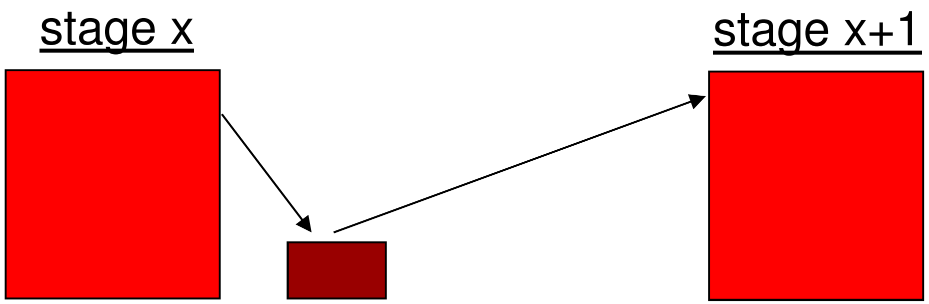
3 Lot-sizing problems

- A **lot size** is defined as the amount of a particular item that is ordered from the plant or a supplier or issued as a standard quantity to the production process,
- I.e., in what follows, we define the lot size as the number of items of one product to be continuously produced without preemption on the same machine
- As relevant costs we consider
  - the **lot size dependent setup costs** and additionally
  - the **lot size dependent inventory costs**.
- Note that there is always **a tradeoff** between these costs
  - The larger the chosen lot size is, the larger is the inventory and, consequently, the inventory costs
  - The smaller the chosen lot size is, the more batches have to be realized and, therefore, the more setup costs are increased
- In what follows, we consider different models computing efficient lot sizes
- These models can be mainly distinguished by their assumptions according to the dependencies between the scheduled products and the occurring demands

Open vs. closed production



Batch currently processed at stage x

- An **open production** is characterized by the fact that the items of the current batch that are already processed at stage x can be further processed at the subsequent stage in spite of the fact that the total batch is not completed
- In contrast to this, a **closed production** does not allow a simultaneous processing of one batch at two neighboring stages. Therefore, each item of a batch currently processed at stage x cannot be processed at the subsequent one before this batch is not completed

Model characteristics

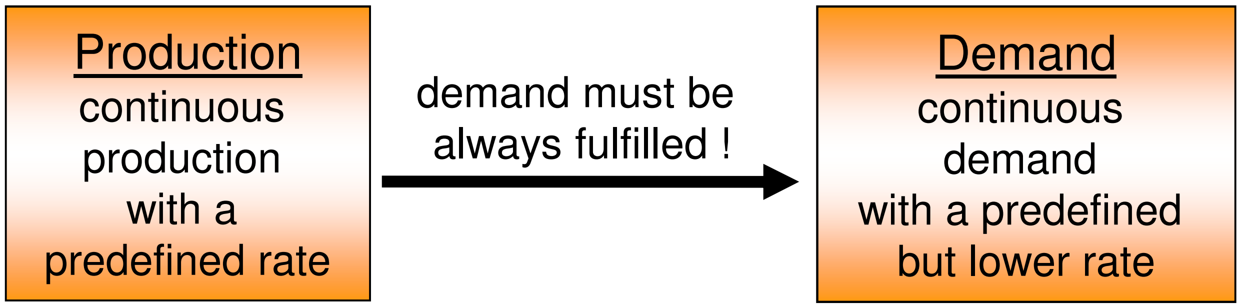
Degree of dependency between the scheduled products	Demand	
	stationary	**dynamic**
independent	EOQ model (Andler model)	SLULSP (=WW) SRP SPLP
dependent	ELSP	MCLSP MLCLSP

Outline of the chapter

1. The EOQ model
2. Extensions to multiple products
3. The SLULSP model (WW model)
    1. Model definition
    2. Dynamic programming approach
    3. Heuristic approaches
4. The CLSP model
    1. Problem definition
    2. The Dixon and Silver heuristic
5. The CLSPL model
    1. Basics
    2. Tightening the model
    3. Time-oriented decomposition heuristic

3.1 The EOQ model

- =**E**conomic **O**rder **Q**uantity model: Most simple model in literature
- Main assumptions of the model
  - Stationary demand
  - Continuous production with a predefined constant velocity
  - Continuous demand with a predefined constant velocity
  - The production always has to fulfill the demands of the subsequent distribution
  - Only one stage and one product are considered
  - Unlimited continuous planning horizon
  - No capacity constraints are modeled
  - Setup costs are independent of a given sequence
  - Therefore, the EOQ is a single item model where the optimal solution can be easily derived from



Parameters

c_I : Cost rate for inventory $\left[\dfrac{[\text{currency units}]}{[\text{quantity units}]\cdot[\text{planning horizon units}]}\right]$;

c_S : Cost rate for each setup $\left[\dfrac{[\text{currency units}]}{[\text{batch}]}\right]$;

x_T : Total production quantity to be produced in the considered planning horizon $\left[\dfrac{[\text{quantity units}]}{[\text{planning horizon units}]}\right]$;

v_D : Demand rate $\left[\dfrac{[\text{quantity units}]}{[\text{time units}]}\right]$;

v_P : Production rate $\left[\dfrac{[\text{quantity units}]}{[\text{time units}]}\right]$;

We assume: $v_D < v_P$;

t_S : Time necessary for the sale of a complete batch of size x $\left[\dfrac{[\text{time units}]}{[\text{batch}]}\right]$ $\left(\text{i.e. } t_S = \dfrac{x}{v_D}\right)$;

t_P : Time necessary for the production of a complete batch of size x $\left[\dfrac{[\text{time units}]}{[\text{batch}]}\right]$ $\left(\text{i.e. } t_P = \dfrac{x}{v_P}\right)$;

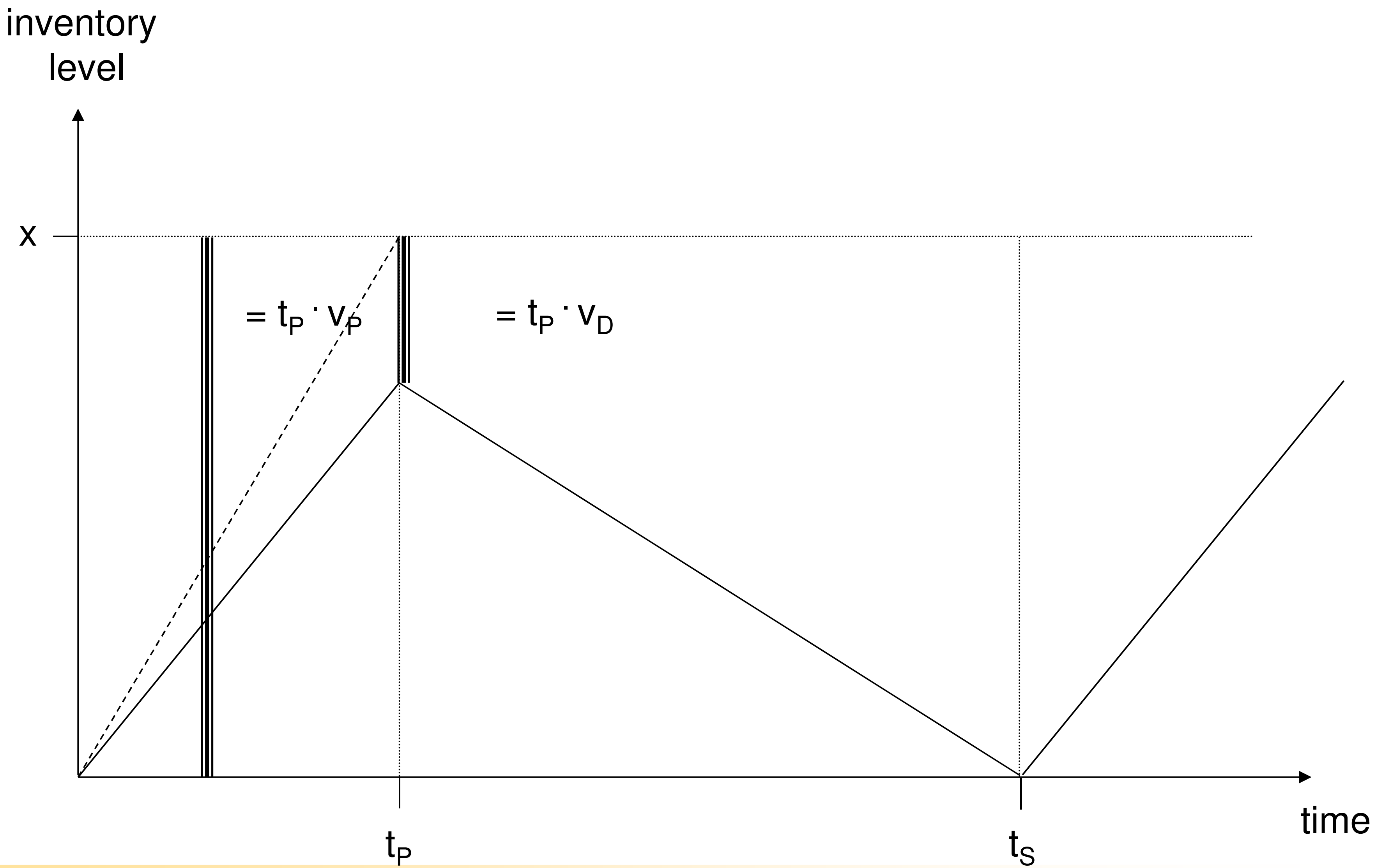
Sought :

x: Lot size $\left[\dfrac{[\text{quantity units}]}{[\text{batch}]}\right]$;

Solution of the model

- In order to derive the optimal lot size, we first have to define the cost function computing the **total lot size dependent costs**
- In order to do so, we need an additional function telling us what proportion of the used lot size is **on the average on stock** during the total planning horizon
- Therefore, we analyze subsequently the inventory level and generate a **function Øl(x) defining the average inventory level** if the lot size x is used during the execution of the production process
- In this connection, we have to distinguish between open and closed production processes

Inventory (open production)



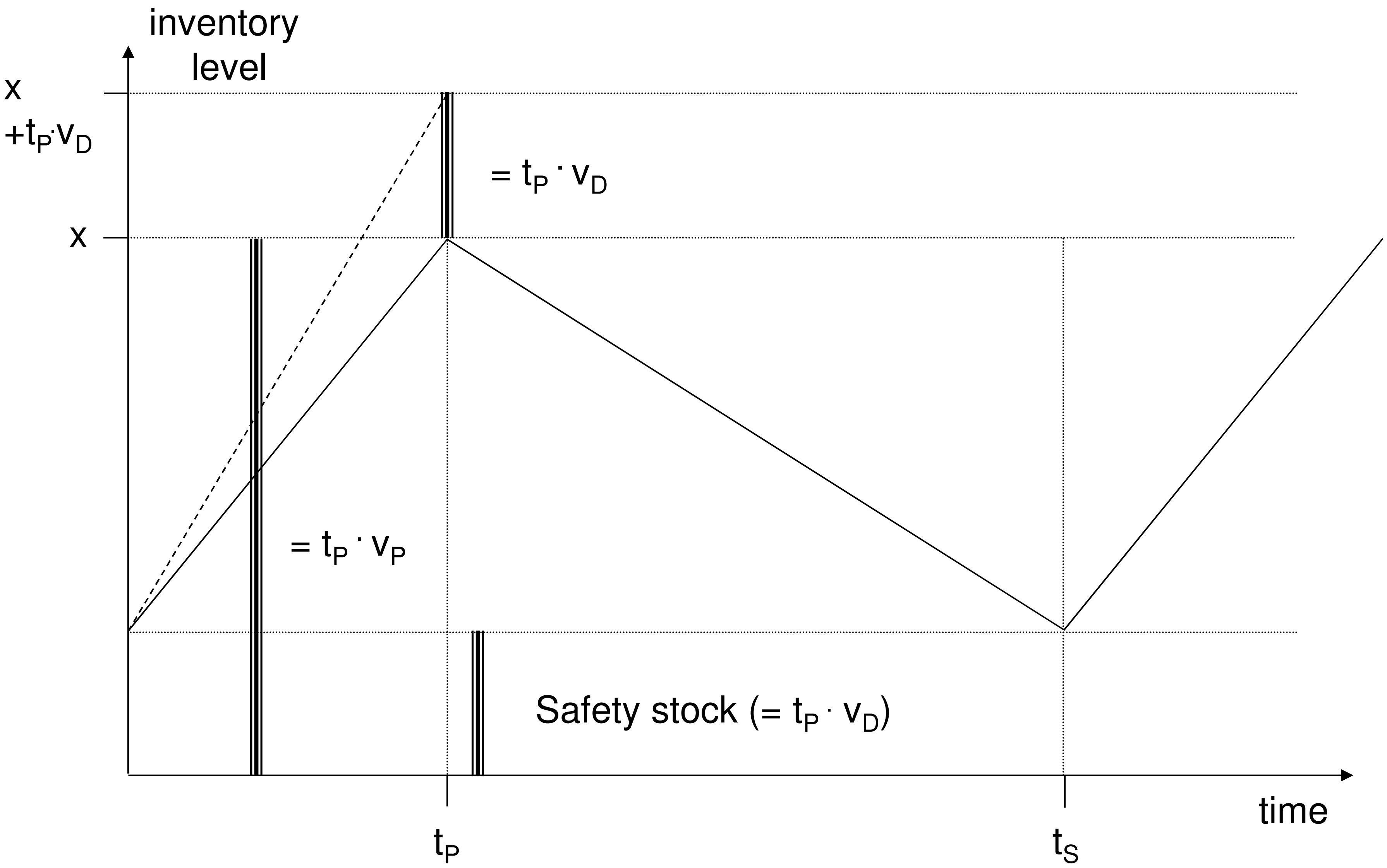
Computation of ØI(x) (open production)

- Inventory level always increases and decreases linearly
- This behavior is constant over the infinite planning horizon and is repeated for each processed batch
- The maximum inventory level is defined by $x - t_P \cdot v_D$
- The minimum inventory level is defined by 0
- Therefore, we get for the average inventory level:

$$\varnothing I(x) = \frac{1}{2} \cdot \left(\left(x - t_P \cdot v_D\right) + 0\right) = \frac{1}{2} \cdot \left(x - \frac{x}{v_P} \cdot v_D \right)$$

$$= \frac{1}{2} \cdot x \cdot \left(1 - \frac{v_D}{v_P} \right) = x \cdot \underbrace{\frac{1}{2} \cdot \left(1 - \frac{v_D}{v_P} \right)}_{\text{Proportion of the lot size x to be on stock on the average in the planning horizon}}$$

Inventory (closed production)



Computation of ØI(x) (closed production)

- Inventory level increases and decreases linearly
- This behavior is constant over the infinite planning horizon and is repeated for each processed batch
- The maximum inventory level is defined by x
- The minimum inventory level is defined by $t_P \cdot v_D$
- Therefore, we get for the average inventory level:

$$\varnothing I(x) = \frac{1}{2} \cdot \left(x + t_P \cdot v_D\right) = \frac{1}{2} \cdot \left(x + \frac{x}{v_P} \cdot v_D \right)$$

$$= \frac{1}{2} \cdot x \cdot \left(1 + \frac{v_D}{v_P}\right) = x \cdot \underbrace{\frac{1}{2} \cdot \left(1 + \frac{v_D}{v_P}\right)}_{\substack{\text{Proportion of the lot size x} \\ \text{to be on stock on the average} \\ \text{in the planning horizon}}}$$

Finding the optimal lot size (open production)

- Now, we can define the total cost function depending on the chosen lot size x:

$$\boxed{C_{total}(x) = \frac{x_T}{x} c_S + \varnothing I(x) \cdot c_I = \frac{x_T}{x} c_S + \frac{1}{2} \cdot x \cdot \left(1 - \frac{v_D}{v_P}\right) \cdot c_I}$$

- By using this function, we can easily derive the optimal lot size

Finding the optimal lot size (open production)

$$C_{total}(x) = \frac{x_T}{x} \cdot c_S + \frac{1}{2} \cdot x \cdot \left(1 - \frac{v_D}{v_P}\right) \cdot c_I$$

$$\frac{C_{total}(x)}{\partial x} = \frac{-x_T}{x^2} \cdot c_S + \frac{1}{2} \cdot \left(1 - \frac{v_D}{v_P}\right) \cdot c_I$$

$$\frac{\frac{C_{total}(x)}{\partial x}}{\partial x} = 2 \cdot \frac{x_T}{x^3} \cdot c_S > 0$$

$$\frac{C_{total}(x)}{\partial x} = 0 \Leftrightarrow \frac{-x_T}{x^2} \cdot c_S + \frac{1}{2} \cdot \left(1 - \frac{v_D}{v_P}\right) \cdot c_I = 0 \Leftrightarrow \frac{1}{2} \cdot \left(1 - \frac{v_D}{v_P}\right) \cdot c_I = \frac{x_T}{x^2} \cdot c_S$$

$$\Leftrightarrow x^2 \cdot \frac{1}{2} \cdot \left(1 - \frac{v_D}{v_P}\right) \cdot c_I = x_T \cdot c_S \Leftrightarrow x^2 \cdot = \frac{2 \cdot x_T \cdot c_S}{\left(1 - \frac{v_D}{v_P}\right) \cdot c_I} \Leftrightarrow x = \sqrt{\frac{2 \cdot x_T \cdot c_S}{\left(1 - \frac{v_D}{v_P}\right) \cdot c_I}}$$

Finding the optimal lot size (closed production)

- Now, we can define the total cost function depending on the chosen lot size x:

$$C_{total}(x) = \frac{x_T}{x} c_S + \varnothing I(x) \cdot c_I = \frac{x_T}{x} c_S + \frac{1}{2} \cdot x \cdot \left(1 + \frac{v_D}{v_P}\right) \cdot c_I$$

- By using this function, we can easily derive the optimal lot size

Finding the optimal lot size (closed production)

$$C_{total}(x) = \frac{x_T}{x} \cdot c_S + \frac{1}{2} \cdot x \cdot \left(1 + \frac{v_D}{v_P}\right) \cdot c_I$$

$$\frac{C_{total}(x)}{\partial x} = \frac{-x_T}{x^2} \cdot c_S + \frac{1}{2} \cdot \left(1 + \frac{v_D}{v_P}\right) \cdot c_I$$

$$\frac{\frac{C_{total}(x)}{\partial x}}{\partial x} = 2 \cdot \frac{x_T}{x^3} \cdot c_S > 0$$

$$\frac{C_{total}(x)}{\partial x} = 0 \Leftrightarrow \frac{-x_T}{x^2} \cdot c_S + \frac{1}{2} \cdot \left(1 + \frac{v_D}{v_P}\right) \cdot c_I = 0 \Leftrightarrow \frac{1}{2} \cdot \left(1 + \frac{v_D}{v_P}\right) \cdot c_I = \frac{x_T}{x^2} \cdot c_S$$

$$\Leftrightarrow x^2 \cdot \frac{1}{2} \cdot \left(1 + \frac{v_D}{v_P}\right) \cdot c_I = x_T \cdot c_S \Leftrightarrow x^2 \cdot = \frac{2 \cdot x_T \cdot c_S}{\left(1 + \frac{v_D}{v_P}\right) \cdot c_I} \Leftrightarrow x = \sqrt{\frac{2 \cdot x_T \cdot c_S}{\left(1 + \frac{v_D}{v_P}\right) \cdot c_I}}$$

Observation

- By analyzing the computation of the optimal lot size, it becomes obvious that for this lot size the **setup costs are identical with the occurring inventory costs**, i.e., it holds:

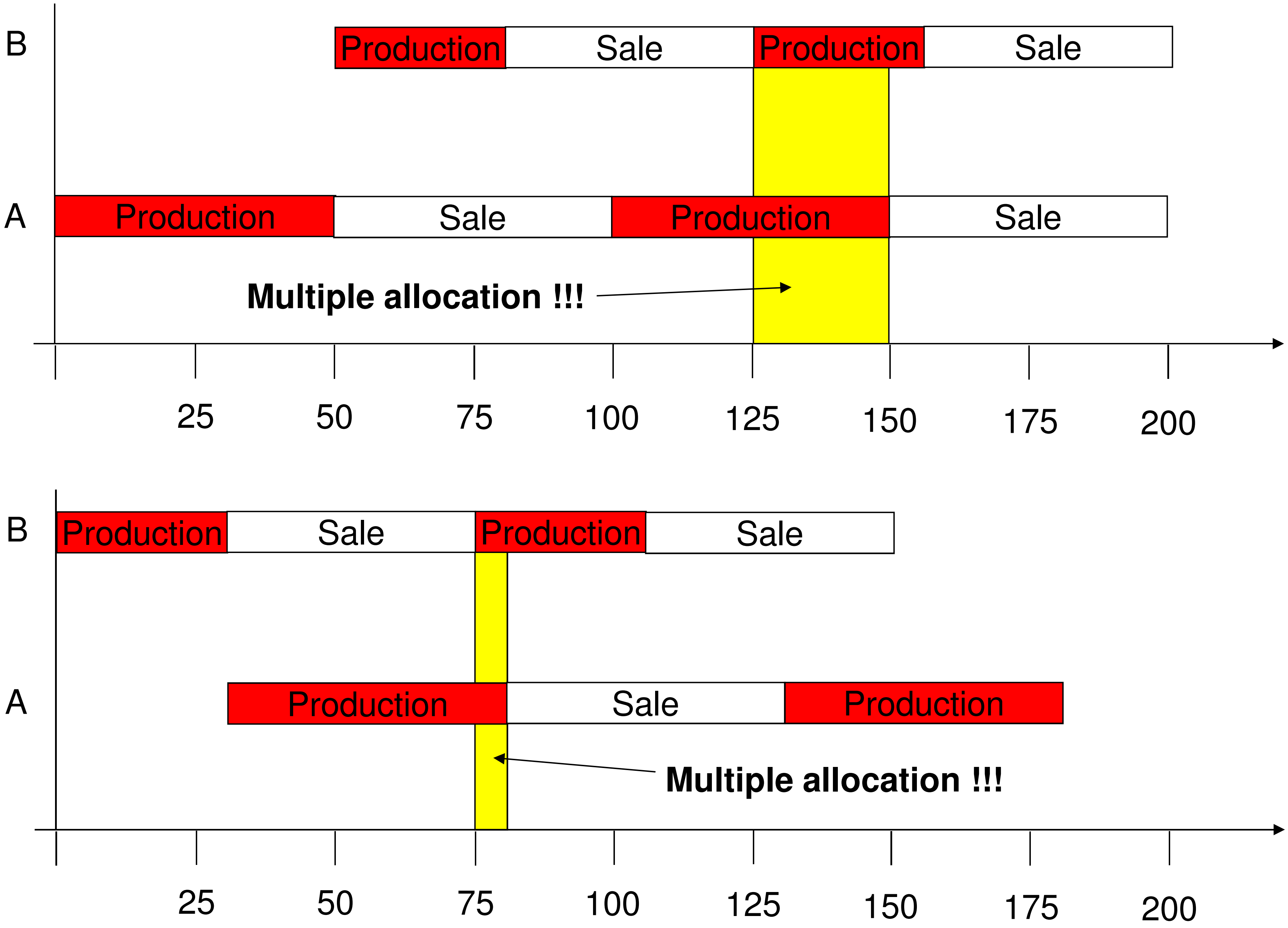
$$\frac{C_{total}(x)}{\partial x} = 0 \Leftrightarrow \frac{-x_T}{x^2} \cdot c_S + \frac{1}{2} \cdot \left(1 + \frac{v_D}{v_P}\right) \cdot c_I = 0 \Leftrightarrow \frac{1}{2} \cdot \left(1 + \frac{v_D}{v_P}\right) \cdot c_I = \frac{x_T}{x^2} \cdot c_S$$

i.e.

$$\Leftrightarrow \frac{1}{2} \cdot x \cdot \left(1 + \frac{v_D}{v_P}\right) \cdot c_I = x \cdot \frac{x_T}{x^2} \cdot c_S \Leftrightarrow \frac{1}{2} \cdot x \cdot \left(1 + \frac{v_D}{v_P}\right) \cdot c_I = \frac{x_T}{x} \cdot c_S$$

3.2 Extensions to multiple products cases

- The optimal individual lot sizes are frequently not applicable if there is more than one product. This can be illustrated by the following simple example
- Example:
  - Two products A and B have to be produced
  - Optimal individual lot sizes
    - x_A=1000 and x_B=3000 [quantity units]/[batch]
    - v_{DA}=10 and v_{PA}=20 [quantity units]/[minute]
    - v_{DB}=40 and v_{PB}=100 [quantity units]/[minute]
    - We can derive the respective time intervals:
      - t_{PA}=1000/20=50 [minutes]/[batch]
      - t_{DA}=1000/10=100 [minutes]/[batch]
      - t_{PB}=3000/100=30 [minutes]/[batch]
      - t_{DB}=3000/40=75 [minutes]/[batch]


B
Production
Sale
Production
Sale
A
Production
Sale
Production
Sale
Multiple allocation !!!
25
50
75
100
125
150
175
200
B
Production
Sale
Production
Sale
A
Production
Sale
Production
Multiple allocation !!!
25
50
75
100
125
150
175
200

Consequence

- We cannot produce A and B in their optimal lot sizes!
- Possible “work around”:
  - Approximate solutions
    - Try to generate a feasible solution as close as possible to the individual optimal lot sizes
  - Computation of optimal cycle times
    - Use lot sizes for the different products leading to an identical number of batches to be processed for all products

Approximate solution I

1. Generate the individual optimal lot sizes $x_{opt,n}$ ($n \in \{1,\ldots,N\}$) for N products by using the computation derived above
2. Calculate the resulting total costs of this optimal solution, i.e.,

$$C_{opt,T} = \sum_{n=1}^{N} C_{opt,T,n}\left(x_{opt,n}\right)$$

3. Define a fixed rate i as an upper bound for the percentage derivation of the resulting costs $C_{res,n}$ in comparison to the theoretic ones, i.e.,

$$\forall n \in \{1,\ldots,N\}: \frac{C_{res,n}}{C_{opt,T,n}\left(x_{opt,n}\right)} \leq 1 + i = q$$

Approximate solution II

4. Calculation of the lot size window of each product is depending on the costs rate

$$\forall n \in \{1,\ldots,N\}: \frac{C_{T,n}(x_n)}{C_{opt,T,n}(x_{opt,n})} \leq 1+i = q \Leftrightarrow x_n^2 - 2 \cdot q \cdot x_{opt,n} \cdot x_n + x_{opt,n}^2 \leq 0$$

$$\Leftrightarrow \underbrace{x_{opt,n} \cdot \left(q - \sqrt{q^2 - 1}\right)}_{x_{opt,n}^{lower}} \leq x_n \leq \underbrace{x_{opt,n} \cdot \left(q + \sqrt{q^2 - 1}\right)}_{x_{opt,n}^{upper}}$$

5. Check if there are possible constellations inside the computed windows for each product leading to feasible solutions
   - If so, realize the best one
   - Otherwise, continue with step 6
6. Increase i by a predefined percentage rate and proceed with step 4

Pros vs. Cons

Pros

+ High solution quality since the objective functions differ only slightly around the optimal lot size
+ Specific requirements of each product can be respected
+ Flexible adjustment

Cons

– No systematic approach
– Trial and error
– Can become extremely time consuming and, additionally, there is no guarantee for success

Optimal cycle times

- This “work around” tries to generate a realizable solution by requiring an identical number of batches for each considered product in the planning horizon
- To do so, we extend the model defined above by introducing an additional variable c as the sought optimal number of batches to be processed of each product, i.e., $c = x_{T,n} / x_n$
- Therefore, a new model arises with a single variable c while the lot size of each product can be derived from a defined value for c
- The optimal cycle time is defined as the cycle time leading to the minimal total costs of all products

Deriving the optimal cycle time (open production)

Objective function:

$$C_T(c) = \sum_{n=1}^{N} c \cdot c_{S,n} + \frac{1}{2} \cdot \frac{x_{T,n}}{c} \cdot \left(1 - \frac{v_{D,n}}{v_{P,n}}\right) \cdot c_{I,n}; \qquad \frac{C_T(c)}{\partial c} = \sum_{n=1}^{N} c_{S,n} + \frac{1}{2} \cdot \frac{-x_{T,n}}{c^2} \cdot \left(1 - \frac{v_{D,n}}{v_{P,n}}\right) \cdot c_{I,n}$$

$$\frac{\frac{C_T(c)}{\partial c}}{\partial c} = \sum_{n=1}^{N} \frac{x_{T,n}}{c^3} \cdot \left(1 - \frac{v_{D,n}}{v_{P,n}}\right) \cdot c_{I,n} \geq 0;$$

Finding the optimal cycle time

$$\frac{C_T(c)}{\partial c} = 0 \Leftrightarrow \sum_{n=1}^{N} c_{S,n} - \frac{1}{2} \cdot \frac{x_{T,n}}{c^2} \cdot \left(1 - \frac{v_{D,n}}{v_{P,n}}\right) \cdot c_{I,n} = 0$$

$$\Leftrightarrow \sum_{n=1}^{N} c_{S,n} = \sum_{n=1}^{N} \frac{1}{2} \cdot \frac{x_{T,n}}{c^2} \cdot \left(1 - \frac{v_{D,n}}{v_{P,n}}\right) \cdot c_{I,n}$$

$$\Leftrightarrow c^2 = \frac{\sum_{n=1}^{N} \frac{1}{2} \cdot x_{T,n} \cdot \left(1 - \frac{v_{D,n}}{v_{P,n}}\right) \cdot c_{I,n}}{\sum_{n=1}^{N} c_{S,n}} \Leftrightarrow c = \sqrt{\frac{\sum_{n=1}^{N} \frac{1}{2} \cdot x_{T,n} \cdot \left(1 - \frac{v_{D,n}}{v_{P,n}}\right) \cdot c_{I,n}}{\sum_{n=1}^{N} c_{S,n}}}$$

Deriving the optimal cycle time (closed production)

Objective function:

$$C_T(c) = \sum_{n=1}^{N} c \cdot c_{S,n} + \frac{1}{2} \cdot \frac{x_{T,n}}{c} \cdot \left(1 + \frac{v_{D,n}}{v_{P,n}}\right) \cdot c_{I,n}; \quad \frac{C_T(c)}{\partial c} = \sum_{n=1}^{N} c_{S,n} + \frac{1}{2} \cdot \frac{-x_{T,n}}{c^2} \cdot \left(1 + \frac{v_{D,n}}{v_{P,n}}\right) \cdot c_{I,n}$$

$$\frac{\dfrac{C_T(c)}{\partial c}}{\partial c} = \sum_{n=1}^{N} \frac{x_{T,n}}{c^3} \cdot \left(1 + \frac{v_{D,n}}{v_{P,n}}\right) \cdot c_{I,n} \geq 0;$$

Finding the optimal cycle time

$$\frac{C_T(c)}{\partial c} = 0 \Leftrightarrow \sum_{n=1}^{N} c_{S,n} - \frac{1}{2} \cdot \frac{x_{T,n}}{c^2} \cdot \left(1 + \frac{v_{D,n}}{v_{P,n}}\right) \cdot c_{I,n} = 0$$

$$\Leftrightarrow \sum_{n=1}^{N} c_{S,n} = \sum_{n=1}^{N} \frac{1}{2} \cdot \frac{x_{T,n}}{c^2} \cdot \left(1 + \frac{v_{D,n}}{v_{P,n}}\right) \cdot c_{I,n} \quad \Leftrightarrow c^2 = \frac{\sum_{n=1}^{N} \frac{1}{2} \cdot x_{T,n} \cdot \left(1 + \frac{v_{D,n}}{v_{P,n}}\right) \cdot c_{I,n}}{\sum_{n=1}^{N} c_{S,n}}$$

$$\Leftrightarrow c = \sqrt{\frac{\sum_{n=1}^{N} \frac{1}{2} \cdot x_{T,n} \cdot \left(1 + \frac{v_{D,n}}{v_{P,n}}\right) \cdot c_{I,n}}{\sum_{n=1}^{N} c_{S,n}}}$$

Pros vs. Cons

+ Frequently a solution is generated that is feasible and quite efficient
+ Systematic approach
+ Fast solution generation

– Generates a rough compromise
– Neglects frequently many insights of the different considered products by a summarized simultaneous examination of all items

3.3 The SLULSP model (WW model)

= **S**ingle-**L**evel **U**ncapacitated **L**ot **S**izing **P**roblem
Also called **W**agner **W**hitin model (WW-model)

- Dynamic model (changing demand)
- Finite planning horizon which is subdivided into several discrete periods of predefined length
- Demand is given for each period but can vary from period to period
- Demand must be satisfied in each period
- Capacity restrictions are not considered
- Single item model

3.3.1 Model definition – Parameters

T: Number of considered periods;

$d_t\left(1 \le t \le T\right)$: Amount demanded in period t;

$i_t\left(1 \le t \le T\right)$: Interest charge per unit of inventory carried forward to period $t+1$;

$s_t\left(1 \le t \le T\right)$: Ordering (or setup) costs in period t;

$p_t\left(1 \le t \le T\right)$: Production costs in period t;

I_0: Initial inventory;

M: Large number;

Model definition – Variables

$x_t \; (1 \leq t \leq T)$: Chosen lot size in period t;

$\gamma_t \; (1 \leq t \leq T)$: Binary derived variable indication a setup operation in period t;

$I_t \; (1 \leq t \leq T)$: Inventory in period t;

Restrictions

- We have to find a program $(x_1,\ldots,x_T)$ for all considered periods, so that all demands are met at minimal total costs
- In each period the current inventory level can be computed by the difference of production and demand added to the inventory of the preceding period
- Setup costs always occur in a period if there is a production quantity unequal to null
- We additionally assume that the initial as well as the final inventory is equal to null

$$\forall t \in \{1,\ldots,T-1\}: I_0 + \sum_{j=1}^{t} x_j - \sum_{j=1}^{t} d_j \geq 0;$$
$$\forall t \in \{1,\ldots,T\}: I_{t-1} + x_t - I_t = d_t;$$
$$\forall t \in \{1,\ldots,T\}: x_t - M \cdot \gamma_t \leq 0;$$
$$\forall t \in \{1,\ldots,T\}: x_t \geq 0$$
$$I_0 = I_T = 0;$$
$$\forall t \in \{1,\ldots,T\}: \gamma_t \in \{0,1\};$$

Objective function

- An efficient production plan should minimize the resulting total sum of setup-, production-, and inventory costs, i.e., we can derive the following objective function:

$$\boxed{\text{Minimize } C_T(x_1,\ldots,x_T) = \sum_{t=1}^{T} \left(s_t \cdot \gamma_t + i_t \cdot I_t + p_t \cdot x_t \right)}$$

Main cognitions – First substantial Theorem

3.3.1.1 Theorem

There exists an optimal program fulfilling the following restrictions:

$$\boxed{\forall t \in \{1,\ldots,T\}: \underbrace{\left(I_0 + \sum_{j=1}^{t-1} x_j - \sum_{j=1}^{t-1} d_j \right)}_{=I_{t-1}} \cdot x_t = 0}$$

I.e., in each period, there is either an existing inventory or an additional order is generated. This means that the production of additional items is processed if and only if the inventory is totally consumed in the previous periods.

Proof of Theorem 3.3.1.1

Proof:

We assume there is an optimal program not fulfilling the itemized restriction for a minimally chosen period s. Therefore, it holds:

$$\boxed{\left(I_0 + \sum_{k=1}^{s-1} x_k - \sum_{k=1}^{s-1} d_j\right) \cdot x_s > 0 \Rightarrow I_0 + \sum_{k=1}^{s-1} x_k - \sum_{k=1}^{s-1} d_j > 0 \wedge x_s > 0}$$

Let I_{s-1} be the inventory brought into period s. Let $r<s$ be the next preceding period where a production takes place (Note that r is well defined since at least period one fulfills this requirement).

Note that $x_r \geq I_{s-1}$ since $I_{r-1}=0$ (s was minimally chosen) and we have an inventory in period s.

Proof of Theorem 3.3.1.1

If it holds $c_{r,s}>p_s$ ($c_{r,s}$ are the total costs for producing one unit of demand of period s in period r and carry it over to period s), we produce the I_{s-1} items not until period s. Since this reduces the total costs, it contradicts the optimality of the solution found.

Thus, we know $c_{r,s} \leq p_s$. Hence, we abstain from producing in period s and increase the production quantity in period r by x_s items. Owing to the optimality, it holds that $c_{r,s}=p_s$ and we can transform the solution as intended without losing its optimality.

Second substantial Theorem

3.3.1.2 Theorem

There exists an optimal program so that:

$$\boxed{\forall t \in \{1,\dots,T\}: x_t = 0 \vee x_t = \sum_{j=t}^{k} d_j \text{ for some } k, t \le k \le T}$$

Proof:

We assume again that an optimal program does not fulfill the defined restriction. Since the occurring demand must be always satisfied by the production, there is a period t where it holds:

$$\boxed{x_t = \sum_{j=t}^{k} d_j + c \text{ with } c > 0, c < d_{k+1} \text{ and } k < T}$$

Proof (continued):

Therefore, we know that there is a period s>t where it holds:

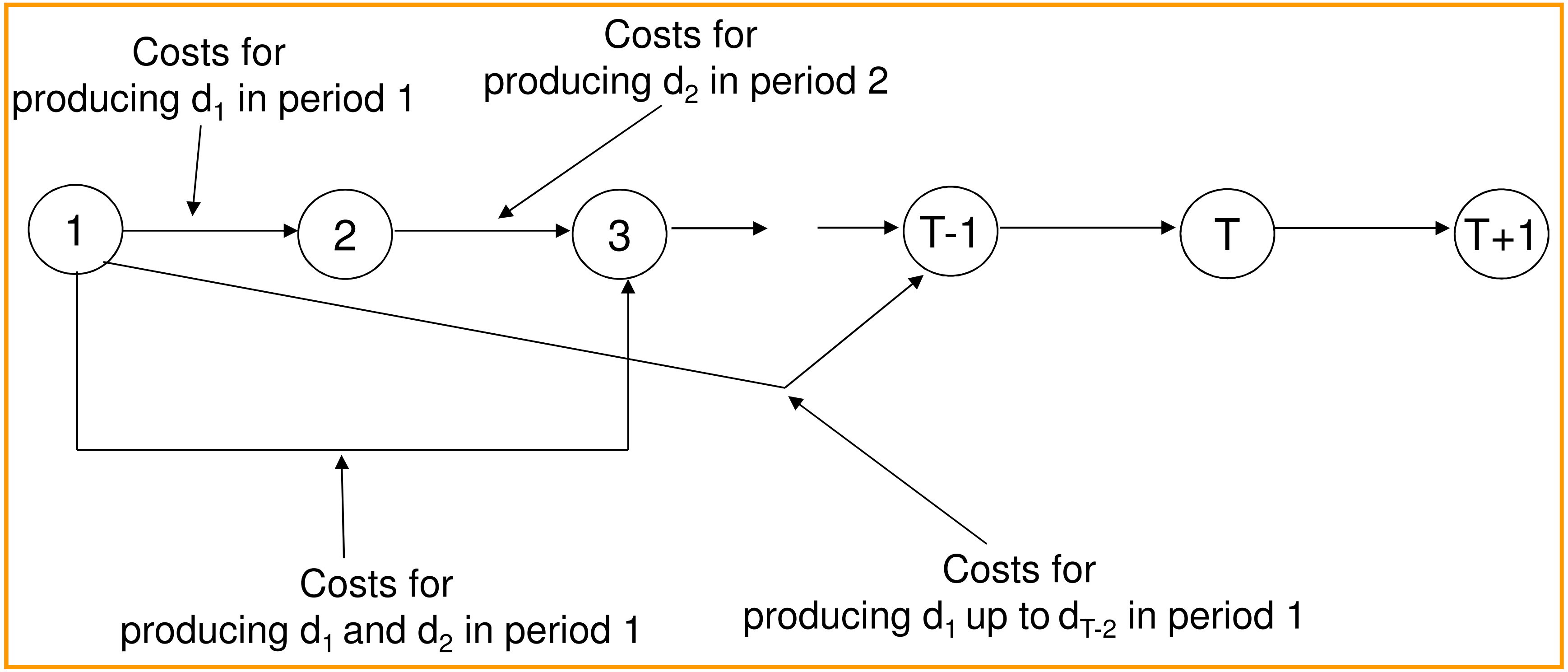
$$\boxed{\left(I_0+\sum_{k=1}^{s-1}x_k-\sum_{k=1}^{s-1}d_j\right)\cdot x_s>0 \Rightarrow I_0+\sum_{k=1}^{s-1}x_k-\sum_{k=1}^{s-1}d_j>0 \wedge x_s>0}$$

Now, we can apply Theorem 3.3.1.1 to finish the proof

Graph representation

- By using the two Theorems defined above, we can define an alternative problem definition
- This description transforms the problem into a **shortest path problem**
- In this graph for each considered period an additional node is inserted defining the isolated decision situation where in this period no inventory is left over
- Each edge represents a specific lot size leading to the subsequent period where a further production becomes necessary again
- With each edge a cost weight is associated representing the additional costs occurring in the realization of the respective lot size in the mapped constellation
- Finding a cost minimal production plan is equivalent to the computation of the shortest path in the defined graph

Illustration



$\Rightarrow$ Additional costs incurred by a specific lot size represented by an edge leading from node r to t ($t \geq r$):

$$w_{r,t} = s_r + p_r \cdot \left(\sum_{k=r}^{t-1} d_k \right) + \sum_{a=r+1}^{t-1} \sum_{b=r}^{a-1} (i_b \cdot d_a)$$

SRP =Shortest Route Problem

Parameters :

$\forall i \in \{1,\ldots,T\} : \forall j \in \{i+1,\ldots,T+1\} : w_{i,j}$: Costs for the satisfaction of the demand of the periods i through $j-1$ by the production in i

T : Total number of periods

Variables :

$\forall i \in \{0,\ldots,T\} : \forall j \in \{i+1,\ldots,T+1\} : x_{i,j}$: Binary decision variable indicating whether the demand of the periods i through $j-1$ is satisfied by the production in i

Objective function :

Minimize $Z = \sum_{r=1}^{T} \sum_{t=r+1}^{T+1} w_{r,t} \cdot x_{r,t}$

SRP =Shortest Route Problem

Restrictions :

$$\sum_{t=2}^{T+1} x_{1,t} = 1$$

$$\forall t \in \{2,\ldots,T\} : -\sum_{l=1}^{t-1} x_{l,t} + \sum_{l=t+1}^{T+1} x_{t,l} = 0$$

$$\forall s \in \{0,\ldots,T-1\} : \forall t \in \{s+1,\ldots,T+1\} : x_{s,t} \in \{0,1\}$$

3.3.2 Dynamic programming approach

- Wagner and Whitin propose a dynamic programming algorithm working with the following recursive function
- Recursive function
  - For periods i≤j $p_{i,j}$ defines a policy satisfying the demand of the periods i,…,j by a production in period i
  - In this coherence, C(i,j) (or $C_{i,j}$) gives the respective total costs of policy $p_{i,j}$
  - By using these notations, we come to the following simple functional dependency for the calculation of the minimal costs f_i to satisfy the demands $d_1,\ldots,d_i$

$$f_i = \min_{1 \le l \le i} \left\{ f_{l-1} + C(l,i) \right\} \text{ with}$$

$$f_0 = 0$$

and

$$\forall i,j \in \{1,\ldots,T\} (i \le j): C(i,j) = p_i \cdot \sum_{k=i}^{j} d_k + s_i + \sum_{r=i}^{j-1} \sum_{R=r+1}^{j} i_r \cdot d_R$$

Computational effort

- In the worst case, we have altogether T recursions
- In the recursion for f_i, we have to consider altogether O(i) constellations
- Altogether, we need $O(T^2)$ parameters during the recursion
- Total effort: $O(1+2+3+4+..+T)=O(T^2)$

Example

- 6 periods
- Setup costs per batch: $s_1=s_2=s_3=s_4=s_5=s_6=500$
- Production costs are neglected
- Inventory costs per item and period $i_1=i_2=i_3=i_4=i_5=i_6=1$
- Demands:

t	1	2	3	4	5	6
d_t	20	80	160	85	120	100

Owing to these simplifications we get:

$$\forall i,j \in \{1,\ldots,T\}(i \le j): C(i,j) = 500 + \sum_{r=i}^{j-1} \sum_{R=r+1}^{j} d_R = 500 + \sum_{r=i+1}^{j} (r-i)\cdot d_r$$

Preliminary work (C(i,j))

	Last period where a consumption takes place					
Last period where a production takes place	**1**	**2**	**3**	**4**	**5**	**6**
1	500	580	900	1155	1635	2135
2		500	660	830	1190	1590
3			500	585	825	1125
4				500	620	820
5					500	600
6						500

Recursive computation

$$f_0 = 0$$

$$f_1 = \min\{f_0 + c_{1,1}\} = 500$$

$$f_2 = \min\{f_0 + c_{1,2}, f_1 + c_{2,2}\} = \{0 + 580, 500 + 500\} = \{580, 1000\} = 580$$

$$f_3 = \min\{f_0 + c_{1,3}, f_1 + c_{2,3}, f_2 + c_{3,3}\} = \{0 + 900, 500 + 660, 580 + 500\}$$
$$= \{900, 1160, 1080\} = 900$$

$$f_4 = \min\{f_0 + c_{1,4}, f_1 + c_{2,4}, f_2 + c_{3,4}, f_3 + c_{4,4}\}$$
$$= \{0 + 1155, 500 + 830, 580 + 585, 900 + 500\}$$
$$= \{1155, 1330, 1165, 1400\} = 1155$$

Recursive computation

$$f_5 = \min\{f_0 + c_{1,5}, f_1 + c_{2,5}, f_2 + c_{3,5}, f_3 + c_{4,5}, f_4 + c_{5,5}\}$$
$$= \{0 + 1635, 500 + 1190, 580 + 825, 900 + 620, 1155 + 500\}$$
$$= \{1635, 1690, 1405, 1520, 1655\} = 1405$$

$$f_6 = \min\{f_0 + c_{1,6}, f_1 + c_{2,6}, f_2 + c_{3,6}, f_3 + c_{4,6}, f_4 + c_{5,6}, f_5 + c_{6,6}\}$$
$$= \{0 + 2135, 500 + 1590, 580 + 1125, 900 + 820, 1155 + 600, 1405 + 500\}$$
$$= \{2135, 2090, 1705, 1720, 1755, 1905\} = 1705$$

Recursive construction of the solution

- Consider f_6:
  - Best solution $f_2+c(3,6)$, i.e., the demand of the periods 3,4,5, and 6 is produced in period 3
  - For the first two periods we have to go on with f_2
- Consider f_2:
  - Best solution $f_0+c(1,2)$, i.e., the demand of the periods 1 and 2 is produced in period 1
  - Therefore, altogether we have two batches produced in period 1 and in period 3
- Summary:
  - Period 1: Production of $d_1+d_2=100$
  - Period 3: Production of $d_3+d_4+d_5+d_6=160+85+120+100=465$
  - Total costs: 1,705

Further improvements and observations

- The algorithm described above generates an optimal solution within $O(T^2)$ steps
- By using specific data structures, the computational effort for finding the optimal solution can be reduced to $O(T \log(T))$
- For the special case characterized by constant production costs $p=p_1=p_2=\ldots=p_T$, this effort can be additionally reduced to $O(T)$ (cf. Federgruen and Tzur (1991))
- This solution is only optimal if the starting and ending inventory is zero. However, this is not necessarily a valid assumption for a realistic application in a rolling time horizon

3.3.3 Heuristic approaches

- In the following, we consider different heuristic approaches. These procedures can be applied for large problem instances as well as in a modified version for the multiple product constellations
- Described approaches
  - Method of a “least-unit cost” approach
  - Silver-Meal procedure

Least unit cost approach

- Consider an arbitrary period t (1≤t≤T). If a batch satisfying the subsequent periods t to s (s≥t) is produced, we have the following average costs per item:

$$c_{t,s}^{unit} = \frac{s_t + \sum_{c=t}^{s-1} \sum_{b=c+1}^{s} i_c \cdot d_b + p_t \cdot \sum_{c=t}^{s} d_c}{\sum_{c=t}^{s} d_c}$$

- In every period t the period s is sought which fulfills the following expression:

$$\min\left(\left\{ j \mid c_{t,j+1}^{unit} > c_{t,j}^{unit} \ \text{ with } T-1 \geq j \geq t \right\} \cup \{T\}\right)$$

A heuristic approach

1. $cp = 1$
2. While $cp < T$ do
3. Planning of batch in period cp

3.1 $j = cp$;

3.2 $c^{unit}_{cp,j} = \dfrac{s_{cp} + p_{cp} \cdot d_{cp}}{d_{cp}}$; $it = true$;

3.3 While it do

3.3.1 Compute $c^{unit}_{cp,j}$

3.3.2 if $\left(c^{unit}_{cp,j} > c^{unit}_{cp,j-1}\right)$ then $it = false$ else $it = true$

3.3.3 if $it = true$ then $j = j + 1$

3.3.4 od

Example

$$t = 1:\ j = 1: c_{1,1}^{unit} = \frac{500}{20} = 25$$

$$t = 1:\ j = 2: c_{1,2}^{unit} = \frac{580}{100} = 5{,}8$$

$$t = 1:\ j = 3: c_{1,3}^{unit} = \frac{900}{260} = 3{,}46$$

$$\mathbf{t = 1: j = 4: c_{1,4}^{unit} = \frac{1155}{345} = 3{,}35}$$

$$t = 1:\ j = 5: c_{1,5}^{unit} = \frac{1635}{465} = 3{,}52$$

$$q_1 = 345$$

Example

$$t = 5 : j = 5 : c_{5,5}^{unit} = \frac{500}{120} = 4{,}17$$

$$\mathbf{t = 5 : j = 6 : c_{5,6}^{unit} = \frac{600}{220} = 2{,}72}$$

$$q_5 = 220$$

$$\text{Costs} : 1755$$

Silver-Meal procedure

- The Silver-Meal procedure works quite similar to the least-unit cost procedure considered before
- The only difference results from a modified criteria to decide about the number of subsequent periods satisfied by the production of a period currently considered
- This criterion is given by the **average costs per period** occurring for the realization of a specific lot size
- To do so, consider an arbitrary period t (1≤t≤T). If a batch satisfying the subsequent periods t to s (s≥t) is produced, we have the following average costs per time period:

$$\boxed{c_{t,s}^{period} = \frac{s_t + \sum_{c=t}^{s-1} i_c \cdot \left(\sum_{b=c+1}^{s} d_b \right) + p_t \cdot \sum_{c=t}^{s} d_c}{s-t+1}}$$

- Similar to the least-unit cost procedure in every period t, the period s is sought that fulfills the following expression

$$\boxed{\min\left(\left\{ j \mid c_{t,j+1}^{period} > c_{t,j}^{period} \text{ with } T-1 \geq j \geq t \right\} \cup \{T\}\right)}$$

The resulting procedure

1. $cp = 1$
2. While $cp < T$ do
3. Planning of batch in period cp

3.1 $j = cp;$

3.2 $c_{cp,j}^{period} = \dfrac{s_{cp} + p_{cp} \cdot d_{cp}}{j - cp + 1};\ it = true;$

3.3 While it do

3.3.1 Compute $c_{cp,j}^{period}$

3.3.2 if $\left(c_{cp,j}^{period} > c_{cp,j\text{-}1}^{period}\right)$ then $it = false$ else $it = true$

3.3.3 if $it = true$ then $j = j + 1$

3.3 od

Example

$$t=1: j=1: c_{1,1}^{period} = \frac{500}{1} = 500$$

$$\mathbf{t=1: j=2: c_{1,2}^{period} = \frac{580}{2} = 290}$$

$$t=1: j=3: c_{1,3}^{period} = \frac{900}{3} = 300$$

$$t=1: j=4: c_{1,4}^{period} = \frac{1155}{4} = 288{,}75$$

$$t=1: j=5: c_{1,5}^{period} = \frac{1635}{5} = 327$$

$$q_1 = 100$$

Example

$$t=3: j=3: c_{3,3}^{period}=\frac{500}{1}=500$$

$$t=3: j=4: c_{3,4}^{period}=\frac{585}{2}=292{,}5$$

$$\mathbf{t=3: j=5: c_{3,5}^{period}=\frac{825}{3}=275}$$

$$t=3: j=6: c_{3,6}^{period}=\frac{1125}{4}=281{,}25$$

$$q_3=365$$

Example

$$t = 6: \; j = 6: c_{6,6}^{period} = \frac{500}{1} = 500$$

$$q_6 = 100$$

$$\text{Total costs}: 580 + 825 + 500 = 1905$$

Observations

- The SRP can be characterized as a pure shortest path problem where we have to find the shortest connection between source and sink comprising total costs for the production of the demanded quantities
- Unfortunately, the integration of additional existing restrictions frequently given in real applications cannot be handled
- In order to do so, a modified version SRP_G of the SRP is proposed where we use continuous variables instead of integers
- In this model we can add arbitrary capacity restrictions often occurring in industrial applications

3.4 The CLSP model

=**C**apacitated **L**ot-**S**izing **P**roblem

- Extension of the SLULSP model by integrating **multiple products with dynamically changing demands**
- The available capacities are limited and must be shared between the different products
- Big-bucket model, i.e., long periods, J jobs per bucket to be processed

Big- vs. Small-bucket problems

- In literature, two main types of lot-sizing models are distinguished:
  - **Big-bucket models:** The planning horizon is divided into larger sub-horizons (called buckets) which allow the processing of multiple products where different setup states are necessary. Consequently, the respective models characterized as big-bucket approaches are defined as multiple product concepts, where individual setup and processing times for each resource are present (cf. CLSP). Setup states between neighboring buckets are not preserved while it is assumed that, due to the time dominance of the bucket sizes in comparison to the setup times, the non-preservation of specific setup states between successive buckets causes only small and negligible errors

Big- vs. Small-bucket problems

- **Small-bucket models:** Allow only at most one setup activity per bucket. Therefore, the model additionally comprises a sequence decision with respect to the jobs to be processed on the considered machines. As a consequence, a problem instance occurs which comprises frequently a large number of buckets by mapping realistic sized problems

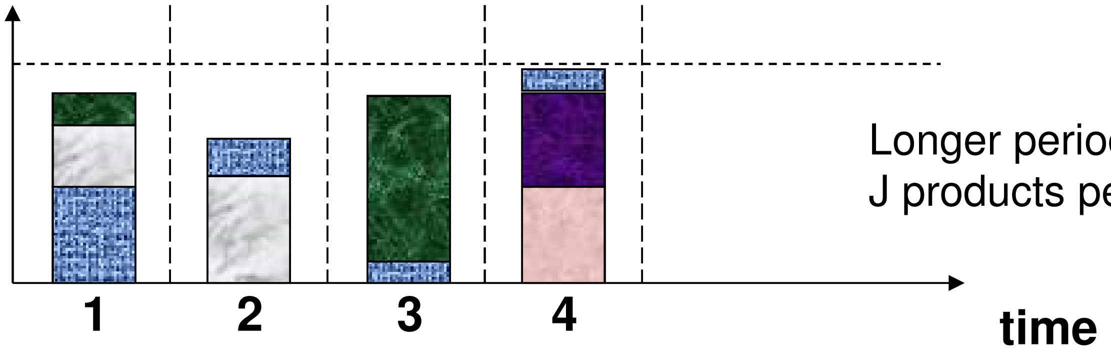
➢ Trend towards the more accurate small-bucket models, especially for applications with larger lead times (inherent drawback of big-bucket models)

Linked models

- To combine the advantages of the two types by preventing the respective disadvantages, the **lot-sizing models with linked lot sizes** are proposed in new publications,
- …namely the CLSP*L* as a big-bucket model with the additional attribute that existing **setup states can be preserved** between successive buckets

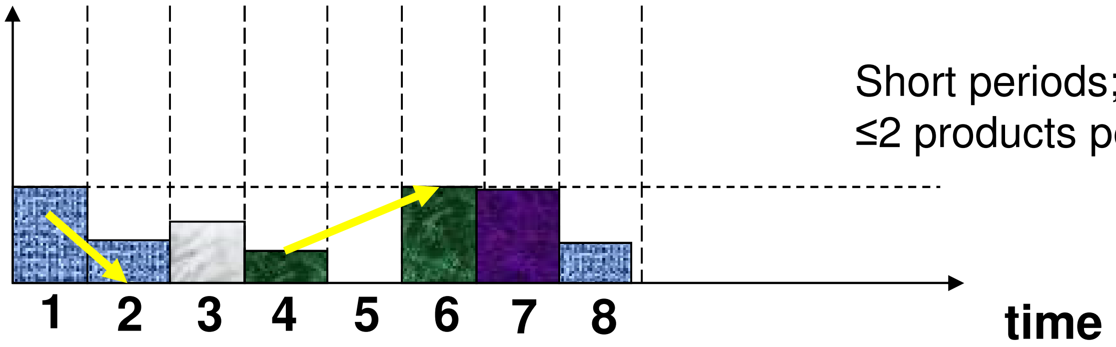
Illustration

Big-bucket model



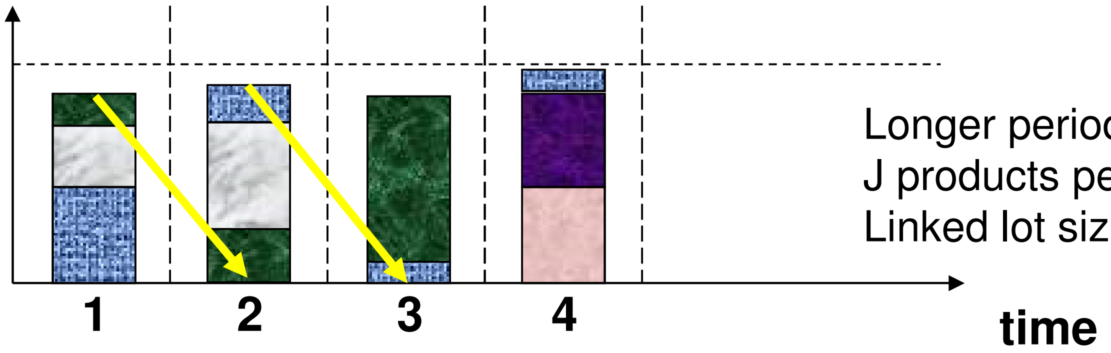
Longer periods;
J products per bucket

Small-bucket model



Short periods;
≤2 products per bucket

CLSP*L*



Longer periods;
J products per bucket;
Linked lot sizes

CLSP – Assumptions

- The **planning horizon is fixed** and divided into T time buckets, numbered from 1 to T
- **Resource consumption to produce a product j on a specific resource m is fixed**, and there exists a unique assignment of products to resources
- Setup processes incur **setup costs** and **consume setup time**, thereby reducing capacity in the respective period. Costs and consumed time occur sequence-independent
- **No setup state can be preserved to the subsequent bucket**

3.4.1 Mathematical definition – Parameters

T: Number of considered periods $[-]$;

J: Number of available resources $[-]$;

K: Number of products $[-]$;

M: Large number $[-]$;

$b_{j,t}\ (1 \le j \le J; 1 \le t \le T)$: Capacity of resource j in period t [time units];

$P_{k,t}\ (1 \le k \le K; 1 \le t \le T)$: Primary, gross-demand for item k in period t [product units];

$h_k\ (1 \le k \le K)$: Holding cost for one unit of product k per period [currency units/product units];

3.4.1 Mathematical definition – Parameters

$s_k \left(1 \le k \le K\right)$: Ordering (or setup) costs for product k $\left[\text{currency units/product units}\right]$;

$p_{k,t} \left(1 \le k \le K; 1 \le t \le T\right)$: Production costs for product k in period t $\left[\text{currency units/product units}\right]$;

$to_{j,k} \left(1 \le j \le J; 1 \le k \le K\right)$: Operating time for each item of product k on resource j $\left[\text{time units/product units}\right]$;

$ts_{j,k} \left(1 \le j \le J; 1 \le k \le K\right)$: Setup time for product k on resource j $\left[\text{time units/batches}\right]$;

Mathematical definition – Variables

$X_{k,t} \left(1 \le k \le K; 1 \le t \le T\right)$: Lot size of product k in period t;

$\gamma_{k,t} \left(1 \le k \le K; 1 \le t \le T\right)$: Binary derived variable indicating a setup operation of product k in period t;

$y_{k,t} \left(1 \le k \le K; 0 \le t \le T\right)$: Derived variable defining the inventory of product k at the end of period t;

Objective function:

$$\text{Minimize } Z = \sum_{k=1}^{K} \sum_{t=1}^{T} s_k \cdot \gamma_{k,t} + h_k \cdot y_{k,t} + p_{k,t} \cdot X_{k,t}$$

Mathematical definition – Restrictions I

$$\forall k \in \{1,\ldots,K\} : \forall t \in \{1,\ldots,T\} : y_{k,t-1} + X_{k,t} - y_{k,t} = P_{k,t};$$

The demand in every period of each product must be fulfilled by the inventory and additional production

$$\forall k \in \{1,\ldots,K\} : \forall t \in \{1,\ldots,T\} : X_{k,t} - M \cdot \gamma_{k,t} \leq 0;$$

Derivation of the binary setup variables

$$\forall j \in \{1,\ldots,J\} : \forall t \in \{1,\ldots,T\} : \sum_{k=1}^{K} \left(to_{j,k} \cdot X_{k,t} + ts_{j,k} \cdot \gamma_{k,t} \right) \leq b_{j,t}$$

Compliance with the time restriction of each available resource

Mathematical definition – Restrictions II

$\forall k \in \{1,...,K\} : \forall t \in \{1,...,T\} : X_{k,t} \geq 0$ Non-negative lot sizes

$\forall k \in \{1,...,K\} : y_{k,0} = 0 \wedge y_{k,T} = 0$ Start and end inventory is zero

$\forall k \in \{1,...,K\} : \forall t \in \{1,...,T\} : y_{k,t} \geq 0$ Non-negative inventories

$\forall k \in \{1,...,K\} : \forall t \in \{1,...,T\} : \gamma_{k,t} \in \{0,1\}$ Possible values for the derived variable

3.4.2 Solution methods

- The CLSP can be directly solved by using a standard solver
- This, however, causes frequently an unacceptable computational effort
- Two different solution methods are frequently proposed to prevent this computational effort:
  - Use of the Shortest Path Problem SRP_G:
    - Integration of capacity restrictions
    - Easier to solve due to its flow attitude
  - Use of appropriate heuristics
    - The procedure of Dixon and Silver
    - The ABC-procedure of Maes

The procedure of Dixon and Silver

- Heuristic approach
- The use of only one resource (one machine) is assumed
- Bases on (idea derived from) the Silver-Meal heuristic
  - In every iteration the procedure tries to minimize the average costs per period caused by each product
  - But due to the simultaneous production of several products, capacity restrictions can prevent a sequence defined according to this criterion
  - Therefore, the procedure has to define additionally some priority rules to decide about which product can be produced according to the decisions of the Silver-Meal procedure

Preparing main instruments

- In each iteration the procedure tries to extend the production in the current period by integrating the demand of a following period of some product, i.e., all products compete for implementing its production in a subsequent period if this integration attains a reduction of the respective average costs per period
- But, owing to the fact that the available capacity defines the bottleneck of the planning process, all cost reductions are interpreted according to their capacity requirements, i.e., for each product k the integration of the demand of the j+1-th period in the production generated in period i is rated by the following priority $\Delta_{k,i}$:

$$\Delta_{k,i} = \left[c_{k,i,j}^{period} - c_{k,i,j+1}^{period}\right] \Big/ \left[to_k \cdot d_{k,j+1}\right]$$

with $d_{k,j}$ as the demand of product k in period j

Interpretation

- $\Delta_{k,i}$ gives the relative reduction of average production costs per period of product k produced in period *i* by integrating the demand of period *j+1* per capacity unit to be used for its realization
- By integrating the products in the sequence of non-increasing Δ-values, a solution arises optimally applying the Silver-Meal criterion according to the local minimization of total average costs per period

Pseudocode of the Dixon and Silver procedure

```
While current period i≤T do
    Iteration i:
        While capacity is available and costs per
        period reductions for some products are still
        possible, do
                Enlarge production quantities by
                integrating subsequent periods of
                the product with largest Δ-value
Od (end while)
```

Capacity requirements

- In order to respect the capacity requirements in the CLSP model, it may become necessary to advance some productions to earlier periods
- First of all, we therefore have to check whether a given instance is solvable at all. This can be checked by the following requirements:

$$\forall t \in \{1,\ldots,T\}: \quad \underbrace{\sum_{j=1}^{t} \sum_{k=1}^{K} to_k \cdot d_{k,j}}_{\text{Total capacity demand up to period t}} \leq \underbrace{\sum_{j=1}^{t} b_j}_{\text{Total available capacity up to period t}}$$

- Note that average setup times are already subtracted from the capacities
- Note further that these times cannot be exactly computed beforehand since they depend on the found solution

Consequences

- In order to guarantee a feasible solution generated in the computation of the procedure of Dixon and Silver, we have to introduce some additional shortcuts
- In what follows, we define:

$$\forall i \in \{1,\ldots,T\} : \forall j \in \{i, i+1,\ldots,T\} : n_{i,j,k} :$$
$$\text{Already in period } i \text{ produced quantity}$$
$$\text{of the demand of product } k \text{ needed}$$
$$\text{in period } j$$

Definition of used parameters

-Capacity usage in i for j: $\forall i \in \{1,...,T\}: \forall j \in \{i,...,T\}: CU_{i,j} = \sum_{k=1}^{K} to_k \cdot n_{i,j,k}$

-Total capacity usage in i: $\forall i \in \{1,...,T\}: CU_i = \sum_{j=i}^{T} CU_{i,j}$

-Total capacity demand of period j: $\forall j \in \{1,...,T\}: CD_j = \sum_{k=1}^{K} to_k \cdot d_{k,j}$

-Total net-demand of period j in period i: $\forall i \in \{1,...,T\}: \forall j \in \{i,...,T\}:$

$$CN_{i,j} = CD_j - \sum_{t=1}^{i-1} \sum_{k=1}^{K} to_k \cdot n_{t,j,k}$$

-Needed capacity in j remaining for i: $\forall i \in \{1,...,T\}: \forall j \in \{i,...,T\}:$

$$CR_{i,j} = CN_{i,j} - b_j$$

Observation

- A capacity shortage $CR_{i,j}>0$ can be satisfied only if an additional production in the periods i,i+1,…,j-1 is established
- Therefore, a period i has to account for a production of the period j if the intermediate periods are not able to fulfill its capacity requirements
- Therefore, in order to guarantee the feasibility of a generated solution – if possible – a period i has to integrate the additional capacity requirements:

$$\max\left\{0, \max\left\{\sum_{j=i+1}^{t}\left(CR_{i,j} - CU_{i,j}\right) \mid i \in \{1,\ldots,T-1\} \wedge i < t \leq T\right\}\right\}$$

Consequences

- By integrating these additional quantities in the i-th period, the demand of all subsequent periods can be satisfied
- As a consequence, we fulfill the following necessary restrictions ensuring the feasibility:

$$\forall i \in \{1,\ldots,T-1\} : \forall t \in \{i+1,\ldots,T\} : \sum_{j=i+1}^{t} CR_{i,j} \leq \sum_{j=i+1}^{t} CU_{i,j}$$

The Dixon and Silver procedure

- As a consequence, the procedure of Dixon and Silver respects the advanced production of future deficits to prevent any violation of the defined capacity requirements
- Therefore, by considering each period and its production program only once, its determination always results in a program where it remains possible to fulfill the demand requirements of all subsequent periods

The procedure of Dixon and Silver

- Step 1: Initialization of variables
  - Check whether the entire problem is solvable. If

$$\forall t \in \{1,\dots,T\}: \quad \underbrace{\sum_{j=1}^{t}\sum_{k=1}^{K} to_k \cdot d_{k,j}}_{\text{Total capacity demand up to period t}} \le \underbrace{\sum_{j=1}^{t} b_j}_{\text{Total available capacity up to period t}}$$

    then the problem is solvable. Otherwise, stop with the output "Problem cannot be solved"
  - Current period is i=1

Continuation of step 1

- Initialization of the product ranges

  For all products k: $r_{k,i}=0$
- Initialization of production quantities

  For all products k: $x_{k,i}=d_{k,i}$
- Generate the respective remaining capacities in period i

$$\boxed{RC_i = b_i - \sum_{k=1}^{K} to_k \cdot d_{k,i}}$$

Step 2

- Generate the earliest period where the current production program of period *i* is not able to guarantee a feasible execution of the demanded production quantities, i.e., we compute:

$$\boxed{t_c = \min\left\{ t \mid t > i \wedge \sum_{j=i+1}^{t} CU_{i,j} < \sum_{j=i+1}^{t} CR_{i,j} \right\}}$$

Step 3

- Consider the set M of products whose current range does not cover the period t_c und whose subsequent demand can be integrated in the production of period i, i.e.,

$$\boxed{M = \left\{ k \mid r_{k,i} < t_c - i \wedge d_{k,i+r_{k,i}+1} \cdot to_k \leq RC_i \right\}}$$

- If M contains no products, go to step 4
- Otherwise, determine the product l in M with largest priority $\Delta_{l,i}$
  - If $\Delta_{l,i} \geq 0$: Integrate the demand of the next period of product l and go to step 3 – integration (next slide)
  - Otherwise, go to step 4

Step 3 – Integration

- The extension of the production quantity for product *l* to the next period is advantageous:

$$r_{l,i} = r_{l,i} + 1$$

$$x_{l,i} = x_{l,i} + d_{l,i+r_{l,i}}$$

$$RC_i = RC_i - to_l \cdot d_{l,i+r_{l,i}}$$

$$d_{l,i+r_{l,i}} = 0$$

Go to step 2

Step 4 – Feasibility check

- If $t_c > T$, then the production plan for period i is already feasible and we can switch to the next iteration by setting $i := i + 1$
- Otherwise, we have to resume adapting the production program in period i by integrating the production of future demands
- This is done in step 5

Step 5 – Feasibility construction

- Compute with Q the additional production demand for attaining a feasible constellation after period i, i.e.,

$$\boxed{Q = \max\left\{\sum_{j=i+1}^{t} CR_{i,j} - \sum_{j=i+1}^{t} CU_{i,j} \mid t_c \le t \le T\right\}}$$

Step 6 – Corrections

- Consider all products whose current range does not cover up to period t_c. In case of the k-th product we get:

$$r_{k,i}^{new} = \min\left\{ r_{k,i} + 1, r_{k,i} + \frac{Q}{to_k \cdot d_{k,i+r_{k,i}+1}} \right\}$$

If $r_{k,i}^{new}$ is an integer define the priority as follows:

$$\Delta_{k,i} = \frac{\left[c_{k,i,i+r_{k,i}}^{period} - c_{k,i,i+r_{k,i}^{new}}^{period} \right]}{to_k \cdot d_{k,i+r_{k,i}^{new}}}$$

Otherwise:

$$\Delta_{k,i} = \frac{\left[c_{k,i,i+r_{k,i}}^{period} - c_{k,i,i+r_{k,i}^{new}}^{period} \right]}{Q}$$

Step 6 – continuation

- Integrate the period demand as described above for the product with the largest Δ-priority. Let W, the respective occurring capacity, demand for this integration. Then Q:=Q-W
- If Q>0, repeat step 6 – otherwise, go to the next (i:=i+1) iteration

Example

- Two products, 4 periods to be considered
- Setup costs: s_1=100; s_2=50
- Holding costs: h_1=4; h_2=1
- Production times: to_1=to_2=1
- Capacities: b_1=b_2=b_3=b_4=160

t	1	2	3	4
$d_{1,t}$	110	49	0	82
$d_{2,t}$	48	75	15	120

Iteration i=1

- **Step 1:**
  - General feasibility check:
    - t=1: 158<160 ok
    - t=2: 282<320 ok
    - t=3: 297<480 ok
    - t=4: 499<640 ok
    - $r_{1,1}=0$; $x_{1,1}=110$ product 1
    - $r_{2,1}=0$; $x_{2,1}=48$ product 2
    - $RC_1=2$ Remaining capacity in period 1

t	1	2	3	4
$q_{1,t}$	110	-	-	-
$q_{2,t}$	48	-	-	-
$CN_{i,t}$	-	124	15	202
RC_i	2	160	160	160

Iteration i=1 – step 2

Now, we have to determine if there is a period where the feasibility is endangered by the current production plan

$$t = 2: CN_{1,2} = 49 + 75 = 124 \Rightarrow CR_{1,2} = 124 - 160 = -36$$

$$\Rightarrow \sum_{j=2}^{2} CU_{1,j} = 0 \geq \sum_{j=2}^{2} CR_{1,j} = -36 \qquad ok$$

$$t = 3: CN_{1,3} = 0 + 15 = 15 \Rightarrow CR_{1,3} = 15 - 160 = -145$$

$$\Rightarrow \sum_{j=2}^{3} CU_{1,j} = 0 \geq \sum_{j=2}^{3} CR_{1,j} = -36 - 145 = -181 \qquad ok$$

$$t = 4: CN_{1,4} = 82 + 120 = 202 \Rightarrow CR_{1,4} = 202 - 160 = 42$$

$$\Rightarrow \sum_{j=2}^{4} CU_{1,j} = 0 \geq \sum_{j=2}^{4} CR_{1,j} = -36 - 145 + 42 = -139 \qquad ok$$

$$\Rightarrow t_c = 5 > T = 4$$

Iteration i=1 – step 3

- Now, we try to enlarge the production quantities in order to reduce the costs per period
  - Unfortunately, in this case the small remaining capacity of 2 in period 1 prevents any integration
  - Demands:
    - Product 1: $d_{1,2} \cdot to_1 = 49 > 2$
    - Product 2: $d_{2,2} \cdot to_2 = 75 > 2$
  - **Iteration 1 ends**

Iteration i=2

- **Step 1:**
  - Initialization of product quantities:
    - $r_{1,2}=0$; $x_{1,2}=49$ product 1
    - $r_{2,2}=0$; $x_{2,2}=75$ product 2
    - $RC_2=36$ Remaining capacity in period 2

t	1	2	3	4
$q_{1,t}$	110	49	-	-
$q_{2,t}$	48	75	-	-
$CN_{i,t}$	-	-	15	202
RC_i	2	36	160	160

Iteration i=2 – step 2

Now, we have to determine if there is a period where the feasibility is endangered by the current production plan

$$t = 3 : CN_{2,3} = 0 + 15 = 15 \Rightarrow CR_{2,3} = 15 - 160 = -145$$

$$\Rightarrow \sum_{j=3}^{3} CU_{2,j} = 0 \geq \sum_{j=3}^{3} CR_{2,j} = -145 \qquad ok$$

$$t = 4 : CN_{2,4} = 82 + 120 = 202 \Rightarrow CR_{2,4} = 202 - 160 = 42$$

$$\Rightarrow \sum_{j=3}^{4} CU_{2,j} = 0 \geq \sum_{j=3}^{4} CR_{2,j} = -145 + 42 = -103 \qquad ok$$

$$\Rightarrow t_c = 5 > T = 4$$

Iteration i=2 – step 3

- Now, we try to enlarge the production quantities in order to reduce the costs per period
  - Product 1 has no demand in period 3. Therefore, an enlargement yields always the highest priority and is executed, i.e., $r_{1,2}=1$
  - Product 2 has in period 3 the demand 15, i.e., it holds that $d_{2,3} \cdot to_2=15<RC_2=36$
    - $\Delta_{2,2}=(50/1-(50+1 \cdot 15)/2)/15=(50-32,5)/15=17,5/15=1,16667 \geq 0$, i.e., enlargement is implemented
    - $r_{2,2}=1$

T	1	2	3	4
$q_{1,t}$	110	49	-	-
$q_{2,t}$	48	90	-	-
$CN_{i,t}$	-	-	-	202
RC_i	2	21	160	160

Iteration i=2 – step 2(2)

Now, we have to determine if there is a period where the feasibility is endangered by the current production plan

$$t = 3 : CN_{2,3} = 0 \Rightarrow CR_{2,3} = -160$$

$$\Rightarrow \sum_{j=3}^{3} CU_{2,j} = 15 \geq \sum_{j=3}^{3} CR_{2,j} = -160 \quad ok$$

$$t = 4 : CN_{2,4} = 82 + 120 = 202 \Rightarrow CR_{2,4} = 202 - 160 = 42$$

$$\Rightarrow \sum_{j=3}^{4} CU_{2,j} = 15 \geq \sum_{j=3}^{4} CR_{2,j} = -160 + 42 = -118 \quad ok$$

$$\Rightarrow t_c = 5 > T = 4$$

Iteration i=2 – step 3(2)

- Now, we again try to enlarge the production quantities in order to reduce the costs per period
  - Unfortunately, in this case the small remaining capacity of 21 in period 2 prevents any further integration
  - Demands:
    - Product 1: $d_{1,4} \cdot to_1 = 82 > 21$
    - Product 2: $d_{2,4} \cdot to_2 = 120 > 21$
  - Iteration 2 ends

Iteration i=3

- **Step 1:**
  - Initialization of product quantities:
    - $r_{1,3}=0$; $x_{1,3}=0$ product 1
    - $r_{2,3}=0$; $x_{2,3}=0$ product 2
    - $RC_3=160$ Remaining capacity in period 3

t	**1**	**2**	**3**	**4**
$q_{1,t}$	110	49	-	-
$q_{2,t}$	48	90	-	-
$CN_{i,t}$	-	-	-	202
RC_i	2	21	160	160

Iteration i=3 – step 2

Now, we have to determine whether there is a period where the feasibility is endangered by the current production plan

$$t = 4 : CN_{3,4} = 82 + 120 = 202 \Rightarrow CR_{3,4} = 202 - 160 = 42$$

$$\Rightarrow \sum_{j=4}^{4} CU_{3,j} = 0 < \sum_{j=4}^{4} CR_{3,j} = 42 \qquad \text{Bottleneck!}$$

$$\Rightarrow t_c = 4$$

$$\Rightarrow \text{We have to adapt the current plan!}$$

Iteration i=3 – step 3

- Now, we try to enlarge the production quantities in order to reduce the costs per period
  - Product 1 has in period 4 the demand 82, i.e., it holds $d_{2,4} \cdot to_2 = 82 < RC_3 = 160$

    $\Delta_{1,3} = (0/1 - (100 + 4 \cdot 82)/2)/82 = (-214)/82 =$

    -2,609, i.e., enlargement is not implemented
  - Product 2 has in period 4 the demand 120, i.e., it holds $d_{2,4} \cdot to_2 = 120 < RC_3 = 160$

    $\Delta_{2,3} = (0/1 - (50 + 1 \cdot 120)/2)/120 = (-85)/120 =$

    -0,708333<0, i.e., enlargement is not implemented

Iteration i=3 – step 4

Now, we have to determine whether there is a period where feasibility is endangered by the current production plan

$$t = 4: \;\; CN_{3,4} = 82 + 120 = 202 \Rightarrow CR_{3,4} = 202 - 160 = 42$$

$$\Rightarrow \sum_{j=4}^{4} CU_{3,j} = 0 < \sum_{j=4}^{4} CR_{3,j} = 42$$ Bottleneck!

$$\Rightarrow t_c = 4$$

$\Rightarrow$ We have to adapt the current plan!

Iteration i=3 – step 5

- Q=42: Minimal capacity to be integrated in period is 3 in order to guarantee a feasible production plan
- Respective ranges:
  - Product 1: 42/82=0,51

    $\Delta_{1,3}$=(0/1-(100+4·42)/1,51)/42=-4,2258
  - Product 2: 42/120=0,35

    $\Delta_{2,3}$=(0/1-(50+1·42)/1,35)/42=-1,6223, i.e., enlargement is implemented

    $r_{2,3}$=0,35; $x_{2,3}$=42

Iteration i=3 – step 5

t	1	2	3	4
$q_{1,t}$	110	49	-	-
$q_{2,t}$	48	90	42	-
$CN_{i,t}$	-	-	-	160
RC_i	2	21	118	160

- Iteration 3 ends

Iteration i=4

- **Step 1:**
  - Initialization of product quantities:
    - $r_{1,4}=0$; $x_{1,4}=82$ product 1
    - $r_{2,4}=0$; $x_{2,4}=78$ product 2
    - $RC_4=0$ Remaining capacity in period 4

t	**1**	**2**	**3**	**4**
$\mathbf{q_{1,t}}$	110	49	-	82
$\mathbf{q_{2,t}}$	48	90	42	78
$\mathbf{CN_{i,t}}$	-	-	-	-
$\mathbf{RC_i}$	2	21	118	0

Iteration i=4 – step 2

Now, we have to determine if there is a period where the feasibility is endangered by the current production plan

$$\boxed{\Rightarrow t_c = 5 \geq T = 4}$$

The algorithm stops! Solution generated!

t	1	2	3	4
$q_{1,t}$	110	49	-	82
$q_{2,t}$	48	90	42	78
RC_i	2	21	118	0

Can we improve the solution?

- Idea: It is always advantageous to produce all quantities in the last possible period
- This is not implemented for product 2
- We can move the production of 15 units needed in 3 in this period to save holding costs of 15 currency units, i.e., we generate the solution:

t	1	2	3	4
$q_{1,t}$	110	49	-	82
$q_{2,t}$	48	75	57	78
RC_i	2	36	103	0

Observation

- In literature, it is stated that the procedure of Dixon and Silver yields a high solution quality
- Consequently, this procedure is also used in multiple-stage problems as a subroutine

3.5 The CLSP*L* Model

- All assumptions of the CLSP beside the carry-over-prohibition of setup states are valid
- A setup state is not lost if there is no production on the resource within a bucket
- Single-item production is possible (i.e., the conservation of one setup state for the same product over two consecutive bucket boundaries)

3.5.1 CLSP*L* – Attributes

- The planning horizon T is fixed and divided into time buckets 1,…,T
- Resource consumption to produce a product j on a specific resource m is fixed, and there exists an unique assignment of products to resources
- Setups incur setup costs and consume setup time, thereby reducing capacity in periods where setups occur
- At most one setup state can be carried over on each resource to the next one, consequently no setup activity is necessary in this subsequent period
- Single-item production is possible (i.e. the conservation of one setup state for the same product over two consecutive bucket boundaries)
- A setup state is not lost if there is no production on the respective resource within a bucket
- In the following, we give a detailed mathematical definition of the problem basing on the model proposed by Stadtler and Suerie (2003)

Computation of the net-demands

- In the CLSP*L* introduced here the chosen lot sizes are defined according to the net demands for product j in period t, i.e., we define the proportion of the net demand of a specific product in period t that is satisfied by the production in the considered period.
- This is done in order to get a more strict and compact model definition which can be solved much easier
- To do so, we first have to introduce what we understand as the so called **net demand of a specific product in a defined period**
  - Up to now we have modeled the inventory and gross demands directly within separated variables (derived variables)
  - So far, we have neglected dependencies resulting from **multiple-stage systems**

Computation of the net-demands

- Now, the relative definition requires a detailed handling of these interdependencies. Therefore, we have to derive the net demands instead.
- Consequently, inventory and secondary demands have to be respected
- First of all, we have to map the product structure with all existing interdependencies
- Note that **ending inventory is explicitly allowed**

Generating net demands – Parameters

J : Number of products (or items)

T : Number of considered periods

$\forall j \in \{1,...,J\} : \forall t \in \{1,...,T\} : P_{j,t}$: Primary gross demand of product j in period t

$\forall j \in \{1,...,J\} : \forall t \in \{1,...,T\} : D_{j,t}$: Gross demand of product j in period t

$\forall j \in \{1,...,J\} : \forall t \in \{1,...,T\} : D^n_{j,t}$: Net demand of product j in period t

$\forall i \in \{2,...,J\} : \forall j \in \{1,...,i-1\} : r^n_{i,j}$: The number of units of product (item) i required to produce one unit of product (item) j

In what follows, we assume that the products are ordered according to the adjacency graph, i.e., a lower numbered product is never necessary in order to produce a higher numbered one

Generating net demands – Parameters

Therefore, we can generate the net demands starting with the lowest numbered product which has no successor, i.e.

$\forall j \in \{1,\ldots,J\} : \forall t \in \{1,\ldots,T\}:$

$$\underbrace{D_{j,t}}_{\text{Gross demand}} = \underbrace{P_{j,t}}_{\text{Primary demand}} + \underbrace{\sum_{k=1}^{j-1} r_{j,k}^{d} \cdot D_{k,t}}_{\text{Gross demand of successors}}$$

and

$\forall j \in \{1,\ldots,J\} : \forall t \in \{1,\ldots,T\} : \delta = I_{j,0}:$

$$D_{j,t}^{n} = \max\left\{0, P_{j,t} + \sum_{k=1}^{j-1} r_{j,k}^{d} \cdot D_{k,t}^{n} - \delta\right\} \text{ with } \underbrace{\delta = \max\left\{0, \delta - P_{j,t} - \sum_{k=1}^{j-1} r_{j,k}^{d} \cdot D_{k,t}^{n}\right\}}_{\text{New inventory of product } j \text{ in period } t}$$

CLSP*L* – Parameters

$j = 1,\ldots,J$: Product index or item index

$m = 1,\ldots,M$: Resource index

$t = 1,\ldots,T$: Index of periods

$R_m \left(1 \le m \le M\right)$: Set of products produced on resource m

$a_{m,j} \left(1 \le m \le M; 1 \le j \le J\right)$: Capacity needed on resource m to produce one unit of item j

$B_{j,t} \left(1 \le j \le J; 1 \le t \le T\right)$: Large number, not limiting feasible lot sizes of product j in period t

$C_{m,t} \left(1 \le m \le M; 1 \le t \le T\right)$: Available capacity of resource m in period t

$h_j \left(1 \le j \le J\right)$: Holding cost for one unit of product unit j per period

CLSP*L* – Parameters

$P_{j,t}\left(1 \le j \le J; 1 \le t \le T\right)$: Primary, gross demand for item j in period t

$\left(\text{with } P_{j,T} \text{ including final inventory - if given for the planning horizon } T\right)$

$D_{j,t}\left(1 \le j \le J; 1 \le t \le T\right)$: Gross demand for item j in period t

$D^n_{j,t}\left(1 \le j \le J; 1 \le t \le T\right)$: Net demand for item j in period t

$sc_j\left(1 \le j \le J\right)$: Setup cost for product j

$st_j\left(1 \le j \le J\right)$: Setup time for product j

$S_j\left(1 \le j \le J\right)$: Set of direct successors of product j in the multilevel product structure

$r^d_{j,k}\left(1 \le j \le J; 1 \le k \le j-1\right)$: Units of items j necessary to produce one unit of the direct successor item k

l: Lead time offset $\left(\text{in the following assumed to be } 0\right)$

CLSP*L* – Variables

$I_{j,t}\left(1\leq j\leq J;1\leq t\leq T\right)$: Inventory of item or product j at the end of the period t

$Z_{j,t,s}\left(1\leq j\leq J;1\leq t\leq T;t\leq s\leq T\right)$: Proportion of net demand of product j in period s fulfilled by production in period t

$X_{j,t}\left(1\leq j\leq J;1\leq t\leq T\right)$: Production amount of item or product j in period t

$$\left(\text{sought lot size}\right)\Rightarrow\forall j\in\left\{1,\ldots,J\right\}:\forall t\in\left\{1,\ldots,T\right\}:X_{j,t}=\sum_{s=t}^{T}D_{j,s}^{n}\cdot Z_{j,t,s}$$

$Y_{j,t}\left(1\leq j\leq J;1\leq t\leq T\right)$: Derived binary setup variable

$$=\begin{cases}1\text{: if a setup for item } j \text{ is performed in period } t;\\ 0\text{:otherwise}\end{cases}$$

$W_{j,t}\left(1\leq j\leq J;1\leq t\leq T\right)$: Binary linkage variable indicating that a setup state for product j is carried over from period $t-1$ to period t

$Q_{m,t}\left(1\leq m\leq M;1\leq t\leq T\right)$: Binary variable indicating that production on resource m in period t is limited to a single product, and there is no setup activity necessary, i.e., the setup state is linked from the preceding to the subsequent period

CLSP*L* – Restrictions

$$\forall m \in \{1,\dots,M\}: \forall t \in \{1,\dots,T\}: \sum_{j \in R_m} \sum_{s=t}^{T} a_{m,j} \cdot D_{j,s}^{n} \cdot Z_{j,t,s} + \sum_{j \in R_m} st_j \cdot Y_{j,t} \le C_{m,t} \qquad (1)$$

(Capacity restrictions)

$$\forall j \in \{1,\dots,J\}: \forall t \in \{1,\dots,T\}: \forall s \in \{t,\dots,T\}: Z_{j,t,s} \le Y_{j,t} + W_{j,t} \qquad (2)$$

(Dependency between production and setup and linkage states)

$$\forall j \in \{1,\dots,J\}: \forall t \in \{1,\dots,T\}: \forall s \in \{t,\dots,T\}: Z_{j,t,s} \ge 0 \wedge Y_{j,t} \in \{0,1\} \qquad (3)$$

(Co - domain definition)

CLSP*L* – Restrictions

$$\forall j \in \{1,...,J\} : \forall t \in \{1,...,T\} : \left(D^n_{j,t} > 0\right) : \sum_{s=1}^{t} Z_{j,s,t} = 1 \qquad (4)$$

(Demand fulfillment)

$$\forall m \in \{1,...,M\} : \forall t \in \{2,...,T\} : \sum_{j \in R_m} W_{j,t} \leq 1 \qquad (5)$$

(At most one setup state can be linked per time period and resource)

$$\forall j \in \{1,...,J\} : \forall t \in \{2,...,T\} : W_{j,t} \leq Y_{j,t-1} + W_{j,t-1} \qquad (6)$$

(Dependencies between setup activities and linkage variables)

CLSP*L* – Restrictions

$$\forall m \in \{1,...,M\} : \forall j \in R_m : \forall t \in \{1,...,T-1\} : W_{j,t} + W_{j,t+1} \leq 1 + Q_{m,t} \qquad (7)$$

(Dependencies between different sets of linkage variables)

$$\forall m \in \{1,...,M\} : \forall j \in R_m : \forall t \in \{1,...,T\} : Y_{j,t} + Q_{m,t} \leq 1 \qquad (8)$$

(Dependencies between different sets of linkage variables)

$$\forall m \in \{1,...,M\} : \forall t \in \{1,...,T-1\} : Q_{m,t} \geq 0 \wedge Q_{m,1} = 0 \wedge Q_{m,T} = 0 \qquad (9)$$

(Co-domains of variable)

$$\forall j \in \{1,...,J\} : \forall t \in \{1,...,T\} : W_{j,t} \in \{0,1\} \wedge W_{j,1} = 0 \qquad (10)$$

(Co-domains of linkage variable)

CLSP*L* – Objective function

$$\text{Minimize } Z = \underbrace{\sum_{j=1}^{J}\sum_{s=1}^{T-1}\sum_{t=s}^{T} h_j \cdot (t-s) \cdot D^n_{j,t} \cdot Z_{j,s,t}}_{\text{Holding costs}} + \underbrace{\sum_{j=1}^{J}\sum_{t=1}^{T} sc_j \cdot Y_{j,t}}_{\text{Setup costs}}$$

3.5.2 Tightening the model

- Suerie and Stadtler (2003) propose several extensions of the defined model in order to strengthen it significantly
- Strengthen means that it becomes possible to derive tighter LP bounds
- In particular…
  - new variables are added
  - and three groups of valid inequalities are introduced

Added / exchanged variables

- The resource-dependent variables $Q_{m,t}$ are replaced by product-dependent ones termed as $QQ_{j,t}$
- By using these modified variables instead we can give a more precise definition of occurring setup states linked between subsequent periods
- In detail we define:

$\forall j \in \{1,...,J\} : \forall t \in \{1,...,T\} : QQ_{j,t}$: Binary decision variable. Is one iff the setup state is carried from period t - 1 through $t+1$ while product j is solely produced in period t

Erasing restriction 8

$$\forall m \in \{1,\ldots,M\} : \forall j \in R_m : \forall t \in \{1,\ldots,T\} : Y_{j,t} + Q_{m,t} \le 1 \qquad (8)$$

(Dependencies between different sets of linkage variables)

is replaced by

$$\forall m \in \{1,\ldots,M\} : \forall j \in R_m : \forall t \in \{1,\ldots,T\} : Y_{j,t} + W_{j,t} + \sum_{\substack{k \in R_m \\ k \neq j}} QQ_{k,t} \le 1 \qquad (8a)$$

Note that there can be either a setup activity for item j in period t $(Y_{j,t} = 1)$, a link for item j into period t $(W_{j,t} = 1)$, a single item production for any item $k \neq j$ in period t, or none of those three options, but never two of them simultaneously.

Erasing restriction 6

$$\forall j \in \{1,\ldots,J\} : \forall t \in \{2,\ldots,T\} : W_{j,t} \leq Y_{j,t-1} + W_{j,t-1} \qquad (6)$$

is replaced by

$$\forall j \in \{1,\ldots,J\} : \forall t \in \{2,\ldots,T\} : W_{j,t} \leq Y_{j,t-1} + QQ_{j,t-1} \qquad (6a)$$

Note there can be setup state carried over in period t only if either item j was set up in period t-1 $(Y_{j,t-1} = 1)$ or the setup state is already carried over from period t-2 to t-1 and there is a single item production in period t-1 $(QQ_{j,t-1} = 1)$.

Range-restriction of values for QQ

$$\forall j \in \{1,\ldots,J\} : \forall t \in \{2,\ldots,T-1\} : \forall s \in \{t, t+1\} :$$
$$QQ_{j,t} \leq W_{j,s} \qquad (9)$$

$$\wedge$$

$$\forall j \in \{1,\ldots,J\} : \forall t \in \{1,\ldots,T-1\} : QQ_{j,t} \geq 0$$
$$(QQ_{j,1} = 0 \wedge QQ_{j,T} = 0) \qquad (10)$$

Restriction 7

$$\forall m \in \{1,\dots,M\} : \forall j \in R_m : \forall t \in \{1,\dots,T-1\} : W_{j,t+1} + W_{j,t} \leq 1 + Q_{m,t} \qquad (7)$$

is replaced by

$$\forall j \in \{1,\dots,J\} : \forall t \in \{1,\dots,T-1\} : W_{j,t+1} + W_{j,t} \leq 1 + QQ_{j,t} \qquad (7a)$$

BUT:

Is this restriction really necessary to define the model?

Restriction 7

$$\forall j \in \{1,\ldots,J\} : \forall t \in \{1,\ldots,T-1\}:$$

$$W_{j,t+1} + W_{j,t} \quad \left(\text{using}: (6\text{a}): W_{j,t} \le Y_{j,t-1} + QQ_{j,t-1}\right)$$

$$\le Y_{j,t} + QQ_{j,t} + W_{j,t} \quad \left(\begin{array}{l} \text{using}: (8\text{a}): Y_{j,t} + W_{j,t} + \sum_{\substack{k \in R_m \\ k \ne j}} QQ_{k,t} \le 1 \\ \Rightarrow Y_{j,t} + W_{j,t} \le 1 - \sum_{\substack{k \in R_m \\ k \ne j}} QQ_{k,t} \end{array} \right)$$

$$\le QQ_{j,t} + 1 - \sum_{\substack{k \in R_m \\ k \ne j}} QQ_{k,t} \le 1 + QQ_{j,t}$$

$$\Rightarrow W_{j,t+1} + W_{j,t} \le 1 + QQ_{j,t} \quad (7\text{a})$$

Observation

- **Restriction 7 can be erased** due to the combined application of restrictions 6 and 8
- By analyzing the transformations on the previous slide, it becomes obvious that the restrictions 6 and 8 together form restrictions that are considerably tighter than the restriction 7

Valid inequalities

- In the following, additional restrictions are introduced to achieve a further tightening of the model definition
- To do so, basic attributes of adequate solutions are elaborated and subsequently fixed by the integration of additional restrictions in the model definition

Preprocessing – Inequalities

- Now attributes of the given test data are used to define additional restrictions
- In detail, the possible range of the new introduced QQ-variables is limited
- This can be done in a step called **preprocessing**
- Therefore, in this preprocessing step available capacities are computed and compared with the cumulative slack capacities summed up to the respective period
- Since there is no backlog allowed, impossible single item productions in some periods may be identified and, therefore, excluded

Example

Item j	$a_{m,j}$	Net demand in period 1	Net demand in period 2	Net demand in period 3
1	1	20	20	20
2	1	30	40	40
3	1	20	20	20
Available capacity		100	100	100
Cumulative slack capacity		30	50	

Observations

- **Period 2**

  Single item production is not possible at all

  Why?

  – Necessary is a capacity requirement shift of at least 40 units to period 1

  – But: In period 1 there is a slack capacity of only 30 units

- **Period 3**

  Single item production is not possible for products 1 and 3

  Why?

  – Necessary is a capacity requirement shift of at least 60 units to period 2

  – But: In period 2 there is a cumulative slack capacity of only 50 units

General speaking

- Let U denoting the length of the interval under consideration:

  "If cumulative slack capacity (up to period t-1) is less than the amount that has to be pre-produced to allow single-item production of just one product in the interval under consideration [t; t+U-1], then at least two products have to be produced in the interval [t; t+U-1]"

- This implies that at least one setup activity has to be performed, which implies that not all periods of the interval [t; t+U-1] can have single-item production

Additional model restrictions – Type 1

$$\forall m \in \{1,\ldots,M\} : \forall j \in R_m : \forall U \in \{1,2,3\} : \forall t \in \{2,\ldots,T-U+1\} :$$

$$\text{if} : \underbrace{\sum_{s=1}^{t-1} C_{m,s} - \sum_{s=1}^{t-1} \sum_{k \in R_m} a_{m,k} \cdot D_{k,s}^{n} - \sum_{k \in R_m, \text{ with: } \sum_{s=1}^{t-1} D_{k,s}^{n} > 0} st_k}_{\text{Slack capacity in periods 1 to } t\text{-1}}$$

$$- \underbrace{\sum_{s=t}^{t+U-1} \sum_{k \in R_m, k \neq j} a_{m,k} \cdot D_{k,s}^{n}}_{\text{Necessary capacity in 1 to } t\text{-1 for single item production of product type j in periods } t,\ldots,t+U\text{-1}} < 0 :$$

$$\sum_{s=t}^{t+U-1} QQ_{j,s} \leq U - 1 \qquad (11)$$

Additional model restrictions – Type 2 (ext)

$$\forall m \in \{1,\ldots,M\}: \forall V \in \{1,2,3\}: \forall t \in \{2,\ldots,T-V+1\}:$$

$$\text{if}: \underbrace{\sum_{s=1}^{t-1} C_{m,s} - \sum_{s=1}^{t-1}\sum_{k\in R_m} a_{m,k}\cdot D^n_{k,s} - \sum_{k\in R_m,\ \text{with: } \sum_{s=1}^{t-1} D^n_{k,s}>0} st_k}_{\text{Slack capacity in periods 1 to } t\text{-1}}$$

$$- \underbrace{\sum_{s=t}^{t+V-1}\sum_{j\in R_m} a_{m,j}\cdot D^n_{j,s}}_{\substack{\text{Necessary capacity in 1 to } t\text{-1 for the production of} \\ \text{all items in the subsequent periods } t,\ldots,t+V\text{-1}}} + \underbrace{\max_{j\in R_m}\left(\sum_{s=t}^{t+V-1} a_{m,j}\cdot D^n_{j,s}\right)}_{\substack{\text{Maximal capacity remaining} \\ \text{in periods } t \text{ to } t+V\text{-1}}} < 0:$$

$$\sum_{s=t}^{t+V-1}\sum_{j\in R_m} QQ_{j,s} \le V-1 \qquad (12)$$

Inventory / Setup – Inequalities

- If $Y_{j,t}=W_{j,t}=0$ for product j, there is no production in t for product j and therefore the stock has to satisfy the occurring demand
- These dependencies can be generalized to intervals of the periods t to t+p
- Therefore, we can add the following restrictions to the model

Additional restrictions

$$\forall j \in \{1,\dots,J\} : \forall t \in \{1,\dots,T-1\} : \forall p \in \{1,\dots,T-t\} :$$

$$\underbrace{I_{j,t-1} + \sum_{k \in S_j} r_{j,k} \cdot I_{k,t-1}}_{\text{Total quantities of } j \text{ already in stock in } t\text{-1}} \geq \underbrace{\sum_{s=t}^{t+p-1} D^n_{j,s}}_{\substack{\text{Total net demand in} \\ \text{the interval } t \text{ to } t+p-1}} \cdot \underbrace{\left(1 - W_{j,t} - \sum_{r=t}^{s} Y_{j,r}\right)}_{\substack{\text{1 iff, no linking or setup operation} \\ \text{takes place for product type } j \\ \text{in the periods } t \text{ to } s}}$$

with:

S_j : Set of successor items $\left(\text{direct or indirect}\right)$ of item j

Capacity/Single-Item – Inequalities

- Now, additional restrictions are defined which map the capacity consequences of an occurred single item production
- Therefore, it is distinguished whether there is a single item production on a considered resource or not
  - In the first case we can significantly strengthen the existing capacity restriction
  - In the latter case the original capacity restriction remains

Additional restrictions

$$\forall m \in \{1,\ldots,M\} : \forall t \in \{2,\ldots,T-1\} :$$

$$\underbrace{\sum_{j \in R_m} \left(a_{m,j} \cdot X_{j,t} + st_j \cdot Y_{j,t} \right)}_{\text{Total capacity demand on resource m}}$$

$$\leq C_{m,t} \cdot \underbrace{\left(1 - \sum_{j \in R_m} QQ_{j,t} \right)}_{=1 \text{ iff there is no single item production of resource } m} + \underbrace{\sum_{j \in R_m} a_{m,j} \cdot XQ_{j,t}}_{\text{Demand of the single item production on resource } m}$$

New derived variables

$\forall j \in \{1,...,J\} : \forall t \in \{2,...,T-1\} : XQ_{j,t}$:

Production quantity of item j in period t, if this is a single-item production period, i.e., we have to add the following restrictions:

$$\forall j \in \{1,...,J\} : \forall t \in \{2,...,T-1\} : XQ_{j,t} \leq X_{j,t}$$

$$\forall j \in \{1,...,J\} : \forall t \in \{2,...,T-1\} : XQ_{j,t} \leq \min\left(\underbrace{\frac{C_{m,t}}{a_{m,j}}}_{\text{Capacity of resource } m}, \underbrace{\sum_{s=t}^{T} D^{n}_{j,s}}_{\text{Maximal demand in } t} \right) \cdot QQ_{j,t}$$

Solution approaches for the CLSP*L*

- Suerie and Stadtler use a standard MIP solver (XPRESS-MP, Release 12)
- They apply two different variants
  - **Branch & Cut:**
    - The additional restrictions are omitted in the initial model formulation which is solved in each node of the solution tree
    - However, the restrictions are stored in a *cut pool*. If a found solution violates such a restriction this restriction is subsequently added to the model
  - **Cut & Branch:**
    - All additional restrictions are inserted in the model and therefore respected in each node by the computed solutions
    - By doing so the LP becomes more restrictive

Observations

- Branch & Cut yields smaller matrices and faster solution times at each node at the price of some separation procedure
- On the other hand, both might require immense amounts of memory and time
- Therefore, a heuristic modified version of the procedures has been applied

3.5.3 Time-oriented decomposition heuristic

- Stadtler has applied this version already to the MLCLSP (Stadtler (2003))
- Main characteristics
  - The time horizon is separated into three parts
    - The lot-sizing window,
    - the time intervals preceding the window and finally
    - the time intervals following the window
  - In successive planning steps, the lot-sizing window is moved through the planning horizon

Decisions in the parts …

- Lot-sizing window:
  - Only in this part lot-sizing decisions dealing with binary variables are considered
- Preceding part:
  - Binary setup variables are fixed and cannot be changed at all
- Following part:
  - Only inventory balance and capacity constraints (without the inclusion of setup times) are included in the model definition to anticipate future capacity bottlenecks

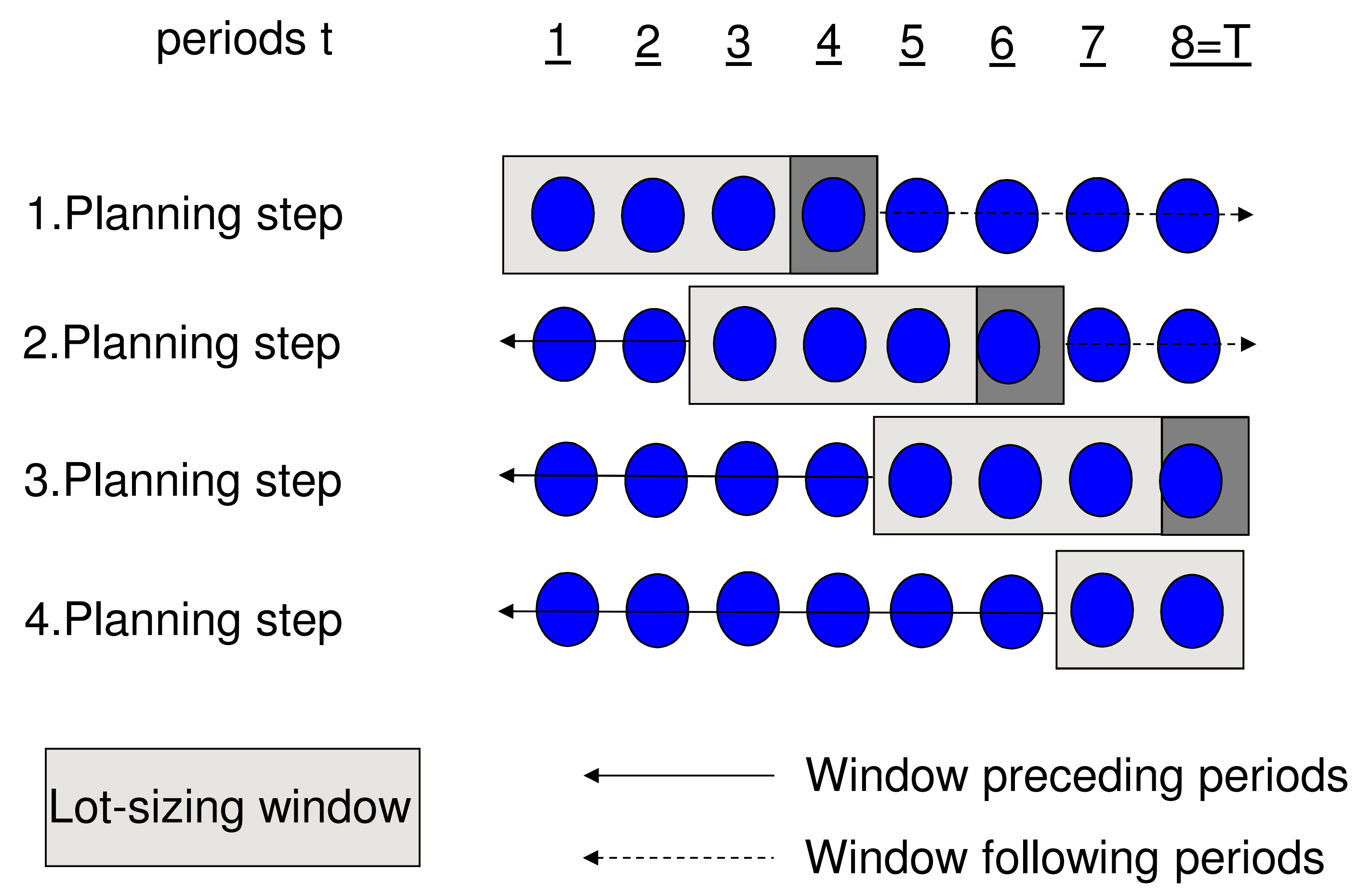
- Objective function:
  - Minimization of setup- and inventory holding costs **up to the end of the lot-sizing window**

Idea

- Finding of a tight model formulation inside a variable lot-sizing window, gathering their benefits without accepting the drawback of an inflated matrix, if such a model formulation is used for the whole planning horizon
- Parameters ((Δ,Ψ,Φ)-setting)
  - Δ: Length of the lot-sizing window
  - Ψ: Overlap of two consecutive lot-sizing windows
  - Φ: Number of periods at the end of the lot-sizing window with relaxed integrality constraints in respect of the setup variables

  i.e. $\Phi \leq \Psi \leq \Delta$

(4/2/1) setting


periods t
1
2
3
4
5
6
7
8=T
1.Planning step
2.Planning step
3.Planning step
4.Planning step
Lot-sizing window
Window preceding periods
Window following periods

Planning horizon effect

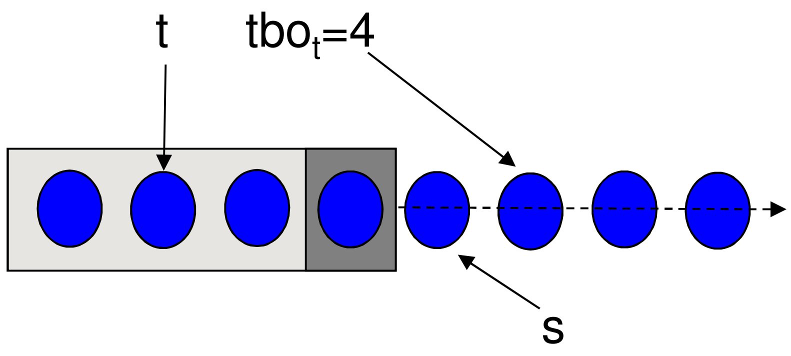
- Due to the fact that the objective function measures costs to the end of the lot-sizing-window only, possible enlargements of the production quantities at the end of the window are quite unlikely, since
  - they cause additional setup- and production costs but do not result in
  - any savings …
- Therefore, in order to deal with this problem, Suerie and Stadtler propose a bonus concept rewarding productions at the end of the planning horizon.
- Overlapping of lot-size windows also reduces the planning horizon effect

Bonus computation

- First, i.e., as an **offline processing step**, we execute the Silver Meal heuristic on the non-capacitated version of the problem for each period.
- Therefore, we get myopic TBO (time between orders) tbo_t for every period t
- If a production quantity in period t is enlarged to cover up to period s, we charge the total costs C(t,s) defined below
- In this situation, we assume that there is a current lot-sizing window starting at period T_{fix} and ending in period T_{int}
- Note that we assume that s is somewhere between the end of the window and the current tbo_t, i.e., we want to give a bonus only to enlargements likely to be prevented by the horizon effect

Bonus computation

Current planning step



$$\forall t \in \{T^{fix}+1,\dots,T^{\text{int}}\}: \forall s \in \{T^{\text{int}}+1,\dots,t+tbo_t-1\}:$$

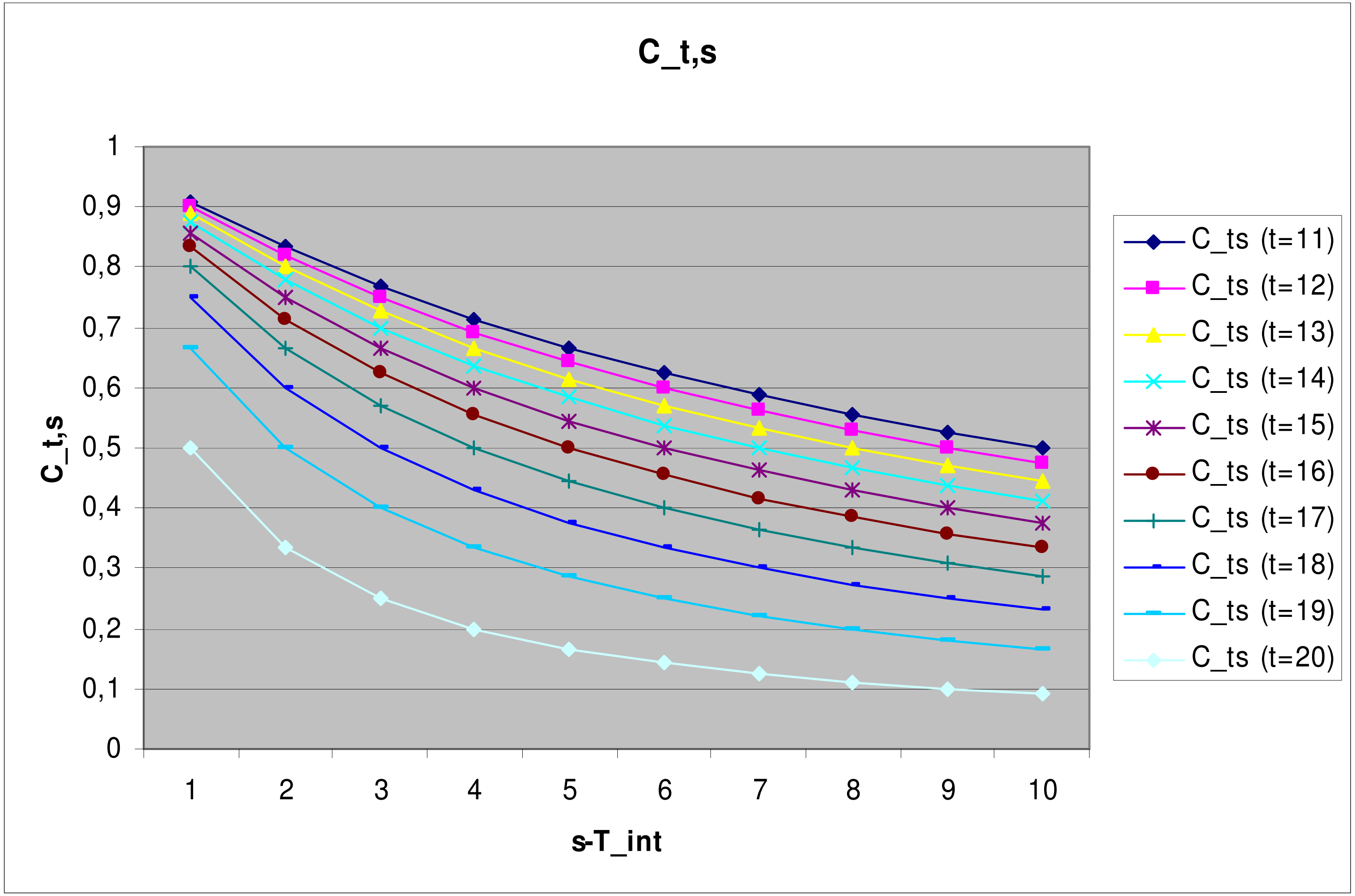
$$(\text{if } t+tbo_t > T^{\text{int}}):$$

$$C(t,s) = \underbrace{T^{\text{int}}-t+1}_{\text{Periods to the end of the window}} \Big/ \underbrace{s-t+1}_{\text{Periods between } t \text{ and } s} \cdot (\text{Costs for setup and holding})$$

Bonus for enlarging

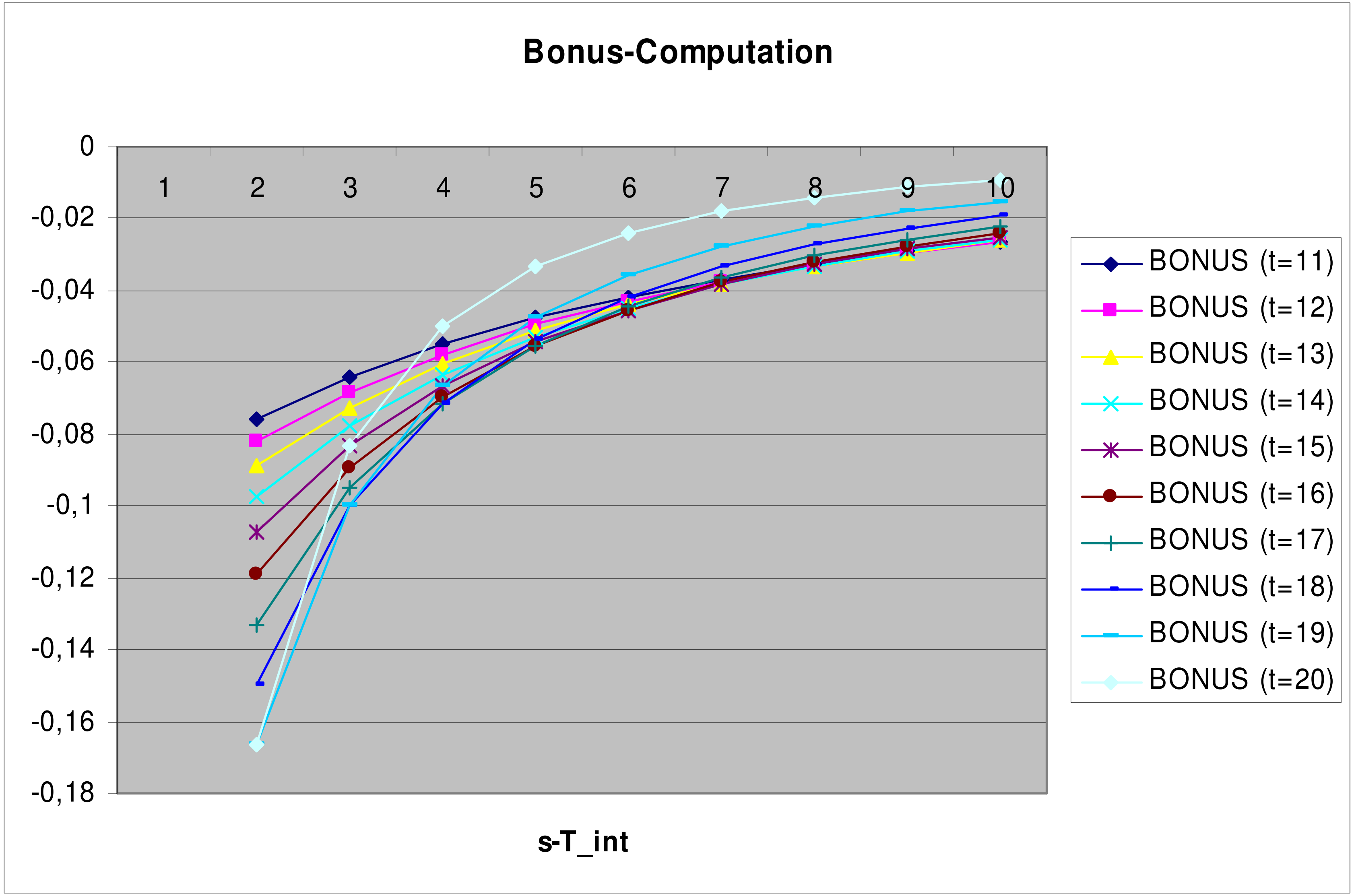
$$\text{BONUS}_{t,s} = C(t,s) - C(t,s-1)$$

Example



T_{int}=20; T_{fix}=10; Window size=10

Example



Bonus-Computation
0
-0,02
-0,04
-0,06
-0,08
-0,1
-0,12
-0,14
-0,16
-0,18
1
2
3
4
5
6
7
8
9
10
s-T_int
BONUS (t=11)
BONUS (t=12)
BONUS (t=13)
BONUS (t=14)
BONUS (t=15)
BONUS (t=16)
BONUS (t=17)
BONUS (t=18)
BONUS (t=19)
BONUS (t=20)

Feasibility of capacity demands

- By introducing the inventory balancing constraints for all periods T_{int}, …, T following the lot-sizing window the general feasibility of the generated sub-solution should be preserved
- In periods following the lot-sizing window only continuous production quantities can be chosen while the total capacity in each period can be extended by overtime that is charged by a predefined rate per time unit in the objective function

Estimating setup times

- Unfortunately, setup activities in these periods following the lot-sizing window are not planned explicitly and therefore **unknown in respect of there capacity requirements**. We only model the balance restriction as specific flow requirements resulting in production quantities
- But to anticipate future capacity bottlenecks, **different variants for estimating the occurring setup times are tested**, itemized subsequently

Estimating setup times - ST^{MIN}

- This version do not reduces the available capacity by any setup activity to be executed
- I.e. this version neglects all capacity consumptions due to setup times in periods following the lot-sizing window
- I.e., somehow a “best case consideration”
- Problem:
  - Underestimation of capacity requirements

Estimating setup times - ST^{MAX}

- This version assumes that all items have to be produced in every period, i.e. we have to setup all resources in each period
- I.e. in this version available capacity per period is reduced by the sum of setup times of item producible on the specific machine
- Consequently, if capacities are tight, infeasible problems for one or more planning steps will sometimes emerge, resulting in no solution for the complete problem
- I.e., somehow a “worst case consideration”
- Problem:
  - Overestimation of capacity requirements

Estimating setup times - ST^E

- This version lays somewhere between the extreme cases itemized above
- Capacity losses due to setups are estimated by their average consumption that is implemented in the periods preceding the lot-sizing window plus a predefined safety margin

Computational results

- All following results are measured on a PC (Windows NT 4.0) with Pentium IV 1.7 GHz microprocessor, and 256 MB RAM.
- As a MIP solver, XPRESS-MP release 12 with standard setting is used

Used approaches

1. **Basic:** Most simple version using the basic model definition without any extensions (extended formulation & valid inequalities)
2. **Extended:** Using the extended formulation but still omits the valid inequalities
3. **C&B:** Uses the valid inequalities additionally, Cut & Branch approach as described above
4. **B&C:** Uses the valid inequalities additionally, Branch & Cut approach as described above

Single-Level Test Instances

- First experiments were done by testing the different approaches on famous benchmarks proposed in literature
- In the first phase the version ST^{MAX} was proven to be not advantageous and is therefore discarded for the rest of the evaluations

Instances – Single Level

Class	#Products	#Periods	#Instances
1	6	15	116
2	6	30	5
3	12	15	5
4	12	30	5
5	24	15	5
6	24	30	5
7	10	20	180
8	20	20	180
9	30	20	180

Results for class 1

- 10 seconds computational time per experiment
- Best solution found so far is taken as the result
- It can be observed, that the proposed model formulation with valid inequalities not only yields better solutions but also better lower bounds
- Independent from the version – B & C or C & B – the yielded **solution quality of these approaches was significantly higher than the solution of the results of the standardized versions**

Results for class 1

Approach	Gap to LB	Avg. time first solution
Basic	6,26 %	0,11 sec
Extended	3,94 %	1,19 sec
C & B	2,72 %	2,66 sec.
B & C	2,62 %	2,34 sec.

Branch & Cut

- Giving additionally at most 600 seconds per each of the 116 instances the performance of the best approach the Branch & Cut procedure is tested in more detail
- In 91 cases the optimality of the best found solution could be proven in the given time limit

Parameters

- For the MIP formulations, the solution after 30 seconds is taken for classes 1-3 and 5, whereas 60 seconds of computational time are allowed for class 4 and 6-9
- For some experiments no solution was attained
- Therefore, the limit is enlarged until the first valid constellation could be generated
- Sometimes up to 20 minutes were necessary
- As LB the LP relaxation after automatic cut generation of the extended model with valid inequalities is chosen
- In contrast, the time-oriented decomposition heuristic provides excellent solutions in a very short time interval, which shows the effectiveness of the model decomposition

All classes – Heuristic comparison

	Branch & Cut		Heuristic (6/2/2, ST^{MIN})	
Classes	**Gap to LB**	**Avg. time**	**Gap to LB**	**Avg. time**
1,2	2,18 %	22 sec	2,52 %	5,3 sec
3,4	1,12 %	45 sec	0,84 %	9 sec
5,6	0,36 %	52,4 sec	0,42 %	11,2 sec
7-9	1,64 %	142,9 sec	2,69 %	13,3 sec

Observations

- Surie and Stadtler reports comparisons to the new Tabu Search procedure proposed by Gopalakrishnan et al. (2001) and conclude that their decomposition heuristic outperforms this approach according to solution quality as well as to computational time
- But the approach was **not tested on the same computational system.** However, they only report the results of this reference achieved on a Pentium III, 550 MHz system. This restricts the meaning of this conclusion significantly

Modified Single-Level Test Instances

- In classes 7-9, the impact of the CLSPL is rather poor, since **only a single from 30 setup states is carried over a period**
- Feature to carry over one setup state over two consecutive bucket boundaries is **never used**
- One answer could be, the CLSP*L* should be applied if **only a few items require one resource** and/or **some of them are long runners, whereas demand for the other items is rather low**
- For its evaluation, further test instances were generated additionally
- Owing to executed aggregations these instances are characterized by significantly smaller sets of items to be produced on the resources
- Again, 60 seconds computational time are allowed per instance

Main results

- It can be observed that the option to carry over a setup state over two consecutive periods is now used frequently
- In detail, there are 3.9 single-item productions per periods on average
- The new test instances were more difficult to solve on the average due to a larger average gap to LB
- Again, B & C was the best approach, but the heuristic reaches nearly the same solution quality while consuming significantly less computational time

Multiple-Level Test Instances

- Further multiple level instances were tested
- Time limit 600 seconds for finding a solution
- 60 instances comprising the production of 10 products on 3 resources over 24 periods each

Results

	Branch & Cut	(6/2/2) Time limit 60 seconds		(6/2/2) Time limit 180 seconds		(4/2/2) Time limit 60 seconds	
Test set	Gap to LB	Gap to LB	Avg. time	Gap to LB	Avg. time	Gap to LB	Avg. time
B+	37,5 %	32,2 %	53,2 sec	29,6 %	139,5 sec	29,1 %	38,7 sec

Results

- The heuristic approaches now outperforms the Branch & Cut procedure
- Even enlarging its computational time to 24 hours(!) does not help. Using this additional time, the procedure reduces the gap significantly but cannot outperform the solution quality of the best heuristic using only 60 seconds
- Due to complexity, it becomes interesting to limit the length of the time window
- To do so, complexity remains controllable
- Still, the time-oriented decomposition heuristic generates presumable good results in reasonable time

Conclusions

- Under specific propositions the use of the CLSPL model seems to be advantageous
- The heuristic approach seems to be very efficient but needs the use of an appropriate MIP solver and its complex model definition
- Some drawn conclusions against the use of the Tabu Search approach have to be reevaluated by additional tests under equal conditions
- Future work:
  - Parallel resources
  - Scheduling integration
  - Real-time restrictions

Literature for section 3

- Dixon, P. S. and Silver, E. A., “A heuristic solution procedure for the multi-item, single-level, limited capacity, lot-sizing problem,” Journal of Operations Management Vol. 1, 1981, pp. 23-38
- Federgruen, A.; Tzur, M.: A simple forward algorithm to solve general dynamic lot sizing models with n periods in O(n log(n)) or O(n) time. Management Science 37, pp.909-925, 1991.
- Haase, K.: Lotsizing and Scheduling for Production Planning. Springer, Berlin, 1994. (**ISBN-10:** 0-3875-7833-1)
- Kimms, A.: Multi-Level Lot Sizing and Scheduling. Physica, Heidelberg, 1997. (**ISBN-10:** 3-7908-0967-5)
- Stadtler, H.: Mixed integer programming model formulations for dynamic multi-item multi-level capacitated lotsizing. European Journal of Operational Research, Vol.94, pp.561-581, 1996.
- Stadtler, H.: Improved rolling schedules for the dynamic single-level lot-sizing problem. Management Science 46, pp.318-326, 2000.
- Stadtler, H.: Multilevel Lot sizing with setup times and multiple constrained resources: Internally rolling schedules with lot-sizing windows. Operations Research, Vol.51, No.3, pp.487-502, 2003.

Literature for section 3

- Suerie, C.; Stadtler, H.: The Capacitated Lot-Sizing Problem with Linked Lot Sizes. Management Science 49, No.8, pp.1039-1054, 2003.
- Tempelmeier, H.: Material-Logistik. 7th edition. Springer, (in German) 2008. (**ISBN-10:** 3-5407-0906-1)
- Wagner, H.M.; Whithin, T.M.: Dynamic version of the economic lot size model. Management Science 5, pp.145-156, 1958.