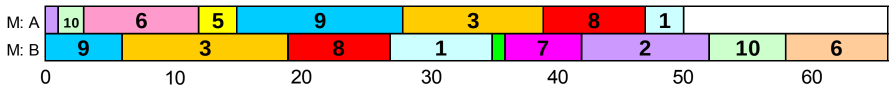
Job	2	10	6
p(i)1	1	2	9
p(i)2	10	6	8

Johnson's algorithm (reversed) for {BA}:

Job	9	3	8	1
p(i)1	13	11	8	3
p(i)2	6	13	10	8

sequence for A: 2-10-6-5-9-3-8-1
sequence for B: 9-3-8-1-4-7-2-10-6

makespan:67



Operations Scheduling

- **Dispatching**
  - job shop scheduling
  - dispatching rules
  - Basic idea:
    - schedule an operation of a job as soon as possible
    - if more than one job is waiting to be processed by the same machine schedule the one with best priority
  - Define:
    - A= set of idle machines
    - J_k= the index of the last job scheduled on machine k
    - U_k= the set of jobs that can be processed on machine k
    - H_k = the completion time of the job currently processed on machine k
    - u_{it} = the priority of job i at time t

Operations Scheduling

- Step 0. Initialize: t=0; H_k=0,k=1,2,...,m; A={1,2,...,m}; U_k={i|operation 1 of i is on machine k, i=1,2,...,n}; s_{ij}=c_{ij}=0. Go to step 4.
- Step 1. Increment t;

  Let

  $$t = \min_{k=1,m;k\notin A} H_k, \text{and } K = \{k \mid H_k = t\}$$

- Step 2. Find the job or jobs that complete at time t and the machine released. Set A = A∪K.
- Step 3. Determine the jobs ready to be scheduled on each machine; Let U_k={i|job i uses machine k and all operations of job i before machine k are completed}, k=1,2,...,m. If U_k=0 for k=1,2,...,m,Stop. If U_k=0 for k∈A, go to Step 1.

Operations Scheduling

- Step 4. For each idle machine try to schedule a job; for each k ∈ A with Uk≠0,

  $$\text{let } i^* \text{ be the job with the best priority} : u_{i^*t} = \min_{i \in U_k} u_{it}$$

  Schedule job i* on machine k

  $$\text{Set } J_k = i, s_{i^*k} = t, c_{i^*k} = t + p_{i^*j(k)}, H_k = c_{i^*k}$$

  Remove the scheduled job from U_k

  $$U_k \leftarrow U_k - \{i\}$$

  and the machine from A

  $$A \leftarrow A - \{k\}$$

  Go to Step 1

Operations Scheduling

- **Many priority measures possible:**
  - SPT
  - FCFS
  - MWKR (most work remaining)
  - EDD
  - EDD/OP
  - SLACK, SLACK/OP
  - Critical ratio: slack/remaining time
  - ...

Operations Scheduling

Quick Closures: job-shop dispatch heuristic

Quick Closure has four machines in the shop: (1) brake, (2) emboss, (3) drill, (4) mill; The shop has currently orders for six different parts, which use all the four machines, but in a different order.
Processing time:

	Operation			
Job	1	2	3	4
1	6/1	8/2	13/3	5/4
2	4/1	1/2	4/3	3/4
3	3/4	8/2	6/1	4/3
4	5/2	10/1	15/3	4/4
5	3/1	4/2	6/4	4/3
6	4/3	2/1	4/2	5/4

Finish all six parts as soon as possible!

Solution: We use a dispatch procedure with MWKR as the priority.

Operations Scheduling

Step1 t = 0, $H_1 = H_2 = H_3 = H_4 = 0$, A = {1, 2, 3, 4}, U_1 = {1, 2, 5}, U_2 = {4}, U_3 = {6}, U_4 = {3}; s_{ij} = c_{ij} = 0, i = 1, 2, 3, 4, 5, 6; and j = 1, 2,3, 4; Go to step 4

Step 4 u_{10} = -(6+8+13+5) = -32, u_{20} = -12, u_{50} = -17; thus s_{11} = 0, c_{11} = 0 + 6 = 6, H_1 = 6.
Remove job 1 from U_1, U_1 = {2, 5} and machine 1 from A, A = {2, 3, 4}.
Set k = 2; there is only one job in U_2 so we schedule it on machine 2; i* = 4, s_{41} = 0, c_{41} = 5, H_2= 5, U_2 = { }, and A = {3, 4}.
We schedule J6 and J3 on M3 and M4 (tab: t = 0 row). Go to step 1.

Step 1 t = $\min_{k=1,m:k \varepsilon A} H_k$ = min{6, 5, 4, 3} = 3, and K = {k\\$H_k$ = 3} = {4}; H_k *min* is bold in the table;

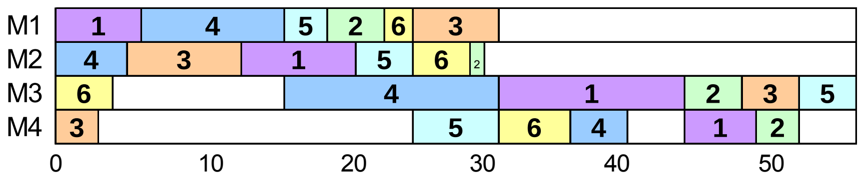
Step 2 J3 completes at time 3 on M4, so i^3 = {i\\J_k = i, k ε K} ={3}, K = {4},
and A = {} U {4} = {4}, (tab: t = 3 row)

Step 3 U1 = {2, 5}, U2 = {3}, U3 = U4 = { }; Since no jobs are waiting for M4, no jobs can be scheduled to start at time 3; go to step 1 etc.

Operations Scheduling

t	it	K	A	U1	U2	U3	U4	H1	H2	H3	H4
0	-	-	1,2,3,4	**1**,2,5	**4**	**6**	**3**	6	5	4	3
3	3	4	4	2,5	3			6	5	**4**	
4	6	3	3,4	2,5,6	3			6	**5**		
5	4	2	2,3,4	2,4,5,6	**3**			**6**	13		
6	1	1	1,3,4	2,**4**,5,6	1			16	**13**		
13	3	2	2,3,4	2,3,5,6	**1**			**16**	21		
16	4	1	1,3,4	2,3,**5**,6		**4**		**19**	21	31	
19	5	1	1,4	**2**,3,6	5			23	**21**	31	
21	1	2	2,4	3,6	**5**	1		**23**	25	31	
23	2	1	1,4	3,**6**	2	1		**25**	**25**	31	
25	6,5	1,2	1,2,4	**3**	2,**6**	1	**5**	31	**29**	31	31
29	6	2	2		**2**	1	6	31	**30**	31	31
30	2	2	2			1,2	6	**31**		**31**	**31**
31	3,4	1,3	1,2,3			**1**,2,3	4,**6**			44	36
36	6	4	1,2,4			2,3,5	**4**			44	**40**
40	4	4	1,2,4			2,3,5				**44**	
44	1	3	1,2,3,4			**2**,3,5	**1**			**48**	49
48	2	3	1,2,3			**3**,5	2			52	**49**
49	1	4	1,2,4			5	**2**			**52**	**52**
52	3,2	3,4	1,2,3,4			**5**				**56**	
56	5	3	1,2,3,4								

Operations Scheduling



Chapter 10

Section 5.5:
Bottleneck Scheduling

Operations Scheduling

- **Shifting Bottleneck Procedure**
- **heuristic to minimize makespan for multiple machine job shops**
- **Main idea:**
  1. for each job on each machine calculate the minimal amount of time needed before and after the processing of this job
     → generates minimal makespan problem with release times and tails
  2. for each machine solve this problem for each machine (e.g. Schrage heuristic) and determine the machine with the maximal makespan (bottleneck machine)
  3. Fix the found sequence on the bottleneck machine, update release times and tails on the remaining machines and repeat 2. for the remaining machines until schedules for all machines have been determined

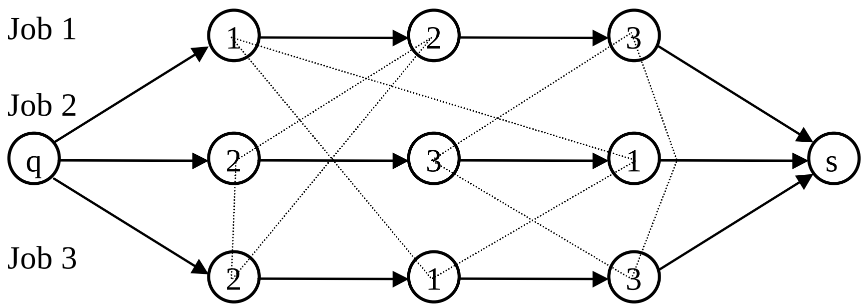
Operations Scheduling

- **Shifting Bottleneck Procedure Example:**
- 3 machines (M1, M2, M3), 3 jobs (1,2,3)
- **Job routings:** 1: M1-M2-M3
  2: M2-M3-M1
  3: M2-M1-M3
- **Processing times:**

p_{ik}	M1	M2	M3
1	3	3	2
2	3	2	3
3	3	4	1

Operations Scheduling

- **Machine-Flow-Graph:**



Operations Scheduling

- **Problems with release times and tails for each machine:**
- **M1:**

	1	2	3
r_j	0	5	4
p_j	3	3	3
n_j	5	0	1

M2:

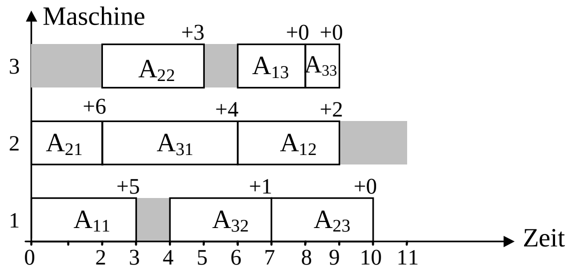
	1	2	3
r_j	3	0	0
p_j	3	2	4
n_j	2	6	4

M3:

	1	2	3
r_j	6	2	7
p_j	2	3	1
n_j	0	3	0

Operations Scheduling

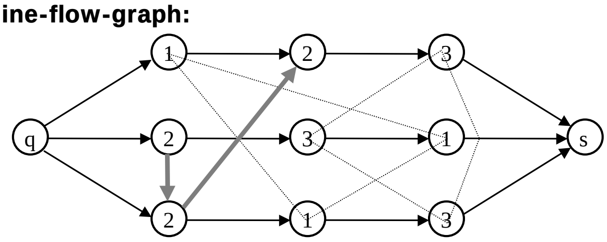
- Schrage heuristic gives the following solutions for the three machines:



- ➡ machine 2 is bottleneck with $C_2 = 11$
- ➡ fix sequence on machine 2

Operations Scheduling

- **Machine-flow-graph:**



- **Update release times and tails on M1 and M3:**
- **M1:**

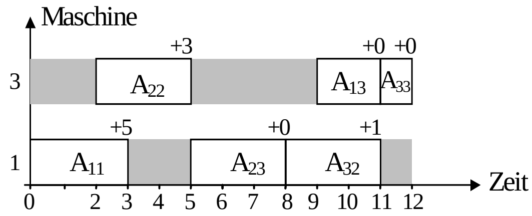
	1	2	3
r_j	0	5	6
p_j	3	3	3
n_j	5	0	1

M3:

	1	2	3
r_j	9	2	9
p_j	2	3	1
n_j	0	3	0

Operations Scheduling

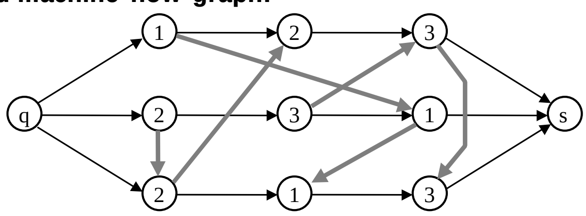
⌘ **Schrage heuristic for M1, M3:**



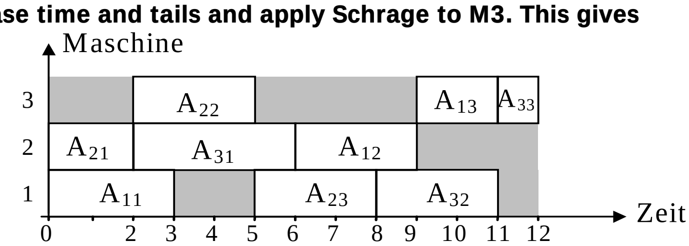
⌘ → both machines could be considered the bottleneck with C=12, fix sequence on M1

Operations Scheduling

⌘ **Updated machine-flow-graph:**



⌘ **update relase time and tails and apply Schrage to M3. This gives**



⌘ **with C_{max}=12**

Operations Scheduling

- **Finite Capacity Scheduling**
  - MRP systems generally assume constant lead times, ignore setups
  - MRP plans might be unrealistic
  - Traditionally problem hiden by inventory and excess capacity
  - Reducing Inventory and capacity makes finite capacity scheduling crucial
  - Computer-assisted finitie capacity scheduling systems rather than manual scheduling by foreman

Operations Scheduling

- **Work to do: 8.3abcde, 8.4, 8.5, 8.6, 8.10, 8.14, 8.16, 8.18 (with the following due dates: 42, 50, 12, 63, 23, 34, 36, 42, 54, 32) 8.30ab, 8.32abc, 8.36ab, 8.43, 8.44, 8.49ab, 8.51ab, 8.56, 8.57 (apply shifting bottleneck procedure)**
- **Minicase: Ilana Designs**