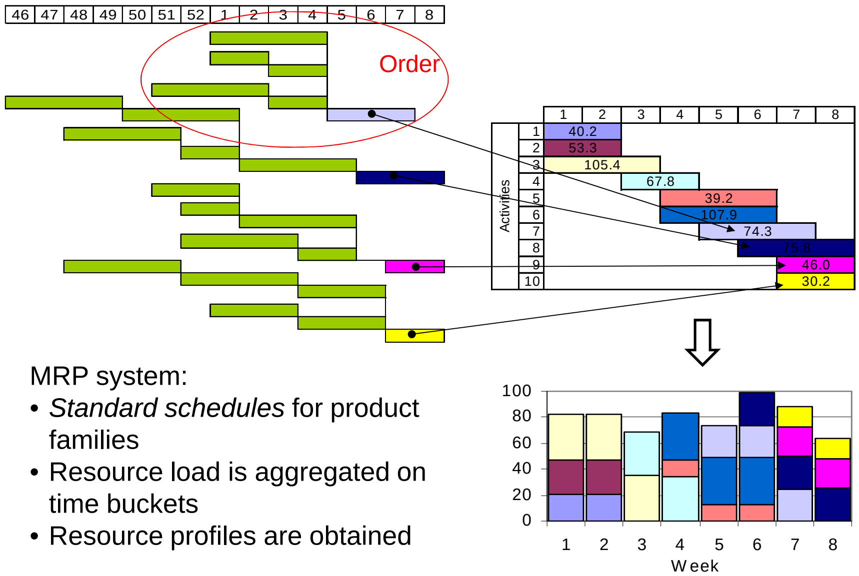
Aggregate planning in MTO-production

- In MTO-production batch formation is not usually a relevant problem, because product variation is large, which in fact is the reason for MTO-production in the firs place
- Product structures and production processes are typically complicated
- Orders appear randomly and resource load is fluctuating
- Pressure for short delivery times is severe
- Leveling of resource load is important for economical reasons
- Levelling affects resource utilization rate and need of capacity and reduces marginal cost due to overtime work, and emergency subcontracting
- Aggregate planning systems appear as MRP (Manufacturing Resource Planning) functions in ERP (Enterprise Resource Planning systems

Aggregate planning in MTO-production

- Purpose of aggregate planning is to schedule orders based on experience, present status and forecasts so that
  - The ability to take new orders is good
  - Resource loading matches capacity
  - Delays are minimized
  - Work in process is kept at a low level
  - Need for rescheduling is kept minimal
- MTO aggregate planning system's basic inputs:
  - Order release dates
  - Order due dates
  - Material delivery times
  - Resource capacities
  - Orders' resource loading (work contents)
  - Process precedence requirements (process flow)

Aggregate planning in MTO-production



MRP system:

- *Standard schedules* for product families
- Resource load is aggregated on time buckets
- Resource profiles are obtained

Aggregate planning in MTO-production

- System user sees the status of the system when allocating new orders and knows basic specifications of the orders
- In addition an order forecast may be available
- Two extreme cases are *forward planning*, in which orders are scheduled as early as possible, and *backward planning*, in which orders are started as late as possible taking capacity constraints and release and due dates into consideration
- Slack and possibilities to affect timing are the bigger the bigger the difference between shortest possible delivery time and due date is and the more capacity is available
- In practice room for optimization exist, because only some orders are taken with minimum delivery time

MTO aggregate planning strategies

- **Forward planning** provides the best ability to take new orders, but work in process is maximized and need for rescheduling may be considerable
- On the other hand, reserving capacity for rush orders is important, because these orders are taken at best price and make your customers happy
- In **backward planning** orders may be lost for lack of capacity
- But, in MTO production customers often make changes to orders afterwards. These changes are the easier to make the later manufacturing takes place

- In practice one has to use sales forecasts and adjust timing tactics accordingly
- Rescheduling causes confusion, errors and generates indirect administrative costs

MTO aggregate planning optimization

Optimization criteria

- In the following models the main optimization criteria are resource levelling or peak load minimization
- The idea is that in practice production needs to deliver products as promised to customers and therefore due dates are constraints that can not be violated and capacity has to be flexible
- Tardiness minimization is also a common optimization criterion with capacity taken as a constraint
- Both am. criteria can be combined in multi-objective optimization, where penalties are determined to resource load limit and due date violations

Optimization model

- Because different kinds of orders load resources at different amounts and at different times in relation to finishing time, the problem is difficult to solve manually
- As a linear optimization model it works well

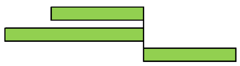
MTO aggregate planning – standard schedules

- Usually customer orders are handled as projects with standard templates for schedules for product families
- Work content is evenly distributed over allowed process step duration
- Resource loading is aggregated on each time bucket

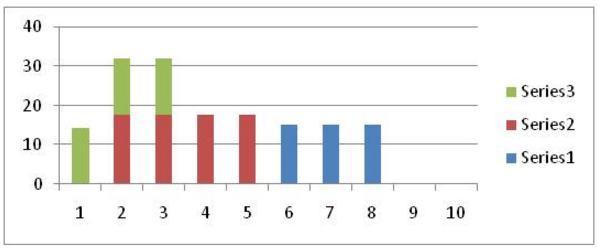
Schedule

Order #	Resource #	P	WC	1	2	3	4	5	6	7	8	9	10
1	1	3	45	0	0	0	0	0	15	15	15	0	0
1	2	2	67	0	0	0	0	0	0	33.5	33.5	0	0
1	3	2	23	0	0	0	0	0	0	0	0	12	11.5
2	1	4	70	0	17.5	17.5	17.5	17.5	0	0	0	0	0
2	2	2	34	0	0	0	17	17	0	0	0	0	0
2	3	3	89	0	0	0	0	0	29.7	29.7	29.7	0	0
3	1	3	43	14.3	14.3	14.3	0	0	0	0	0	0	0
3	2	1	35	0	0	35	0	0	0	0	0	0	0
3	3	2	43	0	0	0	21.5	21.5	0	0	0	0	0

Precedences:



Resource 1



P = Allowed duration (window) for process step
WC = Work Content

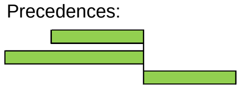
MTO aggregate planning – project control

- In project control method single process steps may be moved within timing and precedence constraints

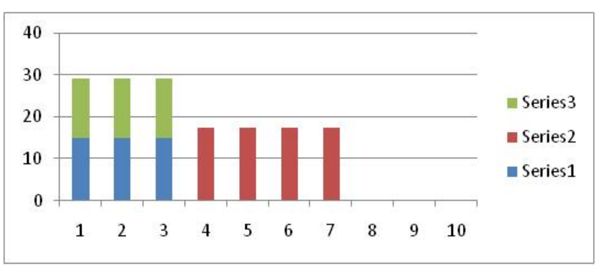
Schedule

Order #	Resource #	P	WC	1	2	3	4	5	6	7	8	9	10
1	1	3	45	15	15	15	0	0	0	0	0	0	0
1	2	2	67	0	0	0	33.5	33.5	0	0	0	0	0
1	3	2	23	0	0	0	0	0	11.5	11.5	0	0	0
2	1	4	70	0	0	0	17.5	17.5	17.5	17.5	0	0	0
2	2	2	34	17	17	0	0	0	0	0	0	0	0
2	3	3	89	0	0	0	0	0	0	0	29.7	29.7	29.7
3	1	3	43	14.3	14.3	14.3	0	0	0	0	0	0	0
3	2	1	35	0	0	35	0	0	0	0	0	0	0
3	3	2	43	0	0	0	21.5	21.5	0	0	0	0	0

Precedences:



Resource 1



MTO planning optimization – project control

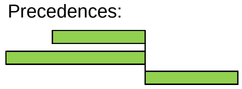
$\text{Min} \sum_{\forall k} f_k$	Minimization of sum of resource load peaks
s.t.	
$C_{ik} = \sum_{t=1}^{HZ} t c_{ikt}, \quad \forall i,k$	c_{ikt} = 1if job is finished at time t, otherwise 0 C_{ik} = Finishing time of job *i*
$\sum_{t=1}^{HZ} c_{ikt} = 1, \quad \forall i,k$	Jobs end once
$D_i \geq C_{ik} \geq C_{hk} + P_{ik}, \ \forall i, h \in U(i), k$	Precedences and due dates not violated *U(i)* is set containing steps preceding *i*
$C_{ik} - P_{ik} \geq A_{ik}, \quad \forall i,k$	Earliest allowed starting time A_{ik}
$R_{ikt} = WC_{ik} / P_{ik} \sum_{u=t}^{t+P_{ik}-1} c_{iku}, \quad \forall i,k$	Total work content is evenly distributed over all *t* in timing window R_{ikt}
$f_k - \sum_{i=1}^{I} R_{ikt} \geq 0, \quad \forall t,k$	Peak load

MTO planning optimization – free work content allocation within process step timing window

- Free work content allocation within process step timing window allows much better levelling than even allocation

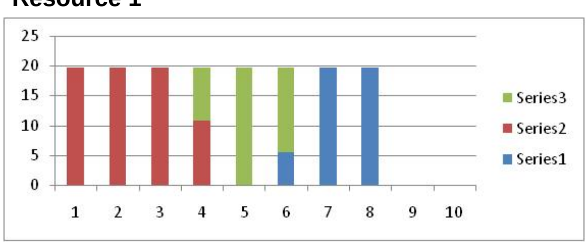
Schedule

Precedences:



Order #	Resource #	P	WC										
1	1	3	45	0	0	0	0	0	5.5	19.8	19.8	0	0
2	1	4	70	19.8	19.8	19.8	10.8	0	0	0	0	0	0
3	1	3	43	0	0	0	9.0	19.8	14.3	0	0	0	0
1	2	2	67	0	0	0	0	0	0	35.0	32.0	0	0
2	2	2	34	0	0	0	34.0	0	0	0	0	0	0
3	2	1	35	0	0	0	0	0	35.0	0	0	0	0
1	3	2	23	0	0	0	0	0	0	0	0	0	23.0
2	3	3	89	0	0	0	0	29.7	29.7	29.7	0	0	0
3	3	2	43	0	0	0	0	0	0	0	29.7	13.3	0

Resource 1



MTO planning optimization – free work content allocation within process step timing window

$$\text{Min} \sum_{\forall k} f_k$$

s.t.

$$C_{ik} = \sum_{t=1}^{HZ} t c_{ikt}, \quad \forall i$$

$$\sum_{t=1}^{HZ} c_{ikt} = 1, \quad \forall i,k$$

$$D_i \ge C_{ik} \ge C_{hk} + P_{ik}, \forall i, h \in U(i), k$$

$$C_{ik} - P_{ik} \ge A_{ik}, \quad \forall i,k$$

$$X_{ikt} = \sum_{u=t}^{t+P_{ik}-1} c_{iku}, \quad \forall i,t$$

X_{ikt} = 1, if t in allowed timing window, otherwise 0

$$R_{ikt} \le M X_{ikt}, \quad \forall i,k,t$$

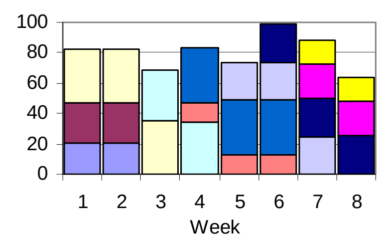
Work may only be allocated to timing window

$$\sum_{t=1}^{HZ} R_{ikt} = WC_{ik}, \quad \forall i,k$$

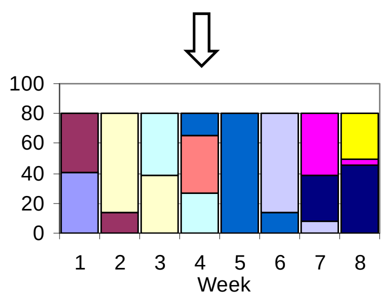
All work content must be allocated

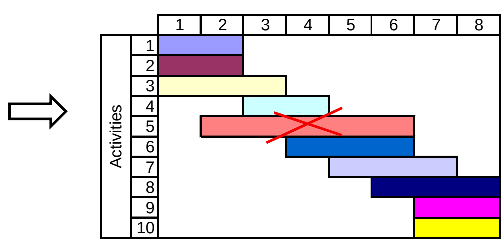
$$f_k - \sum_{i=1}^{I} R_{ikt} \ge 0, \quad \forall t,k$$

From aggregate planning to fine scheduling

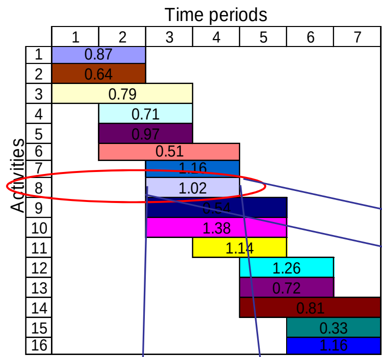


- A particularly good property for a schedule is “tiling” so that no step starts before and ends after another step
- With such timing work is never pre-empted





From aggregate planning to fine scheduling



- Such an aggregate schedule can always be transformed to a feasible detailed schedule for single machines
- For parallel machines this may not be possible, because jobs should be allocated to machines evenly