# Production planning and control with discrete lotsizing and a rolling horizon

**Wilhelm Dangelmaier,** Heinz Nixdorf Institut
**Bastian Degener,** Heinz Nixdorf Institut

Heinz Nixdorf Institut
Fürstenallee 11
D-33102 Paderborn

whd@hni.uni-paderborn.de

## The production system

We consider a production system for producing a certain set of products. The products are known in advance and can therefore be produced prior to a customer's order.

Orders are only given for a certain time horizon. By the course of time new customer orders are revealed. No possible plan can take all customer orders in consideration, because only a certain, restricted part of the future is known in advance, the planning horizon. Although a scheduling strategy cannot provide an optimal solution, due to a lack of complete knowledge about future customer orders, it should still avoid circumstances that will handicap future production processes. Hence, our basic rule: Independent of what the future will bring, the performance of the production, respectively the scheduling strategy is not allowed to fall below a certain bound. Of course, this bound should depend on the value of an optimal solution. Unfortunately this bound can only be determined as soon as all customer orders are known- and that is in retrospect. We are facing the challenge: How should a company produce today, even though only customer orders for the near future are known, or even worse, only the next customer order? Problems like this are handled within the framework of online optimizing [KrRa05]. Here, in contrast to classical offline optimization the input is revealed piece by piece and hence the optimal solution is unknown in advance, but still certain bounds are guaranteed to be kept independent of future events.

A schedule covers several periods with several time segments each (e.g. a horizon with 4 weeks and each week with 10 or 15 working shifts). It is updated periodically. At any one time the first period is considered to be the schedule implemented. The realization of a schedule is therefore the sustained implementation of

Dagstuhl Seminar Proceedings 09261
Models and Algorithms for Optimization in Logistics
http://drops.dagstuhl.de/opus/volltexte/2009/2186

respective first periods. Updates to the schedule only occur at proper points of time that are previously fixed for scheduling. Those are beyond the calendar (that is e.g. not during the weekend if a weekly planning cycle is considered). The scheduled demands (reduction in inventory) can be postponed or changed in value until their final fixation (latest by the time they are scheduled in the first scheduling period).

The considered production system is planned under the assumptions of the Discrete Lotsizing and Scheduling Problem; the production of a product covers always complete time segments (e. g. shifts, hours or discrete parts of an hour).

| | |
|---|---|
| $I^{PF}$ | Set of products respectively product indices, $i = \{1,...,n_{PF}\}$ |
| $T_P$ | Time model with the set of time segments respectively their indices $t = \{1,...,n^t\}$ and the set of points of time for the end of a time segment (planning horizon) $T = \{0,...,n^t\}$ ; $t = T$ |
| $b_{it}$ | Demand for product $i$ in time segment $t$ |
| $a_t$ | Available capacity in time segment $t$ |
| $b_i$ | Production coefficient for product $t$ |
| $k_i^{set}$ | Setup costs for product $t$ |
| $k_{it}^{qty}$ | Cost per unit for product $i$ in time segment $t$ |
| $k_i^{str}$ | Storage cost rate for product $i$ per time segment $t$ |
| $B_{i0}$ | Initial inventory for product $i$ |
| $B_{iT}^{sht}$ | Minimum inventory level for product $i$ at the end of time segment $t$ |
| $x_{it}$ | Lot size for product $i$ in time segment $t$ |
| $B_{iT}$ | Demand for product $i$ at the end of time segment $t$ |
| $\delta_{it}^{set}$ | Setup costs for product $t$ |
| $\delta_{it}^{pdn}$ | Production indicator for product $i$ in time segment $t$ |
| $w_i$ | Maximum amount of product $i$ in time segment $t$ |

The DLSP reduces the problem within a time segment to the decision whether to produce or not. Thus, the amount of production depends directly on the indicator variables and it holds $x_{it} = w_i \cdot \delta_{it}$. Since the DSLP in principle considers only time segments of unit supply of capacity, $w_i$ results as a time invariant ratio of the capacity per time segment $a_t$ and the production coefficient $b_i$. This leads to the following formulation of the DLSP:

Minimize $\sum_{i,t}(k_i^{set} \cdot \max\{0, \delta_{it}^{set} - \delta_{i,t-1}^{set}\} + k_{it}^{qty} \cdot w_i \cdot \delta_{it}^{set} + k_i^{str} \cdot B_{iT})$ under the restriction

$$\forall i \in I^{PF}, \forall t, T \in T_P: \quad B_{iT} = B_{i,T-1} + w_i \cdot \delta_{it}^{set} - b_{it} \qquad \text{(DLSP, 1)}$$

$$\forall t, T \in T_P: \quad \sum_i \delta_{it}^{set} \le 1 \qquad \text{(DLSP, 2)}$$

$$\forall i \in I^{PF}, \forall t, T \in T_P: \quad B_{iT} \ge B_{iT}^{sht} \qquad \text{(DLSP, 3)}$$

$$\forall i \in I^{PF}, \forall t, T \in T_P: \quad \delta_{it}^{set} \in \{0,1\} \qquad \text{(DLSP, 4)}$$

## Offline-/Online-Approach

When all future customer orders are known to an offline-algorithm, then

- it produces whatever is needed tomorrow/in the following period, today/in the active period
- it behaves demand-oriented (production deadlines)
- it does not require a fixed supply

and still[1] operates entirely from stock. The problem was mapped out as DLSP in the same manner. The cruel adversary orders exactly those products that are not in stock, independent from an online-algorithm's production. This can only be avoided by a trivial upper limit: No matter what is ordered, one can supply, because one has everything in stock - and the offline-algorithm can supply the customer order, due to its demand orientation at full supply availability. Therefore the offline-algorithm wins (almost) always.

– *Figure as metrical task-system / workfunction-approach*

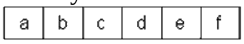
The workfunction-approach allows the following proceeding:

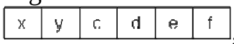
1. Customer orders of each single period and their placement times are viewed as "tasks".
2. The respective bin status reports, thus the amount of (different) product units that are in stock (for each product just on product unit), are viewed as "conditions". In the workfunction-mindset, the new bin status report has to be introduced at the beginning of a period. Therefore certain products are to be withdrawn and others to be produced. Once the new condition is established, it has to be decided what is still to be done about the customer orders, but cannot be covered by stock withdrawal. From the workfunction-algorithm's view, this is the actual "task". Since, from workfunction-algorithm's perspective the bin status report is not to be altered by the "task" in this period, the following is applicable: Alterations of the bin status report that are effected by the admis-

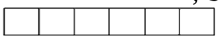
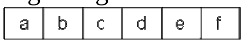
[1] with the leadtime 1 period with periodically repeated planning

sion of products in a period are not delivery time critical. They are carried out beforehand without time loss.

3. In reality, anything that is ordered during a task but is not in stock has to be produced during delivery time. Task-specific costs are the
   – production costs during delivery time (bin status report unaltered)
   – production costs for the withdrawn quantity and its restocking
4. When a product is in stock at the beginning of a period, but is expected not to be there at the end of the period, it has to be ordered during the task.

Exactly this circumstance leads to the workfunction-algorithms failure:


be the target state at the end of the period, which the workfunction-algorithm proposes. 

, be the actual condition at the beginning of the period,  be the capacity and 

be the customer order. Therefore there's no use for x and y. There is no possibility to reach the proposed condition from the actual condition.

It is not advisable to demand for the workfunction-algorithm to start with the actual condition at the beginning of a period or to end with for instance on a Friday with the condition that can still be reached. Otherwise any goal proposition would be lost and the workfunction-approach would be conducted ad absurdum. Concepts that scrap products without utilisation during ordering, only compromise availability. To use free capacities in order to approximate the reckoned workfunction-condition leads to maximum stock. Approximation to workfunction-stock should consider that this stock could already be altered in the next restocking period. Hectic rabbit and hedgehog politics are possibly just actionism. Therefore a reduced aberration (with exponential smoothing) could deliver a better standard operating procedure.

– *Figure as paging-problem*

The approach using the cache as stock and the main store as production line (see [KRKA05], p. 29), ensuring hundred per cent availability through the cache, leads to the same problem: if an enquiry cannot be met by the cache, the element will be withdrawn from the main store and placed (analogy: produced) into the cache. In exchange, another element will be overwritten (scrapped?).

– *Figure as linear list*

On this list (see [KRKA05], all products should (quasi) be listed as stock. Missing products could potentially be marked with high costs (within delivery time). Here, access costs are dependent on the sorting of the sought after element. For this reason, the offline-problem differs remarkably from the production system at hand.

## Process model for on-line scheduling

For instance, Figure 1 displays a weekly continued planning, where always one week is fixed respectively. This week is divided for instance in days or shifts (4 week horizon, the schedule is repeated weekly, each week consists of 10 shifts).

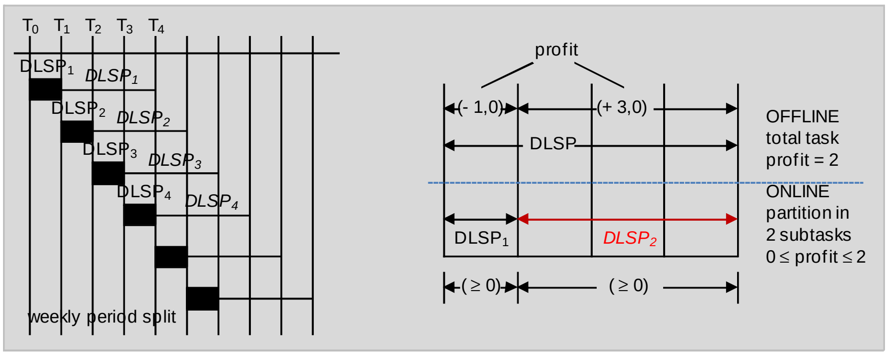



Figure 1: Partition in two subtasks; an example where the DLSP solutions differ

Note that in the example, the profit for the first week of the complete DLSP is negative. Since there is rescheduling at the beginning of each week, it is possible that this value never changes to a positive value, although an off-line algorithm could make a profit over the entire period. A naive approach would therefore fail. We introduce a first possible solution.

Independent of the formal description of the production the following approach is chosen:

- Optimization of the 1. period + optimization of the remaining periods of the planning horizon, considering the joint inventory at the end of the first period.[1]
- Weighting the two (partial) solutions
- Dependent on the chosen weighting, a cost minimal solution for the corresponding planning horizon is chosen.

In Figure 1 at point of time $T_0$ the schedules $DLSP_1$ and ***DLSP**$_1$* are calculated. The schedule that maximizes (profit) respectively minimizes (costs) with $\alpha \cdot DLSP_1 + (1-\alpha) \cdot DLSP_1$ with $0 \leq \alpha \leq 1$ is chosen. This way

- at the end of the first period disprofit is avoided, respectively a (minimum) profit is realized[2]
- a feasible schedule for the planning horizon is created
- due to the DLSP, a mathematical formulation of the problem is enabled

---

[1] Alternatively, one can construct an inventory at the end of the horizon that allows to carry forward the horizon without conflicts, if this is feasible in the given setting.

[2] We assume that an optimum plan is always connected with a profit.

Of course, the solution for the DLSP over the (total-) horizon and the solutions to the two DSLP partial problems that are coupled by the inventory, need not to be identically, because each partial solution has to guarantee a positive value (refer to the example in figure 1).

## Online Scheduling based on a DLSP model

The following notations are used in the approach:

$T_P$ is the total horizon, $T_T^{'}$ and $T_T^{''}$ are the partial horizons at point of time $T$. $k^{sales}$ is the sales revenue per production unit. A revenue can only be obtained in time segments with asset sale (demand) $(b_{it} > 0)$.

$B_{i0}$ initial inventory

$\delta_{i0}^{set}$ initial setup

$T_P$ total horizon

Thus:

Maximize

$$\sum_{T_T^{'} \in T_P} \left( \sum_{t,T \in T_T^{'}} (b_{it} \cdot k_{it}^{sales} - (k_i^{set} \cdot \max\{0, \delta_{it}^{set} - \delta_{i,t-1}^{set}\} + k_{it}^{qty} \cdot w_i \cdot \delta_{it}^{set} + k_i^{str} \cdot B_{iT} \right))$$

under the following restrictions:

– Restrictions for partial problem 1 (partial horizon $T_T^{'}$):

$$\forall i \in I^{PF}, \forall t, T \in T_T^{'}: \quad B_{iT} = B_{i,T-1} + w_i \cdot \delta_{it}^{set} - b_{it} \quad \text{(DLSP, 1)}$$

$$\forall t, T \in T_T^{'}: \quad \sum_i \delta_{it}^{set} \le 1 \quad \text{(DLSP, 2)}$$

$$\forall i \in I^{PF}, \forall t, T \in T_T^{'}: \quad B_{iT} \ge B_{iT}^{sht} \quad \text{(DLSP, 3)}$$

$$\forall i \in I^{PF}, \forall t, T \in T_T^{'}: \quad \delta_{it}^{set} \in \{0,1\} \quad \text{(DLSP, 4)}$$

The initial inventory at the beginning of period $T_T^{''}$ is the inventory at the end of period $T_T^{'}$ of the previous planning cycle.

– Restrictions for partial problem 2 (partial horizon $T_T^{''}$):

$$\forall i \in I^{PF}, \forall t, T \in T_T^{''}: \quad B_{iT} = B_{i,T-1} + w_i \cdot \delta_{it}^{set} - b_{it} \quad \text{(DLSP, 1)}$$

$$\forall t, T \in T_T^{''}: \quad \sum_i \delta_{it}^{set} \le 1 \quad \text{(DLSP, 2)}$$

$$\forall i \in I^{PF}, \forall t, T \in T_T^{''}: \quad B_{iT} \ge B_{iT}^{sht} \quad \text{(DLSP, 3)}$$

$$\forall i \in I^{PF}, \forall t, T \in T_T^{''}: \quad \delta_{it}^{set} \in \{0,1\} \quad \text{(DLSP, 4)}$$

– Joint constraints for both partial horizons $T_T^{'}$ and $T_T^{''}$:

$B_{iT}$ at the end of time horizon $T_T^{'} = B_{iT}$ at the beginning of time horizon $T_T^{''}$

With

$$K_T^{sales'} = \sum_{i,(t,T\in T_T^{'})} (b_{it} \cdot k_{it}^{sales} - (k_i^{set} \cdot \max\{0, \delta_{it}^{set} - \delta_{i,t-1}^{set}\} + k_{it}^{qty} \cdot w_i \cdot \delta_{it}^{set} + k_i^{str} \cdot B_{iT})) \geq X \geq 0$$

and

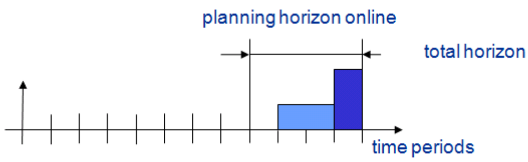
$$K_T^{sales'} = \sum_{i,(t,T\in T_T^{''})} (b_{it} \cdot k_{it}^{sales} - (k_i^{set} \cdot \max\{0, \delta_{it}^{set} - \delta_{i,t-1}^{set}\} + k_{it}^{qty} \cdot w_i \cdot \delta_{it}^{set} + k_i^{str} \cdot B_{iT})) \geq Y$$

as well as $0 \leq \alpha \leq 1$ then yields $(\alpha \cdot K_T^{sales'} + (1-\alpha) \cdot K_T^{sales'}) = \max \sum_T$ .

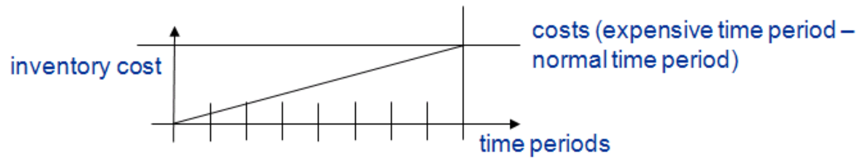
## Competitiveness of the approach

The competitiveness is measured concerning the ratio of the total horizon of the off-line as well as the total horizon of on-line approach. A schedule, that is the solution to the DLSP-problem, can shift orders only towards presence, due to availability constraints. The off-line solution is capable of using the entire horizon, while the on-line solution can only use the planning horizon.

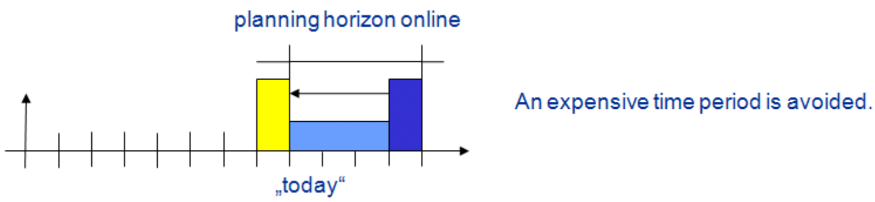
The following considerations are to be reckoned: If all periods are equally expensive, the solution is as follows (disregarding revertive costs): “As late as possible.” Here, certain periods, Saturdays or third shifts respectively, are to be more expensive than regular shifts (for instance first or second shifts). No sense is seen in a differentiation such as “Mondays are more expensive than Tuesdays”. There results no distinction from the offline-algorithm, as long as adjusting an expensive period into a more convenient period within the online-algorithm’s planning horizon is possible.




Differences occur, as soon as the offline-algorithm can use a more convenient period outside the planning horizon. The offline-algorithm is only going to do this if the adjustment costs, thus (further) capital lockup costs, for the then heightened stock do not exceed the savings from the then avoided expensive period.




True for this example is: An adjustment of 9 periods is more expensive than the savings from avoiding an expensive period. Therefore the offline-algorithm will not adjust for 9 periods (or more) (“adjustment limit” = 8 periods). Compared to an online-algorithm, the offline-algorithm wins most if it avoids the expensive period by minimal costs (while the online-algorithm is not capable to do the same!). This is shown in the following delineation:




Therewith, an expensive period is avoided. Concerning the carrying costs, the following is true: An adjustment by 8 periods costs as much as the margin between regular and expensive period. Therewith, the capital lockup costs and the costs for regular/expensive periods are comparable. Then an adjustment by 4 periods costs half: 0,5 * (margin more expensive - regular period). Therewith the offline-algorithm’s savings, which the online-algorithm cannot realise, is: (1-0,5) * (margin more expensive - regular period).

*Example*:

Only the one product case with consistent storage costs is regarded.
“Saturday” with duplicate production costs is considered as period with increased costs.

- Online: The new period’s “Saturday”-demand cannot be adjusted to a more convenient day any longer
- Offline: In the overall horizon, the “Saturday”-demand can be adjusted to a more convenient (“week”-) day (before the planning horizon)

Using this time slot can be iterated by the off-line algorithm, until the upper bound is reached that is set by the inventory cost. The number of those iterations can be set to (total horizon: (bound of offset + (bound of offset - planning horizon))).

Considering the formal description of the heuristic, it follows: A sales revenue is only possible if there are customer and those orders can be satisfied with costs that do not exceed the sales revenues. Thus, the heuristic tends to a just-in-time policy, but with a limited utilization of the capacity! If there are no customer orders present in a period, nothing can be shifted into this period. Otherwise the demand can always be satisfied by the maximal capacity (otherwise a shifting would be required; this might be for instance the 24-hour capacity of the machinery.)

- *Production costs*
- We consider the worst case for the on-line algorithm by assuming a sequence of periods as follows: 1 period with maximal demand followed by a period with minimal demand. The minimal demand is (average demand - (maximal demand - average demand)). This way, the off-line algorithm can level the needed capacity with minimal inventory: 1 period build up (maximal demand- average demand), 1 period reduction (maximal demand - average demand). If the off-line algorithm can balance over the entire planning horizon, for the on-line algorithm holds: 1 period with minimal load without shifting is followed by some periods with average load, is followed by 1 expensive load with maximal load. For the first case we have 1 period with minimal load + minimal balancing followed by 1 period with maximal load and minimal down balancing. Therefore the costs for the expensive capacity are maximal. In both cases the on-line algorithm is forced to choose the worse one.
-

– *Setup costs*

We consider the worst case for the on-line algorithm:

- The optimal lot size leads to an ordering cycle that is larger than the planning horizon. Thus, the off-line algorithm minimizes the sum of the setup and inventory costs.
- If balancing is impossible (revenue = gross demand of the period) then there has to be disposed a product in each period. The inventory is minimal, the setup costs are maximal.
- If balancing is possible, but restricted by the minimal profit, this still holds (as an upper bound)
- Then we have to assume: number of products ≥ number of time segments / period.

– *First case: shifting < planning horizon*

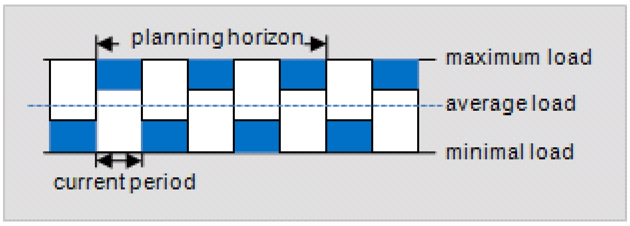



Figure 2: Load model considering competitiveness

The best case for the off-line inventory, concerning a complete balancing of capacity from one period into another one, is the case where there are z (e.g. 5) empty and cheap time segments at the end of a period followed by period with z expensive time segments that are covered. Here an off-line algorithm can completely balance its capacity, but at the price of additional inventory costs.

$$costs_{off} = \overbrace{1/2}^{a} \cdot \overbrace{(\text{amount of product in respective to an expensive time slot within a period})}^{b} \cdot \overbrace{(\text{inventory costs per period})}^{c}$$

For the on-line algorithm:

$$costs_{onl} = (1 - \text{factor for the minimal profit}) \cdot a \cdot b \cdot c + (\text{factor for the minimal profit}) \cdot b \cdot \text{additional costs for expensive periods}$$

In order to shift all expensive time segments to a previous period, the factor of the minimal profit has to be set to "zero". Then the on-line algorithm aligns with the off-line algorithm (for this case). If the factor of the minimal profit is set to "1" no shifting to the previous period is possible and the whole costs for the expensive time segments have to be applied.

Example: Factor of minimal profit = {0.1, 0.4, 0.8}
10 normal time segments and 5 expensive time segments in one period
Shifting to the normal time segments in previous period results in inventory costs for 5 segments
Inventory costs per sement: = 1/10 (costs for producing in an expensive segment)
(Costs for producing in expensive time segments): = 2(costs for producing in normal time segments)
Counting costs in units of production costs in normal segments.

$costs_{off}$ = 10 · normal + 1/2 · 1/10 · 5 · 5 = 11,25
$costs_{onl}$ = 10 · normal +
(1/2 · 1/10 · 5 · 5) · 0,9 + 0,1 · 1 · 5 = 11,625 (+ 0,5)
(1/2 · 1/10 · 5 · 5) · 0,6 + 0,4 · 1 · 5 = 12,75 (+ 2)
(1/2 · 1/10 · 5 · 5) · 0,2 + 0,8 · 1 · 5 = 14,25 (+ 4)

The capacity of the shifting is considered to be given - otherwise the off-line algorithm could not use them either. On the other hand, the minimal profit cannot be fixed somehow. Here the minimal profit is in close relation to the expensive capacity. In fact it depends on the ratio of revenue and costs in a period by a certain part. In the worst case (consider an adversary) this can be "zero". The costs for the on-line algorithm are accordingly:

$$costs_{onl} = 10 \cdot normal + 5 \cdot expensive = 10 \cdot normal + 5 \cdot 2 \cdot normal = 10 + 10 = 20$$

(if the costs of the expensive time segment are set to twice as expensive)

The ratio on-line/off-line is therefore 20/11.25, which can hardly be considered as optimization in the classical sense. Thus, the minimal profit should be defined in a smarter way. An obvious approach would be accumulating the profits over the total horizon, starting with 0 at the beginning (e.g. the value of the last stocktaking) till the current period. It would be reasonable to take the current period into account for the cumulative. However, the following example (figure 3) illustrates that there might be no balancing at all.

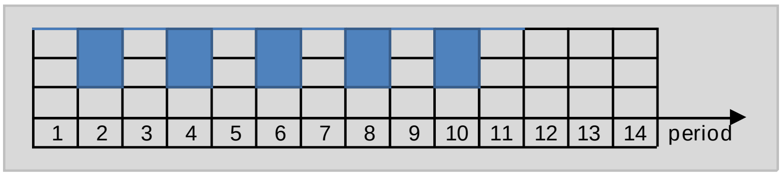
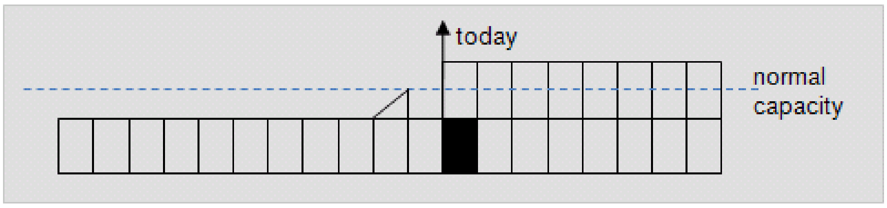


Figure 3: Example for a load model

The only requirement is an appropriate loss in the first (max-) period. Already in the 4th period one might expect that balancing might be useful. Therefore, we chose the following approach: We use the "pull ahead" of the respective off-line solution in the current period, unless there is cumulated profit larger than zero. In this case only a certain fraction is used. That means: once there is a cumulated profit, it is not completely spend anymore. If there is no profit so far, we balance and try to keep the costs as low as possible:

- We shift to the front, if the off-line algorithm was unable to make any profit in the past
- We shift to the front, if the off-line algorithm made a “large” profit in the past

For that purpose an off-line solution for the entire past is calculated. The current period is integrated in this off-line solution and is accounted the average value of shifting forward in the past (average use of capacity in periods that where balanced). This is critical, if there is a minimal profit in the past. Then the off-line algorithm is able to balance, while the on-line algorithm has to use the expensive time slots.




But the off-line algorithm will not further balance when the inventory costs exceed the savings. Then we have for the shifting forward of x periods:

Savings = $x \cdot$ expensive period - $\sum_{s=1}^{x} (2(s-1)+1)$ $\cdot$ inventory costs per period

= $x \cdot$ expensive period - $x^2$ $\cdot$ inventory costs per period

From the first derivative to $x$

$d/dx$ (savings) = (expensive period) - (2 $\cdot$ inventory costs per period)

follows: The savings are maximal for

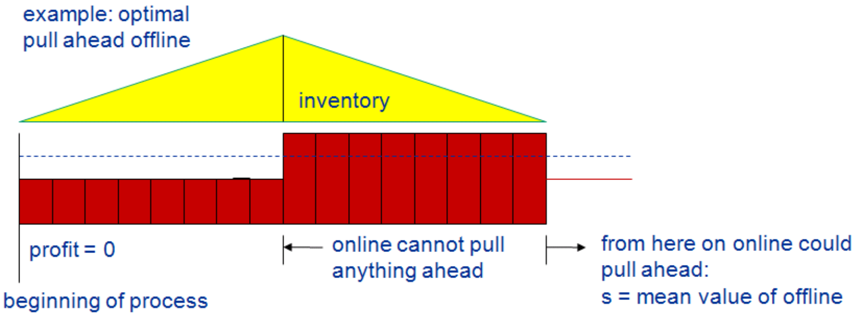
$x^*$ = (expensive period) / (2 $\cdot$ inventory costs per period)

$x^*$ is the best possible horizon for shifting to the front with maximal savings „savings*“.
On the other hand, the on-line algorithm cannot avoid the expensive periods:

costs $_{onl}$ = x · normal + x · expensive (considering normal time slots as given)

costs $_{off}$ = costs$_{onl}$ - savings*

This leads to the ratio (costs$_{onl}$ / costs$_{off}$).




In general holds:

- Best possible shifting $x^* = (k_p^t - k_p^n)/(2 \cdot k_B)$
- Maximal savings* = $x^* \cdot (k_p^t - k_p^n) - (x^*)^2 \cdot k_B$
- Ratio on-line/off-line

$$\frac{k_{onl}}{k_{off}} = \frac{x^* \cdot (k_p^n) + x^* \cdot (k_p^t)}{x^* \cdot k_p^n + x^* \cdot k_p^t - x^*(k_p^t - k_p^n) + (x^*)^2 \cdot k_B} = \frac{x^*(k_p^t + k_p^n)}{2 \cdot k_p^n x^* + (x^*)^2 \cdot k_B}$$

- Example
  (cost for producing in an expensive period) : (inventory costs per period) = $10:1$
  normal period = $1/2$ expensive periods
  Savings: x · expensive period - $\sum_{s=1}^{x} (2(s-1)+1)$ · $1/10$ expensive period

$$k_p^t = 1; k_p^n = 0{,}5; k_B = 0{,}1; x^* = 2{,}5$$

$$\text{Savings}^* = 1{,}25 - 0{,}625 = 0{,}625$$

$$\frac{k_{onl}}{k_{off}} = \frac{2{,}5 \cdot 1{,}5}{2 \cdot 0{,}5 \cdot 2{,}5 + 6{,}25 \cdot 0{,}1} = \frac{3{,}75}{3{,}125} = 1{,}2$$

| | inventory costs | savings production costs | savings Σ | $x^* \frac{\Delta \text{expensive period}}{2 * \text{inventory costs per period}} = 2{,}5$ |
|---|---|---|---|---|
| x = 1 | 1/10 | 1 • 0,5 | 0,4 | |
| x = 2 | 3/10 | 2 • 0,5 | 0,6 | |
| x = 3 | 5/10 | 3 • 0,5 | 0,6 | |

| | | | | |
|---|---|---|---|---|
| x = 4 | 7/10 | 4 • 0,5 | 0,4 | $savings = 2{,}5 \cdot 0{,}5 - 6{,}25 \cdot 0{,}1 = 0{,}625$ |
| x = 5 | 9/10 | 5 • 0,5 | 0 | $costs_{onl} = 2{,}5\ normal + 2{,}5\ expensive = 3{,}75$ |
| x = 6 | 11/10 | 6 • 0,5 | - 0,6 | $costs_{off} = 3{,}75 - 0{,}625 = 3{,}125$ |
| x = 7 | 13/10 | 7 • 0,5 | - 1,4 | $costs_{onl} / costs_{off} = 3{,}75 / 3{,}125 = 1{,}2$ |
| x = 8 | 15/10 | 8 • 0,5 | - 2,4 | |
| x = 9 | 17/10 | 9 • 0,5 | | |

– *Second case: shifting $\geq$ planning horizon*

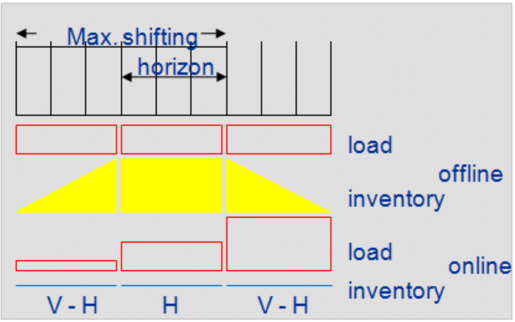



Figure 4: Shifting planning horizon

We consider the time segment $2(V - H) + H$. In the second case holds for the shifting forward by $x$ periods:

Savings = x $\cdot$ Δ expensive period - $\sum_{s=1}^{x}(H + 2(s-1) + 1)$ $\cdot$ inventory cost per period

= x $\cdot$ Δ expensive period - $(x * H + x^2)$ $\cdot$ inventory cost per period

From the first derivative to x

$d/dx$ (savings) = Δ expensive period - $(H + 2x)$ $\cdot$ inventory cost per period

follows: The savings are maximal for

$x*$ = (Δ expensive period – H $\cdot$ inventory costs per period) /
(2 $\cdot$ inventory costs per period)

$x*$ is the best possible horizon for shifting to the front, thus the shifting horizon to use is $V = H + x$.

In general:

- Best possible shifting $x^* = \dfrac{(k_p^t - k_p^n) - d_H \cdot k_B}{2 \cdot k_B}$
- Maximal savings* = $x * \cdot (k_p^t - k_p^n) - (x * \cdot d_H + (x)^2) \cdot k_B$

- Ratio on-line/off-line

$$\frac{k_{onl}}{k_{off}} = \frac{(d_h + x^*)\cdot k_p^n + x^* \cdot k_p^t}{(d_h + x^*)\cdot k_p^n + x^* \cdot k_p^t - x^*(k_p^t - k_p^n) + (x^* \cdot d_H + (x^*)^2 \cdot k_B}$$

$$= \frac{d_H \cdot k_p^n + x^*(k_p^t + k_p^n)}{d_H \cdot k_p^n + x^*(2 \cdot k_p^n + d_H \cdot k_B + x^* \cdot k_B)}$$

- Example

$$k_p^t = 1; k_p^n = 0{,}5; k_B = 0{,}1; d_H = 4; x^* = 0{,}5$$

$$\text{Savings}^* = 0{,}25 - (0{,}5 \cdot 4 + 0{,}25)\cdot 0{,}1 = 0{,}25 - 0{,}225 = 0{,}025$$

$$\frac{k_{onl}}{k_{off}} = \frac{4 \cdot 0{,}5 + 0{,}5(1{,}5)}{4 \cdot 0{,}5 + 0{,}5(1 + 0{,}4 + 0{,}05)} = \frac{2{,}75}{2{,}725} = 1{,}01$$

- 

<table>
<tr><th></th><th>inventory costs</th><th>savings production costs</th><th>savings Σ</th><td rowspan="8">$x^* \frac{0{,}5 - 0{,}4}{2 \cdot 0{,}1} = 0{,}5$<br>$\text{savings} = 0{,}25 - 0{,}225 = 0{,}025$<br>$\text{costs}_{onl} = 4{,}5 \cdot 0{,}5 + 0{,}5 \cdot 1 = 2{,}75$<br>$\text{costs}_{off} = 4{,}5 \cdot 0{,}5 + 0{,}5 \cdot 1 - 0{,}025 = 2{,}725$<br>$\frac{\text{costs}_{onl}}{\text{costs}_{off}} = \frac{2{,}75}{2{,}725} = 1{,}01$</td></tr>
<tr><td>x = 1</td><td>5/10</td><td>1 • 0,5</td><td>0</td></tr>
<tr><td>x = 2</td><td>+ 7/10</td><td>2 • 0,5</td><td>- 0,2</td></tr>
<tr><td>x = 3</td><td>+ 9/10</td><td>3 • 0,5</td><td>- 0,6</td></tr>
<tr><td>x = 4</td><td>+ 11/10</td><td>4 • 0,5</td><td>- 1,2</td></tr>
<tr><td>x = 5</td><td>+ 13/10</td><td>5 • 0,5</td><td>- 2,0</td></tr>
<tr><td>x = 6</td><td>+ 15/10</td><td>6 • 0,5</td><td>- 3,0</td></tr>
<tr><td>x = 7</td><td>+ 17/10</td><td>7 • 0,5</td><td>- 4,2</td></tr>
</table>

## Conclusion

The proposed approach offers for a special form of production (DLSP) an on-line optimization in the sense of a comparison to an optimal off-line solution. The approach is demonstrated on examples with production and inventory costs. An extension to other forms of production as well as other forms of costs (e.g.) setup costs should be possible.

Furthermore, the parameters how the costs for different time slots depend on each other as well as on the inventory costs are fixed in our example to give a first insight how the model works. Future work will consider these parameters and dependency of the competitiveness. Note that the values chosen in this paper are reasonable for realistic scenarios.

# References

DANGELMAIER, W.: Online-Optimierungsansätze zur Steuerung der Produktion in der Serienfertigung. In: Schenk, M. (Hrsg.): Logistik - Effiziente und sichere Warenketten in Industrie und Handel. 11. IFF-Wissenschaftstage 25./26. Juni 2008. Tagungsband S. 145-154. Magdeburg: Fraunhofer-Institut für Fabrikbetrieb und -automatisierung 2008.

DANGELMAIER, W.: PPS-Strategien und Online-Optimierung in der Serienfertigung bei kurzfristigem Horizont. In: Müller, W.; Spanner-Ulmer, B. (Hrsg.): Wandlungsfähige Produktionssysteme. S. 15-24. TBI'08. 13. Tage des Betriebs- und Systemingenieurs. II. Symposium Wissenschaft und Praxis. Wissenschaftliche Schriftenreihe des Instituts für Betriebswissenschaften und Fabriksysteme. Chemnitz, 13. November 2008.

DANNE, CH.; BLECKEN, A.; DANGELMAIER, W.: Complexity-Induced uncertainty in Supply chains - A Framework and Case Studies. In: Pfohl, H.-Ch.; Wimmer, Th. (Hrsg.): Wissenschaft und Praxis im Dialog. Robuste und sichere Logistiksysteme. S. 71-88. 4. Wissenschaftssymposium Logistik. Hamburg: Deutscher Verkehrsverlag 2008.

KRUMKE, S. O.; RAMBAU, J.: Online Optimierung. Vorlesungsskript. Berlin: Technische Universität 2005.