MAGISTERARBEIT

Titel der Magisterarbeit

"Lotsizing and Scheduling Problems
- A Review"

Verfasserin

Sally Tischer, Bakk.

Angestrebter akademischer Grad

Magistra der Sozial- und

Wirtschaftswissenschaften

(Mag. rer. soc. oec.)

Wien, im Oktober 2007

Studienkennzahl lt. Studienblatt:	A 066 915
Studienrichtung lt. Studienblatt:	Betriebswirtschaftslehre
Betreuer:	O.Univ.Prof.Dipl.-Ing.Dr. Richard F. Hartl

I. TABLE OF CONTENTS

II. List of figures

III. List of tables

1 Introduction

Today every customer wants to have low-priced products with good quality. This increases the cost pressure of the producers, and because of that cost-saving production is very important for them to survive among the competitors. So producers try to improve their production processes. It is not possible to optimize the whole process for reasons of complexity, only parts can be looked at.

This diploma thesis looks at the lotsizing and scheduling problems. Lotsizing and scheduling are important parts of the production process system. It is necessary to know how many items and in which order they have to be produced, to minimize the total costs.

In the context of this work an overview of the lotsizing and scheduling problems and the solving methods are given. The problems are described through definitions, mathematical models and examples.

In chapter 1.1 an outline is given of the production planning and control system (PPS). Then the definitions and the relationship of lotsizing and scheduling are specified in chapter 2. In chapter 3 a short overview of the lotsizing and scheduling problems is noted. The different problems follow then in chapter 4 (single-level uncapacitated lotsizing and scheduling problems), chapter 5 (single-level capacitated lotsizing and scheduling problems), chapter 6 (multi-level lotsizing and scheduling problems) and chapter 7 (lotsizing and scheduling problems with multiple machines). In chapter 8 a hierarchical integration is represented. Also a comparison of different models is illustrated in chapter 9. At last the solving methods are mentioned in chapter 10.

1.1 Production Planning and Control System[1]

The production planning and control system, short PPS, is divided into two components (see figure 1), which are production planning and production control. The production planning plans procedures in advance (short and medium term planning), and the production control is responsible for the order clearance and order control on the basis of the production planning.

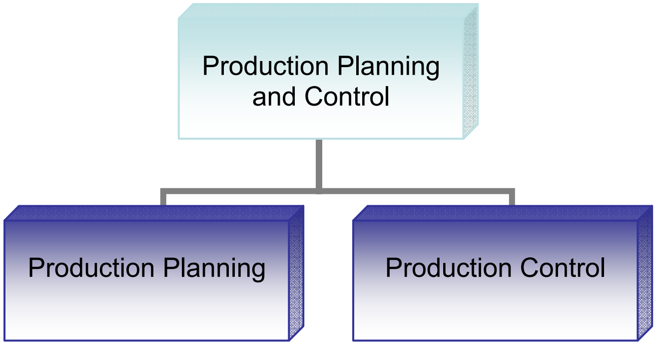


Figure 1: The PPS structure

1.1.1 Production Planning

The production planning consists of three parts (see figure 2), the production program planning, the provision planning and the production process planning.

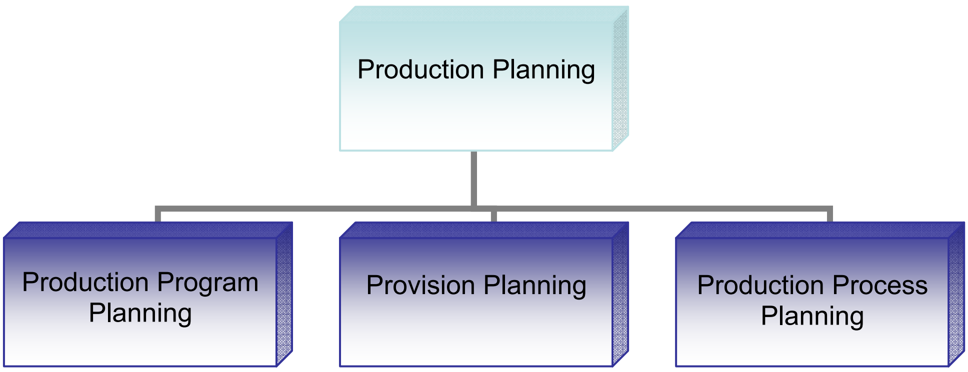


Figure 2: The production planning structure

These parts can be divided into strategic, tactical and operational production planning.

The strategic part contains of the long term decisions from the production program planning and provision planning.

[1] Domschke, Scholl, Voß, 1993, p. 8

There are decisions to make about:

- what to produce (assortment)
- where to produce (location)
- how to produce (method)
- with what to produce (factors of production)

The tactical production planning is the detailed determination of the production program concerning amount of different product variants, and medium term adjustment of capacity for example machines and personal.

The operational part consists of the short term program planning, the supply of materials and the production process planning.

1.1.1.1 Production Program Planning

The production program planning determines the type, the amount and the production date of the primary requirements. It tells which products in what amount are going to be produced in a certain time.

It can be divided on the basis of maturity into potential and actual production program. The potential one consists of long term decisions for what products should be produced by looking at the business objective. The actual production program defines by looking at the potential production program and factors of production the in reality to produced products (type, amount and timeframe). Here also the business objectives are to be considered.

1.1.1.2 Provision Planning

The provision planning acquires the amount of secondary and tertiary requirements for the primary requirement. These production factors are looked at by quality, quantity and date of supply.

The production factors can be for example:

- resources (machines, facilities)
- personal
- materials

1.1.1.3 Production Process Planning

The production process planning looks at the inputs of actual production program and the provision planning, what makes it possible to plan the precise production orders.

Important points to look at are:

- Which orders can be processed together?
- What materials are used?
- In which sequence can the orders be planed?
- Time regulation of the production of the orders?

The production process planning is divided into three parts (see figure 3):

1. Lotsizing (see section 2.1)
2. Lead time scheduling and capacity scheduling:

   The lead time scheduling plans the earliest and latest time for processing the orders.

   The capacity scheduling looks if the needed capacities for the production program are available.
3. Scheduling (see section 2.2)

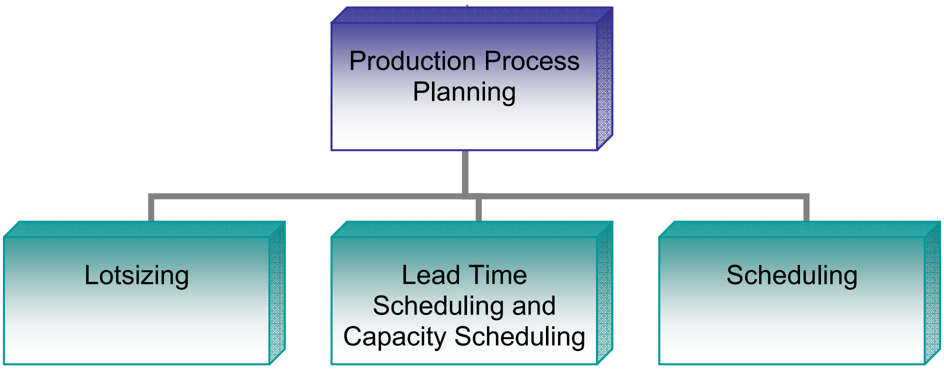


Figure 3: The production process structure

1.1.2 Production Control

The production control consists of two components (see figure 4), the order release and the order monitoring.

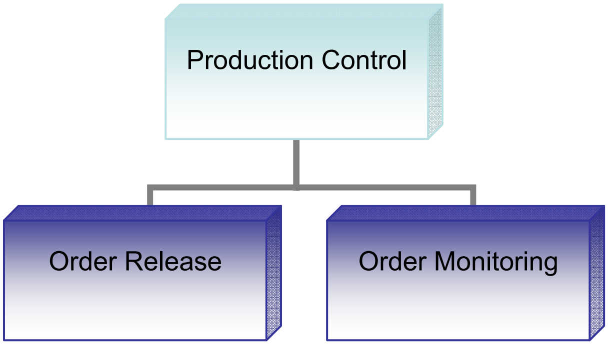


Figure 4: The production control structure

1.1.2.1 Order Release

The function of the order release is to start the production of the orders, which are incoming from the production planning stage.

1.1.2.2 Order Monitoring

The order monitoring checks the production flow, by looking at the response from the actual data of production. The actual data is compared with the target data. When there are any variations between the actual and the target data, then for example the production amount or production deadline has to be examined.

In this work we are looking at the production process planning, in special at lotsizing and scheduling.

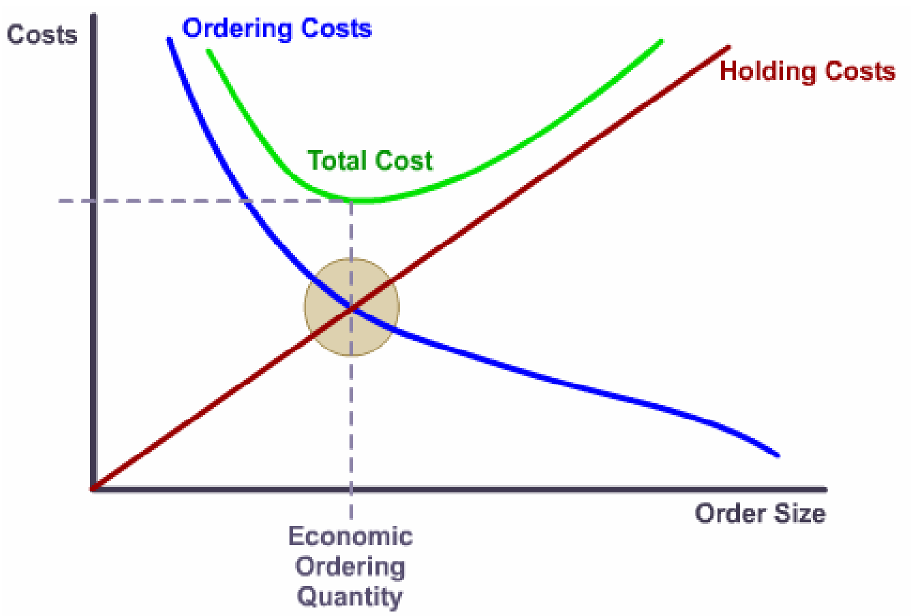
2 Lotsizing and Scheduling

This chapter gives the definitions of lotsizing and scheduling and points out the relationship between each other.

2.1 Lotsizing

Lotsizing is the activity to transform customer demands into lots to minimize the total costs of setup costs, product change costs and inventory costs. A lot is an amount of equal items produced on a machine one after the other without changing the setup state. The amount of items in a lot is called lotsize[2].

The relationship between lotsize and costs[3]:
The number of setups of an item depends on the size of the lot. The greater the lot is the smaller is the number of setups and the setup costs per item. On the other side the inventory costs are increased. So the setup and inventory costs have to be balanced to minimize the total costs (see figure 5).



Note: The minimal total costs are found in the intercept point of the ordering and the holding costs. This is only valid for the EOQ model which has linear holding costs and fixed ordering costs.

Figure 5: The relationship between lotsize and costs[4]

[2] Haase, 1994, p. 1

[3] Haase, 1994, p. 2

[4] http://en.wikipedia.org/wiki/EOQ, 06.08.2006

2.2 Scheduling

Scheduling is the activity to decide when and on which machine different items should be produced so that there are no shortages in the production plan. It gives the start and finish time of every job and the job sequence of each machine[5].

Scheduling problems are specified through three attributes[6].
These are:

- the machine environment (α),
- the job characteristics (β)
- and the optimality criterion (γ).

2.2.1 The machine environment (α)

The machine environment has two global parameters. One tells on which machine each job has to be processed, and the other gives the amount of machines.

The types of machine environment[7] are:

- Single machine
  In this environment there is only one machine available that can process the jobs. Every job is done on the same machine.

- Parallel machine
  Here there are multiple machines, which work parallel, available to process the jobs. Every job can be done on every machine.

[5] Quadt, 2004, p. 2

[6] Brucker, 1995 ,p. 2

[7] http://riot.ieor.berkeley.edu/~vinhun/machine.html, 20.05.2007

- Flow Shop
  There are a series of machines. The first task of every job is done on the first machine, the second task on the second machine and so on. Every job goes through all machines in a one-way order.

- Job Shop
  Here there are a set of machines. Every job can go through the machines in a different, but given order. The flow of the tasks of a job does not have to be one-way, so each job can use a machine more than once.

- Open Shop
  Here there is a set of machines, too. The flow of the tasks is given, but when and on which machine the task has to be done is not given.

2.2.2 The job characteristics (β)

There are six job characteristics ($\beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6$), which build the set β[8] :

1. β_1 is the job characteristic that tells if job splitting is allowed. That means that a job can be interrupted and then goes on at a later time or even on another machine. When job splitting is allowed then $\beta_1 = pmtn$, and when not then β_1 does not appear in β.
2. β_2 describes the precedence relations between jobs. These relations can be represented through different precedence graphs. If $\beta_2 = prec$ then the relation is shown by an acyclic directed graph. When $\beta_2 = tree$ then the graph is a tree. For $\beta_2 = sp - graph$ the relation is given by a series-parallel graph. When there is no precedence then $\beta_2 = 0$ and it does not appear in β.
3. β_3 tells if there are release dates for each job. When $\beta_3 = r_j$ release dates are specified, but when $r_j = 0$ the job characteristic $\beta_3 = 0$ and so it does not appear in β.

[8] Brucker, 1995 ,p. 3

4. β_4 gives the restrictions on the processing times or on the number of operations. If β_4 is $p_i = 1$ then every job has processing time, but when $\beta_4 = 0$ it does not appear in β.
5. β_5 tells if there is a deadline for a job. When $\beta_5 = d_i$ then a deadline is specified for a job but when $\beta_5 = 0$ then it is not in the set β.
6. β_6 indicates that there is a batching problem, what means that a number of jobs have to be summarized to a batch and then scheduled together. In the batch all jobs have to be finished at the same time as the finish time of the last batch job. The number of jobs can reach from one to n jobs. When $\beta_6 = batch$ then a batching problem occurs, otherwise if $\beta_6 = 0$ it does not appear in β.

2.2.3 The optimality criterion (γ)[9]

The optimality criterion is to find a feasible schedule of the jobs that minimizes the total costs.
There are two types of total cost functions, the bottleneck objectives and the sum objectives. Bottleneck objective functions are for example makespan and maximum lateness. A sum objective is for example the total flow time.
The makespan is the completion time of the last job. The total flow time is the total sum of the completion times of all jobs. Lateness is the difference between the completion time and the due date of a job. When the completion time exceeds the due date unit penalty costs arise.

[9] Brucker, 1995 ,p. 6

2.3 Relationship between lotsizing and scheduling

Lotsizing and scheduling decisions depend on each other. Lotsizing is concerned with the product quantities and scheduling with the machines and product sequences. Lotsizing looks at the number of items of a product which can be produced per period. This depends on the setup times, which again depend on the assignment of the machine and the product sequence which are fixed by scheduling. Reverse scheduling needs the production volume to decide about the machine assignment and the sequence[10].

The relationship is shown in figure 6.

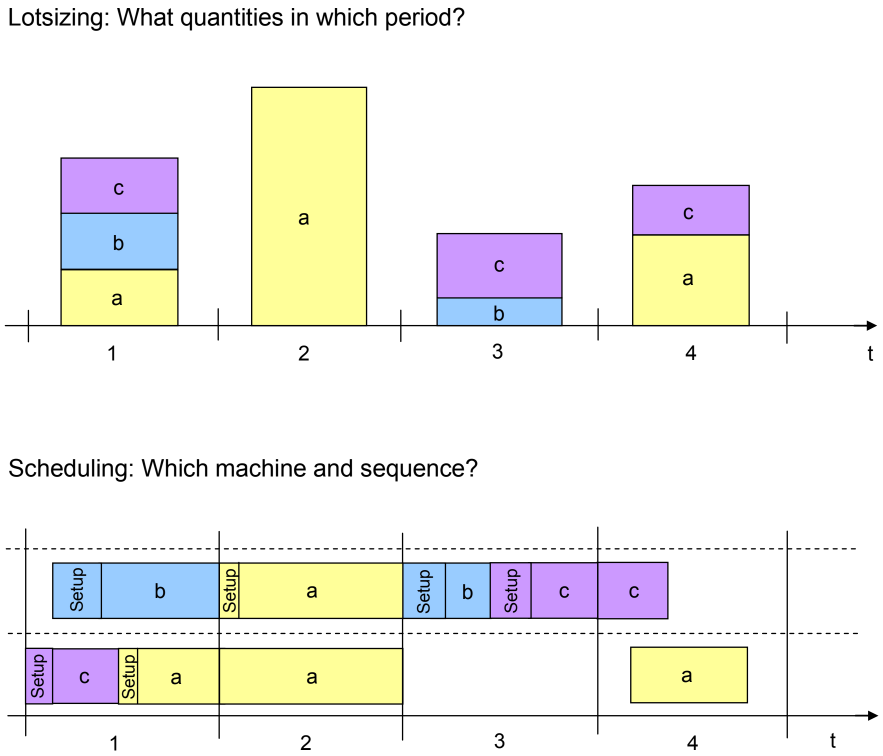


Figure 6: The relationship between lotsizing and scheduling[11]

[10] Quadt, 2004, p. 3

[11] Quadt, 2004, p. 4

3 Overview of lotsizing and scheduling problems

First of all a general problem description of the lotsizing and scheduling problem is given.

Lotsizing and scheduling problems have the objective to make an optimal decision about the lotsizes of the items and also the sequence of the items on the machine to make the production plan feasible.

In the lotsizing part there are items with a given demand that have to be produced on a production facility. The problem consists in determining the item quantity that has to be produced in each period of a given time horizon for every demanded item. The demand has to be satisfied while remaining within the capacity of the production facility and looking at the objective which is the minimization of costs (production costs, inventory costs and/or set-up costs).

In the scheduling part decisions are made about when and on which machine the items (jobs) are processed. Each job has a release date and a due date. The aim of the scheduling problem is to find a feasible plan, what means the sequencing of the items, which minimizes the costs from the lotsizing problem.

An overview of the lotsizing and scheduling problem is shown in figure 7:

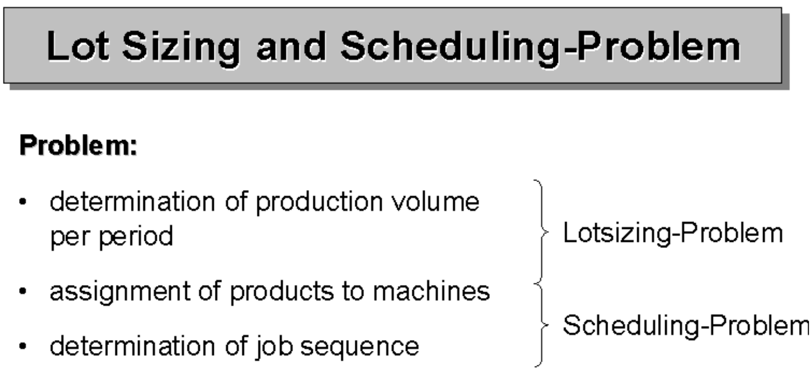


Figure 7: An overview of the lotsizing and scheduling problem[12]

[12] http://www.mathematik-21.de/statussem/poster/quadt/quadt.ppt#258,6,Lot Sizing and Scheduling-Problem, 08.05.2007

There are many lotsizing and scheduling problems named in the literature. Before the detailed models are going to be explained a short overview:

One of the first works on lotsizing was the economic order quantity model (see chapter 4.1) by Harris in 1913 which is based on constant demand. In 1958 the Wagner-Within model (see chapter 4.2) for dynamic demand followed. An ongoing development of this model is the Capacitated Lotsizing Problem. This is a capacitated model where only one item can be produced per period, and the setup can only occur at the beginning of a period or not at all. A setup carry-over is not possible. The model is shown in chapter 5.1. The Capacitated Lotsizing Problem with Linked Lotsizes is an extension of the Capacitated Lotsizing Problem. Here a setup can be carried-over from one period to the next one. It is explained in chapter 5.1.1. The Discrete Lotsizing and Scheduling Problem has a capacity constraint, the so called all-or-nothing-assumption. The constraint means that an item can only be produced over a full period or not at all. The model is defined in chapter 5.2. A similar model is the Continuous Setup Lotsizing Problem, but here the all-or-nothing-assumption is given up, what means that an item has not to be produced over a full period anymore. More details to this model in chapter 5.3. The Economic Lotsizing and Scheduling Problem is a model that shall find cyclical production schedules. Here no two items can be produced at the same time. More details on this model in chapter 5.4. Another development followed 1994, the Proportional Lotsizing and Scheduling Problem. In this model the setup can take place at the beginning or the end of a period and capacity splitting is possible. The model is shown in chapter 5.5. In 1997 the General Lotsizing and Scheduling Problem by Fleischmann and Meyr was introduced. In this model the periods are divided into sub-periods and because of that more than one setup per period is allowed. The model is defined in chapter 5.7. Based on this problem Meyr then developed the General Lotsizing and Scheduling Problem with Setup Times in 2000. It is an extension of the General Lotsizing and Scheduling Problem where setup costs are included into the model. As shown in chapter 5.8. All of the problems named above are single-level problems, what means that only one production level is considered. There is also literature on lotsizing and scheduling models for multi-level (see chapter 6) and multiple machines (see chapter 7) problems. Further on there is looked at hierarchical integration of lotsizing and scheduling (see chapter 8) by Dauzere-Peres and Lasserre.

3.1 Classification of problems

It depends on the lotsizing and scheduling model which problem criteria's are important.
Below some of the criteria's are listed[13]:

- Demand
  There are several demand types[14]:
  - static demand – dynamic demand
    (static: demand is stationary or constant, the value does not change over time; dynamic: the value changes over time)
  - deterministic demand - stochastic demand
    (deterministic: value of the demand, which can be static or dynamic, is known in advance; stochastic: value of the demand is not known in advance, it is random)
  - independent demand – dependent demand
    (independent: one items demand does not depend on decisions of an other items demand, the demand comes from outside for example customer demand; dependent: demand of one item depends on the demand of the previous item, there is a relationship between the item demands)

  Problems with dynamic demand and dependent demand are more complex than problems with static demand or independent demand.

- Number of items
  There are **single-item** and **multi-item** problems. In single-item problems there is only one end item, the final product. While in multi-item problems there are several end items. The complexity of multi-item problems is higher than from the single-item problem.

- Number of levels
  There are single-level and multi-level problems. In the single-level problem the end product is directly produced from the raw material, or for

[13] Haase, 1994, p. 3

[14] Karimi et al., 2003, p. 366

simplification the intermediate steps from the start of the production to the end are disregarded. Only one operation is needed. The demand therefore is independent. In the multi-level problem the raw material is changed in more then one operation to a final product. There are so called parent-component relationships between the items. The demand at one level depends on the demand of another level, the parent level. This demand is dependent.

Multi-level problems are more difficult to solve than single-level problems. These problems have multi-level product structures. Generally there are four structures, the serial product structure, the assembly product structure, the arborescent product structure and the general product structure (see chapter 6.1).

- Planning horizon

  The planning horizon is the length of time which a production schedule plans into the future. The horizon can be finite (a given number of periods with the same length) with a dynamic demand or infinite (an endless operating procedure is assumed) with a stationary demand.

  Another planning horizon is the rolling horizon. There only the first period is implemented. Then an update of the demand takes place. With this information the production plan is calculated again for the same number of periods. That takes place at every beginning of a new period (see figure 8)[15].

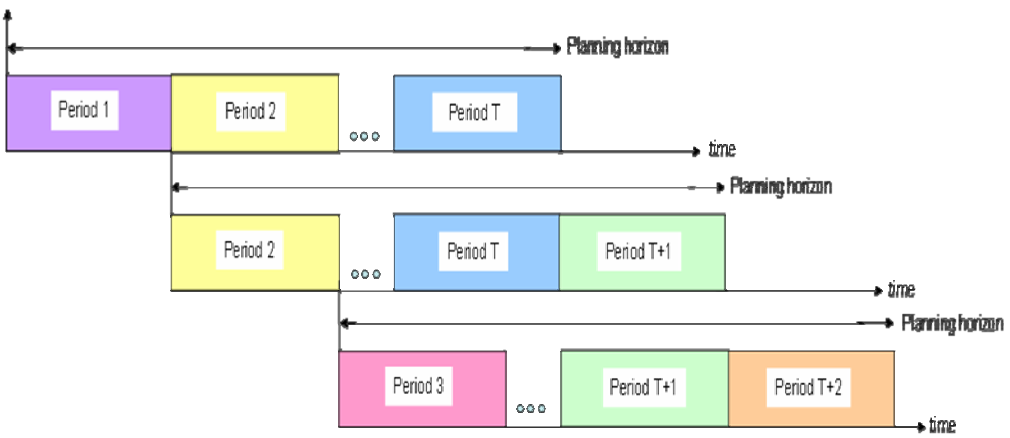


Figure 8: Rolling Horizon[16]

[15] Dauzere-Peres, Lasserre, 1994, p.94

[16] see footnote 15

- Time scale

  The time scale can be discrete or continuous. The discrete time scale can be for example small like hours or shifts, or large like weeks or months. There are two categories for the time periods, which are big bucket and small bucket problems. A big bucket problem has long time periods where several items can be produced in each period, while a small bucket problem has short time periods where only one item can be produced per period.

  The continuous time scale models are the Economic Order Quantity Model (see chapter 4.1) and the Economic Lotsizing and Scheduling Problem (see chapter 5.4). The periods of the continuous time scale models are real, for example 2.7 periods.

- Relevant costs

  Relevant costs are for example setup related costs, inventory related costs and capacity related costs.

  Setup related costs are costs that arise when a setup takes place for an item on a machine. These costs can be item specific costs, sequence dependent costs, start up costs or reservation costs.

  Inventory related costs are costs for keeping an item in inventory during one or more periods. These costs can be holding costs or shortage costs, which are costs that result from not having inventory, or not enough inventory. Inventory related costs are capital costs, they reduce the capital.

  Capacity related costs are costs for using the given capacity or when needed extra capacity. In the second case an example is overtime costs.

- Capacity or resource constraints

  There are two kinds of problems. One is the uncapacitated problem where no restrictions are and the other is the capacitated problem where capacity constraints are stated. Capacitated problems are more complex than uncapacitated ones.

  Resource constraints are for example machine capacity, transportation capacity and the number of workers.

- Setup structure

  There are two setup structures, the simple and the complex structure. When the setup time and setup costs of a period are independent of the sequence of the previous period the setup structure is called simple, but when they are dependent the structure is called complex.

  The complex setup structure is more difficult to solve than the simple one.

- Service policy

  Here are decisions made in case of arising shortages. Then for example backlogging, backorders or stockouts can be used. Problems with shortages are more complex than without shortages.

- Objectives

  The most common objective is to minimize the total costs. Other ones are for example maximization of the service level or maximization of the capacity utilisation.

4 Single-level Uncapacitated Lotsizing and Scheduling Problems

This chapter gives a detailed review of the single-level uncapacitated lotsizing and scheduling models mentioned in chapter 2.

4.1 Economic Order Quantity Model (EOQ)[17]

The Economic Order Quantity Model, developed by Harris in 1913, defines the optimal production quantity that minimizes the total costs of inventory. Harris appointed the problem as "How many parts to make at once"[18]. The model assumes no capacity constraints, infinite planning horizon, continuous time scale, constant demand, no inventory shortages, no backlogging and constant setup and inventory holding costs. The Economic Order Quantity Model is a single-level single-item uncapacitated lotsizing model.

The variables are: $q*$ = optimal order quantity

D = demand of the product

s = costs per order

h = holding cost per unit and per month

TC = total costs

The formula for the optimal order quantity ($q*$) is[19]:

$$TC(q) = \frac{D}{q} * s + \frac{q}{2} * h$$

$$\frac{\mathrm{d}(TC(q))}{\mathrm{d}q} = -\frac{D * s}{q} + \frac{h}{2}$$

$$q* = \sqrt{\frac{2 * D * s}{h}}$$

[17] Salomon, 1991, p. 23

[18] Harris, 1990, p. 947

[19] Skriptum Grundzüge der BWL IV, WS 97/98, p. 58

The graphical presentation of the optimal order quantity is shown in figure 5 (page 6).

An example for the EOQ – model is given by:
There is one product with demand D, ordering cost s and holding cost h which are given below.

D = 100
s = 200
h = 1

The optimal order quantity $q*$ is: $q* = \sqrt{\frac{2*100*200}{1}}$ = 200

4.2 Wagner-Within Model (WW)

The Wagner-Within Model, developed by Wagner and Within in 1958, is an extension of the Economic Order Quantity Model. It is a model with a finite time horizon, no capacity constraints and the demand is dynamic. It is a dynamic algorithm for a single-level single-item uncapacitated lotsizing model. The Wagner-Within Model finds an optimal solution, by using a dynamic programming algorithm[20].

The decision variables are:

x_t = production quantity in period t

I_t = inventory at the end of period t

The parameters are:

d_t = demand in period t

T = number of periods

$v_t(x)$ = production costs of x units in the period t

$h_t(I)$ = inventory costs of I units at the end of the period t

The mathematical formulation of the WW model is[21]:

$$Min \quad \sum_{t=1}^{T}(v_t(x_t) + h_t(I_t)) \qquad (4.1)$$

The objective (4.1) is to minimize the production costs and inventory costs.

$$I_{t-1} + x_t - d_t = I_t \qquad t = 1,...,T \qquad (4.2)$$

The equation (4.2) shows the inventory balance.

$$I_0 = I_T = 0 \qquad (4.3)$$

This equation (4.3) tells that the starting and the ending inventory is zero.

[20] Salomon, 1991, p. 23

[21] Salomon, 1991, p. 24

$$x_t \geq 0 \qquad\qquad t = 1,...,T \qquad\qquad (4.4)$$

Here the non-negativity condition (4.4) is given.

An example for the WW is given by[22]:

There is one product with demands d_t (T = 4 periods), ordering cost s and holding cost h which are given below.

d_t = (20, 40, 20, 30)

s = 70

h = 1

The solution is given in table 1:

t	1	2	3	4
2	70	-	-	-
3	110	140	-	-
4	150	160	180	-
E	240	220	210	220

Table 1: Solution for the WW example[23]

The optimal order quantities q_t and the total costs TRC are:

q_t = (60, 0, 50, 0)

TRC = 210

[22] Skriptum Grundzüge der BWL IV, WS 97/98, p. 64

[23] see footnote 22

5 Single-level Capacitated Lotsizing and Scheduling Problems

This chapter gives a detailed review of the single-level capacitated lotsizing and scheduling models mentioned in chapter 2.

5.1 Capacitated Lotsizing Problem (CLSP)

The Capacitated Lotsizing Problem is a single-level multi-item capacitated lotsizing model. It is an extension of the Wagner-Within Model with capacity constraints. The CLSP is a large bucket problem[24], also called big bucket problem, because several items requiring different setup states can be produced in each period. The planning horizon is finite, demand is dynamic and there are no shortages allowed. The time scale is weeks or months.
The CLSP is based on a fundamental assumption[25]:
There is exactly one setup per period for each item which is produced in the period. The setup occurs at the beginning of the period. If not then there is no setup at all. That means that a lot never continues over periods, and that at the beginning of every period a new setup has to occur.

The decision variables for the CLSP are:

I_{jt} = inventory for item j at the end of period t

q_{jt} = production quantity for item j in period t

x_{jt} = binary variable which tells if a setup for item j occurs in period t or not

The parameters for the CLSP are:

C_t = capacity of machine in period t

d_{jt} = demand for item j in period t

h_j = holding costs for item j

I_{j0} = inventory for item j at the beginning period

J = number of items

[24] Staggemeier, Clark, http://www.cms.uwe.ac.uk/~arclark/ifors.pdf, 05.09.2006

[25] Fleischmann, 1990, p. 337

p_j = capacity needs for the production of one unit of item j

s_j = setup costs for item j

T = number of periods

The mixed-integer model for the CLSP can be formulated as followed[26]:

$$Min \quad \sum_{j=1}^{J}\sum_{t=1}^{T}(s_j x_{jt} + h_j I_{jt}) \tag{5.1}$$

The objective (5.1) of the Capacitated Lotsizing Problem is to minimize the total costs.

$$I_{jt} = I_{j(t-1)} + q_{jt} - d_{jt} \qquad j = 1,...,J \quad t = 1,...,T \tag{5.2}$$

The equation (5.2) shows the inventory constraint.

$$p_j q_{jt} \le C_t x_{jt} \qquad j = 1,...,J \quad t = 1,...,T \tag{5.3}$$

This restriction (5.3) shows that an item can only be produced when the machine is setup for this item.

$$\sum_{j=1}^{J} p_j q_{jt} \le C_t \qquad t = 1,...,T \tag{5.4}$$

Here the capacity constraints (5.4) are mentioned.

$$x_{jt} \in \{0,1\} \qquad j = 1,...,J \quad t = 1,...,T \tag{5.5}$$

The setup variables (5.5) are binary.

$$I_{jt}, q_{jt} \ge 0 \qquad j = 1,...,J \quad t = 1,...,T \tag{5.6}$$

Here the non-negativity conditions (5.6) are given.

The Capacitated Lotsizing Problem does not include scheduling decisions. Usually the CLSP is solved first and after that the scheduling of each period is done[27].

[26] Drexl, Kimms, 1999 (1997), p. 224

[27] Fleischmann, 1990, p. 337

An example for the CLSP is given by[28]:

The number of items J = 3, there are T = 5 periods and the capacity of the machine in period t is C_t = 100. The demands d_{jt}, capacity needs p_j, holding costs h_j and setup costs s_j are given in the table 2 below.

t	1	2	3	4	5	p_j	h_j	s_j
d_{1t}	30		80		40	1	4	400
d_{2t}			30		70	1	3	150
d_{3t}			40		60	1	2	100

Table 2: Data for the CLSP example[29]

The optimal order quantities q_{jt} and the objective function value $Z*$ are given in table 3:

t	1	2	3	4	5	$Z*$
q_{1t}	30		90		30	
q_{2t}		30			70	
q_{3t}		40		60		
						2070

Table 3: Solution for the CLSP example[30]

Comments to the example:

The CLSP is a big bucket problem, because several items requiring different setup states can be produced in each period, what can be seen for example in period 2 where two different items (item 2 and 3) are produced with different setup costs s_j. Only one different item can be produced per period, what means only two items like item 2 and 3 in period 2 or item 1 and 2 in period 5.

[28] Haase, 1994, p. 20

[29] see footnote 28

[30] see footnote 28

The CLSP is a NP-hard problem to solve optimal, and because of that heuristics were developed.

There are two classes of heuristics[31]:

- the single-resource heuristics
- and the mathematical-programming-based heuristics.

The single-resource heuristics, also called common sense heuristics, are divided into two groups. The first one is the period-by-period heuristic, where the problem is looked at from the first period to the last period. The second one is the improvement heuristic. Here the model starts with a solution, which can be feasible or not, and then tries to improve it by local improvement steps. So the model gets a feasible solution.

The mathematical-programming-based heuristics are divided into relaxation heuristics, branch and bound procedures and LP-based heuristics. These models are optimum seeking.

5.1.1 Capacitated Lotsizing Problem with Linked Lotsizes (CLSPL)

The Capacitated Lotsizing Problem was criticized because the model does not allow a setup to be carried over from one period to the next, even if the last product in one period and the first product in the next period are the same. This has led to the Capacitated Lotsizing Problem with Linked Lotsizes model, which is an extension of the CLSP, and so also a big bucket problem. In the CLSPL model the setup can be carried-over from one period to the next period. At most one setup for one product can be carried-over, so that no new setup is necessary in the second period. This preservation of setup state is a characteristic of a small bucket problem, so the model combines the small and the big bucket problem[32] (see figure 9).

[31] Maes, Van Wassenhove, 1988, p. 993

[32] Suerie, Stadler, 2003, p. 1040

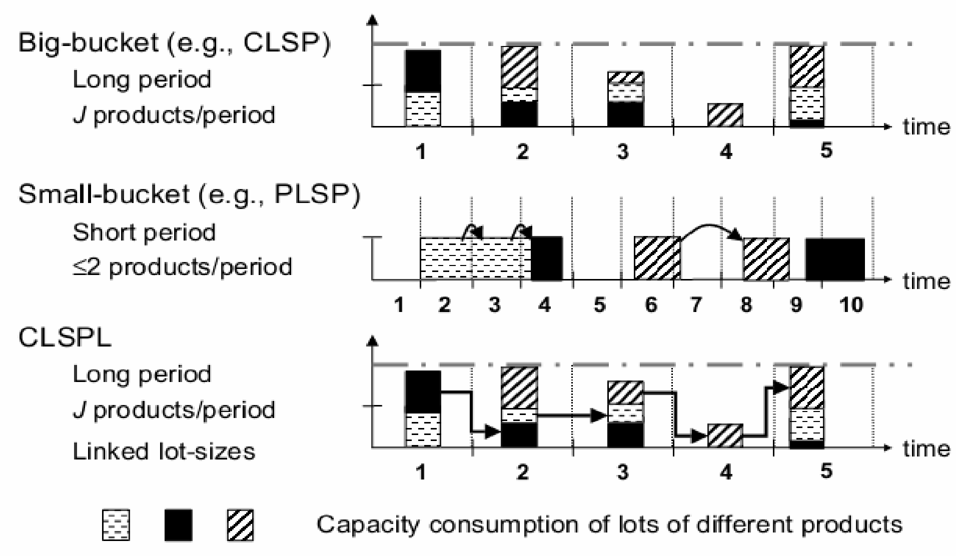


Figure 9: Characteristics of small bucket problem, big bucket problem and CLSPL[33]

The decision variables and the parameters for the CLSPL are the same as for the CLSP.

There are two new decision variables:

v_t = this variable shows if more than one item is produced in period t or not

z_{jt} = the binary variable tells if quantities of item j in period $t-1$ and the following period t are linked or not

x_{jt} = binary variable which tells if a quantity for item j is produced in period t or not

The mixed-integer model for the CLSPL can be formulated as followed[34]:

$$Min \quad \sum_{j=1}^{J}\sum_{t=1}^{T}\left[s_j\left(x_{jt}-z_{jt}\right)+h_j I_{jt}\right] \qquad (5.7)$$

The objective (5.7) of the Capacitated Lotsizing Problem with Linked Lotsizes Problem is to minimize the total costs. The difference to the objective function of the CLSP is $s_j\left(x_{jt}-z_{jt}\right)$ what means that the setup of an item is only calculated, when that item in a period is not linked with the production quantity of the previous period.

[33] see footnote 32

[34] Haase, 1994, p. 18

The following restrictions (5.8) – (5.11) are the same as in the CLSP model:

$$I_{jt} = I_{j(t-1)} + q_{jt} - d_{jt} \qquad j = 1,...,J \quad t = 1,...,T \qquad (5.8)$$

The equation (5.8) shows the inventory constraint.

$$p_j q_{jt} \le C_t x_{jt} \qquad j = 1,...,J \quad t = 1,...,T \qquad (5.9)$$

This restriction (5.9) shows that an item can only be produced when the machine is setup for this item.

$$\sum_{j=1}^{J} p_j q_{jt} \le C_t \qquad \text{t}t = 1,...,T \qquad (5.10)$$

Here the capacity constraints (5.10) are mentioned.

$$I_{jt}, q_{jt} \ge 0 \qquad j = 1,...,J \quad t = 1,...,T \qquad (5.11)$$

Here the non-negativity conditions (5.11) are given.

New restrictions for the CLSPL are:

$$\sum_{j=1}^{J} z_{jt} \le 1 \qquad t = 1,...,T \qquad (5.12)$$

This constraint tells that at most one setup can be carried over from one period to the next one.

$$z_{jt} - x_{j,t-1} \le 0 \qquad j = 1,...,J \quad t = 1,...,T \qquad (5.13)$$

This constraint shows that a setup can be carried over from one period to the next one when the item was setup in the period $t-1$.

$$z_{jt} - x_{jt} \le 0 \qquad j = 1,...,J \quad t = 1,...,T \qquad (5.14)$$

This three restrictions (5.12) – (5.14) show that only one product can be produced at the end of a period and so be linked to the following period.

$$1 - \sum_{j=1}^{J} x_{jt} + Jv_t \geq 0 \qquad t = 1,...,T \qquad (5.15)$$

$$z_{jt} + z_{j,t-1} + v_{t-1} \leq 2 \qquad j = 1,...,J \quad t = 1,...,T \qquad (5.16)$$

These constraints (5.15) + (5.16) indicate that when more than one item is produced in a period, that is when $v_t > 0$, the setup can be either linked with the previous period $t-1$ or with the following period $t+1$ setup.

$$v_t \geq 0 \qquad t = 1,...,T \qquad (5.17)$$

Here the non-negativity condition (5.17) is given.

$$x_{jt}, z_{jt} \in \{0,1\} \qquad j = 1,...,J \quad t = 1,...,T \qquad (5.18)$$

The setup variables (5.18) are binary.

An example for the CLSPL is given by[35]:

The data (J, T, C_t, d_{jt}, p_j, h_j, s_j) is the same as in the example for the CLSP on page 23 (see table 2).

The optimal order quantities q_{jt} and the objective function value Z^* are given in table 4:

t	1	2	3	4	5	Z^*
q_{1t}	30 ∪	∪	80 ∪	40		
q_{2t}	30				70	
q_{3t}	40			30 ∪	30	
						1460

Table 4: Solution for the CLSPL example[36]

Comments to the example:

In the CLSPL quantities of an item in a period and in the following period can be linked, for example item 3 is linked in period 4 and 5. So the setup is carried-over and no new setup has to take place in period 5 for the item 3. In this

[35] Haase, 1994, p. 20

[36] see footnote 35

model more than two items can be produced per period, for example in period 1 there is production of all three items.

Knut Haase, who gave the name to the CLSPL, provided in 1994 a heuristic to solve the Capacitated Lotsizing Problem with Linked Lotsizes named Backward Add CLSPL (BACLSPL). This heuristic moves backwards from the last period to the first period and adds setup and lotsizing decisions to every period[37]. In 2001 a tabu search heuristic was presented. At that time the CLSPL was called Capazitated Lotsizing Problem with Setup Carryover (CLSP-SC)[38].

---

[37] Suerie, Stadler, 2003, p. 1040

[38] Suerie, Stadler, 2003, p. 1041

5.2 Discrete Lotsizing and Scheduling Problem (DLSP)

The Discrete Lotsizing and Scheduling Problem is a single-level multi-item capacitated lotsizing model. It is a small bucket problem, what means that only one item can be produced per period. The planning horizon is finite and demand is dynamic. There is a small time scale, what means hours, days or shifts.

The fundamental assumption for the DLSP is the **"all-or-nothing" assumption[39]:**
It says that one item is produced over a full period or not at all. This is a discrete production policy. There is no changeover allowed.
To produce one item per period the periods are divided in several subperiods. That means that the macro-period is divided into micro-periods. The setup occurs only at the beginning of a period. The setup state is lost over idle periods.
The DLSP considers the sequencing of the lots[40].

The decision variables and the parameters for the DLSP are the same as for the CLSP. As the production of a lot can now last more than one period, there is a new decision variable and a new parameter needed for the setup state in a certain period.

The decision variable is:
y_{jt} = variable which tells us if the machine is set up for item j in period t or not
The parameter is:
y_{j0} = tells us if the machine is set up for item j at the beginning of the period 1 or not

[39] Salomon et al., 1991, p. 801
[40] Fleischmann, 1990, p. 338

The mixed-integer model for the DLSP can be formulated as followed[41]:

$$Min \quad \sum_{j=1}^{J}\sum_{t=1}^{T}(s_j x_{jt} + h_j I_{jt}) \tag{5.19}$$

The objective (5.19) of the Discrete Lotsizing and Scheduling Problem is to minimize the total costs. The function is the same as for the CLSP.

$$I_{jt} = I_{j(t-1)} + q_{jt} - d_{jt} \qquad j = 1,...,J \quad t = 1,...,T \tag{5.20}$$

The equation (5.20) shows the inventory constraint.

$$p_j q_{jt} = C_t y_{jt} \qquad j = 1,...,J \quad t = 1,...,T \tag{5.21}$$

This equation (5.21) is the "all-or-nothing" assumption. The difference to the CLSP is that the two sides have to equal.

$$\sum_{j=1}^{J} y_{jt} \le 1 \qquad t = 1,...,T \tag{5.22}$$

The restriction (5.22) makes sure that only one item can be produced per period.

$$x_{jt} \ge y_{jt} - y_{j(t-1)} \qquad j = 1,...,J \quad t = 1,...,T \tag{5.23}$$

The inequality (5.23) shows the beginning of a new lot.

$$y_{jt} \in \{0,1\} \qquad j = 1,...,J \quad t = 1,...,T \tag{5.24}$$

The setup state variables (5.24) are binary.

$$I_{jt}, q_{jt}, x_{jt} \ge 0 \qquad j = 1,...,J \quad t = 1,...,T \tag{5.25}$$

Here the non-negativity conditions (5.25) are given.

[41] Drexl, Kimms, 1999 (1997), p. 225

An example for the CLSP is given by[42]:

The number of items J = 3, there are T = 10 periods and the capacity of the machine in period t is C_t = 50. The demands d_{jt}, capacity needs p_j, holding costs h_j and setup costs s_j are given in table 5 below.

t	1	2	3	4	5	6	7	8	9	10	p_j	h_j	s_j
d_{1t}		30				80				40	1	2	400
d_{2t}						30				70	1	1,5	150
d_{3t}						40				60	1	1	100

Table 5: Data for the DLSP example[43]

The optimal order quantities q_{jt} and the objective function value $Z*$ are given in table 6:

t	1	2	3	4	5	6	7	8	9	10	$Z*$
q_{1t}		50	50	50							
q_{2t}						50				50	
q_{3t}					50				50		
											2140

Table 6: Solution for the DLSP example[44]

Comments on the example:

The DLSP has the "all-or-nothing" assumption, what means that one item is produced over a full period or not at all. In the example always only one item is produced per period, for example item 1 in period 2, item 1 in period 3, item 1 in period 4 and so on. When production takes place then the whole capacity of C_t = 50 is used. The setup state is lost over idle periods and because of that the objective function value is high.

[42] Haase, 1994, p. 28

[43] see footnote 42

[44] Haase, 1994, p. 29

The following Gantt chart (see figure 10) shows the schedules for the example above:

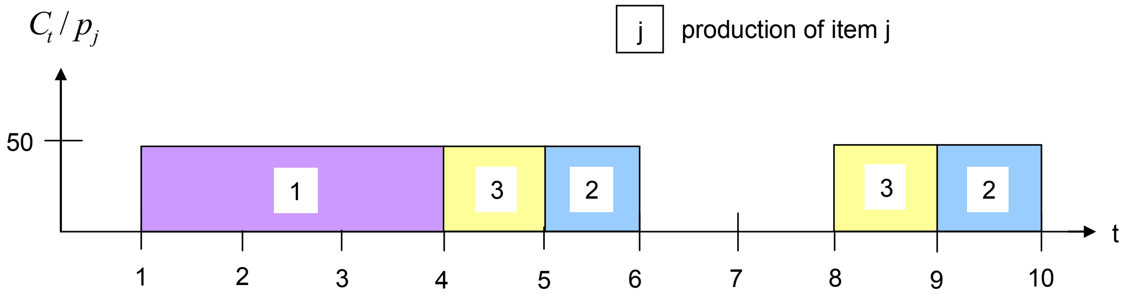


Figure 10: Gantt chart for the DLSP example[45]

The chart shows that item 1 is scheduled from period 2 to period 4, then item 3 in period 5 and item 2 in period 6. Afterwards again item 3 in period 9 and item 2 in period 10. The scheduled periods are always fully used because of the "all-or-nothing" assumption.

The advantage of the DLSP over the CLSP is that the setup state is not lost when the same item is produced over adjacent periods. In the CLSP a new setup is needed in every period.

To solve the Discrete Lotsizing and Scheduling Problem optimal is NP-hard. It can be solved by models based on integer programming formulations. Bernhard Fleischmann used in 1990 a Lagrangean relaxation and a dynamic programming algorithm for the DLSP[46].

[45] see footnote 44

[46] Fleischmann, 1990. p. 340

5.3 Continuous Setup Lotsizing Problem (CSLP)

The Continuous Setup Lotsizing Problem is a sequence independent kind of a small bucket problem, what still means that only one item can be produced per period, but the "all-or-nothing" assumption is given up[47]. That is the difference to the Discrete Lotsizing and Scheduling Problem. It is single-level multi-item capacitated model. The planning horizon is finite and demand is dynamic.

The decision variables and the parameters for the CSLP are the same as for the DLSP.

The mixed-integer model for the CSLP can be formulated as followed[48]:

$$Min \quad \sum_{j=1}^{J}\sum_{t=1}^{T}(s_j x_{jt} + h_j I_{jt}) \qquad (5.26)$$

$$I_{jt} = I_{j(t-1)} + q_{jt} - d_{jt} \qquad j = 1,...,J \quad t = 1,...,T \qquad (5.27)$$

$$p_j q_{jt} \le C_t y_{jt} \qquad j = 1,...,J \quad t = 1,...,T \qquad (5.28)$$

$$\sum_{j=1}^{J} y_{jt} \le 1 \qquad t = 1,...,T \qquad (5.29)$$

$$x_{jt} \ge y_{jt} - y_{j(t-1)} \qquad j = 1,...,J \quad t = 1,...,T \qquad (5.30)$$

$$y_{jt} \in \{0,1\} \qquad j = 1,...,J \quad t = 1,...,T \qquad (5.31)$$

$$I_{jt}, q_{jt}, x_{jt} \ge 0 \qquad j = 1,...,J \quad t = 1,...,T \qquad (5.32)$$

The models for the DLSP and the CSLP are nearly equal. The only difference is in the restriction (5.28) $p_j q_{jt} \le C_t y_{jt}$, where now no equality has to be anymore. An item has not to be produced over a full period. The capacity of a period which is not fully used stays unused.

[47] Brahimi et al., 2006, p. 4

[48] Drexl, Kimms, 1999 (1997), p. 226

An example for the CSLP is given by[49]:

The data (J, T, C_t, d_{jt}, p_j, h_j, s_j) is the same as in the example for the DLSP on page 31 (see table 5).

The optimal order quantities q_{jt} and the objective function value Z^* are given in table 7:

t	1	2	3	4	5	6	7	8	9	10	Z^*
q_{1t}		30	30	50						40	
q_{2t}						30	20	50			
q_{3t}					50				50		
											1910

Table 7: Solution for the CSLP example[50]

Comments on the example:

In the CSLP still only one item can be produced per period, but the "all-or-nothing" assumption is given up. In the example there is only one item per period like item 1 in period 2, item 1 in period 3 and so on, but the full capacity does not have to used anymore. A negative point of this model is that remaining capacity of a period stays unused. In the period 1 only 30 of the machine capacity is needed and so 20 is left unused.

The following Gantt chart (see figure 11) shows the schedules for the example above:

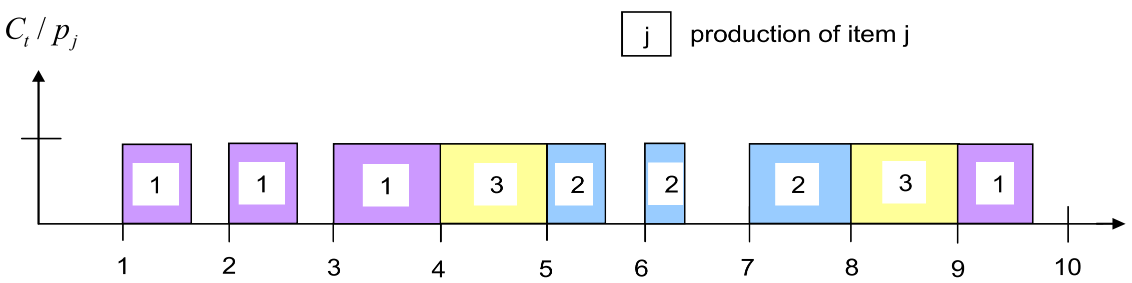


Figure 11: Gantt chart for the CSLP example[51]

[49] Haase, 1994, p. 29

[50] see footnote 49

The chart shows that item 1 is scheduled in period 2, 3 and 4, but the periods 2 and 3 are not fully used because the “all-or-nothing” assumption is given up. Then item 3 in period 5 and item 2 in period 6, 7 and 8, where period 6 and 7 are again not fully used. After that item 3 in period 9 and item 2 in period 10.

The advantage of the CSLP over the DLSP is that when the same item is produced over more periods the setup only takes place in the first period. Then the setup state stays, although a period has not to be fully used. In the DLSP there would be two setups because of the not fully used period.

The CSLP is a NP-hard problem to solve optimally. Karmarkar presented an algorithm based on Lagrangean relaxation in combination with dynamic programming to solve the CSLP[52].

[51] see footnote 49

[52] Salomon, 1991, p. 35

5.4 Economic Lotsizing and Scheduling Problem (ELSP)

The Economic Lotsizing and Scheduling Problem is a capacitated model for a single-level multi-item machine. The planning horizon for the ELSP is infinite, the time scale is continuous, the demand is constant, and backlogging is not allowed. The determination of the lotsizes and the sequencing of the lots (the cycles) are done simultaneously. The objective of the model is to minimize the sum of setup costs and inventory holding costs, while determining the cyclical production schedule[53].
In the model no two items can be produced at the same time, otherwise there would be the phenomenon of interference[54], what is physically impossible.

The Economic Lotsizing and Scheduling Problem to solve optimal is a NP-hard.

There are three analytical approaches to solve the ELSP[55]:

1. The Common Cycle (CC) or Rotation Cycle approach
   Here all product cycle times have to be equal. That means that each product can be produced exactly once in every cycle.
2. The Basic Period (BP) approach
   Here are different cycle times for different products allowed.
3. The Dobson's approach
   Here the lotsizes of a given product can vary over a cyclic schedule.

Further there are also heuristic approaches[56], for example the Madigan's procedure, a heuristic from Stankard and Gupta, a heuristic from Doll and Whybark and a heuristic from Goyal.
The heuristic approaches lead to a better performance then the analytical approaches.

---

[53] Salomon, 1991, p. 27

[54] Hsu, 1983, p. 93

[55] Carreno, 1990, p. 349

[56] Elmaghraby, 1978, p. 591-594

5.5 Proportional Lotsizing and Scheduling Problem (PLSP)

The Proportional Lotsizing and Scheduling Problem is a small bucket problem, where at most two items can be produced per period. It is single-level multi-item capacitated model. The planning horizon is finite and demand is dynamic[57].

The fundamental assumption for the PLSP is:
"At most one changeover of the setup is allowed within each period." [58]
If the capacity of a period is not fully used from one item, the remaining capacity can be used for a second item, but there is no more than one changeover allowed. When there are two items in one period the order in which they are produced is important. The first item must be the same as the last item in the previous period.
The setup can occur at the beginning of a period or at the end of a period. There is only one setup per period, but it can be carried over from the previous period to the next period. The setup state is preserved over idle periods.

The decision variables and the parameters for the PLSP are the same as for the DLSP.

The mixed-integer model for the PLSP can be formulated as followed[59]:

$$Min \quad \sum_{j=1}^{J}\sum_{t=1}^{T}(s_j x_{jt} + h_j I_{jt}) \tag{5.33}$$

$$I_{jt} = I_{j(t-1)} + q_{jt} - d_{jt} \qquad j = 1,\dots,J \quad t = 1,\dots,T \tag{5.34}$$

$$p_j q_{jt} \le C_t (y_{j(t-1)} + y_{jt}) \qquad j = 1,\dots,J \quad t = 1,\dots,T \tag{5.35}$$

$$\sum_{j=1}^{J} p_j q_{jt} \le C_t \qquad t = 1,\dots,T \tag{5.36}$$

$$\sum_{j=1}^{J} y_{jt} \le 1 \qquad t = 1,\dots,T \tag{5.37}$$

---

[57] Drexl, Haase, 1995, p. 74

[58] see footnote 57

[59] Drexl, Kimms, 1999 (1997), p. 227

$$x_{jt} \geq y_{jt} - y_{j(t-1)} \qquad j = 1,...,J \quad t = 1,...,T \qquad (5.38)$$

$$y_{jt} \in \{0,1\} \qquad j = 1,...,J \quad t = 1,...,T \qquad (5.39)$$

$$I_{jt}, q_{jt}, x_{jt} \geq 0 \qquad j = 1,...,J \quad t = 1,...,T \qquad (5.40)$$

The models for the PLSP and the CSLP are nearly equal. The difference is in the restriction (5.35) $p_j q_{jt} \leq C_t(y_{j(t-1)} + y_{jt})$. This inequality shows that the production of an item in a period only takes place when there is a setup either at the beginning or the end of the period.

A new constraint (5.36) is $\sum_{j=1}^{J} p_j q_{jt} \leq C_t$. It secures that when two items are produced per period the capacity requirements are not harmed. This splitting of capacity to produce two items within a period proportional to the quantities leads to the name of the Proportional Lotsizing and Scheduling Problem[60].

An example for the PLSP is given by[61]:

The data (J, T, C_t, d_{jt}, p_j, h_j, s_j) is the same as in the example for the DLSP on page 31 (see table 5).

The optimal order quantities q_{jt} and the objective function value $Z*$ are given in table 8:

t	1	2	3	4	5	6	7	8	9	10	$Z*$
q_{1t}		30		50	50	20					
q_{2t}						30		30	40		
q_{3t}	20	20							10	50	
											1710

Table 8: Solution for the PLSP example[62]

[60] Drexl, Haase, 1995, p. 75

[61] Haase, 1994, p. 29

[62] see footnote 61

Comments on the example:

In the PLSP two different items can be produced per period. The capacity of a period that is not fully used from one item can be used for a second item, but there is only one changeover allowed. In the period 6 for example item 1 needs 20 of the machine capacity, and so 30 stays unused. This 30 can be used for the item 2 in the same period. When there are two items in one period then the order in which they are produced is important, for example the item 3 in period 1 and 2. The first item has to be the same as the last item in the previous period. That means that in period 1 item 3 is produced and in period 2 item 3 is further produced and afterwards item 1. The setup for item 3 is carried-over from period 1 to period 2.

The following Gantt chart (see figure 12) shows the schedules for the example above:

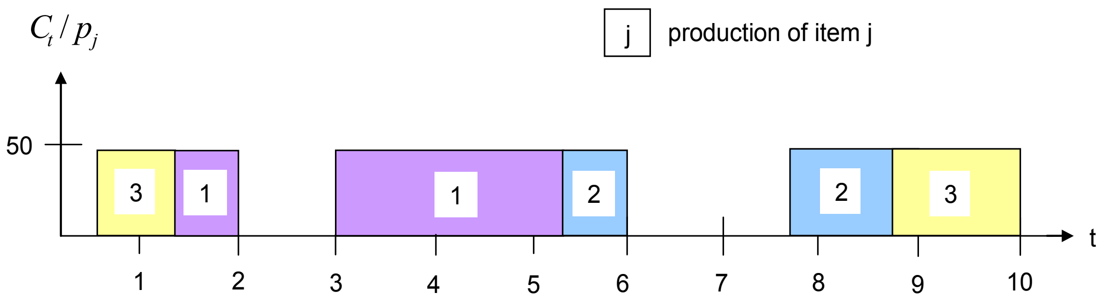


Figure 12: Gantt chart for the PLSP example[63]

The chart shows that item 3 is scheduled in period 1 and 2. Here the setup is carried-over from one period to the following one. In period 4, 5 and 6 item 1 is scheduled. The not used capacity from the item 1 in period 6 is used for the item 2. Then item 2 in period 8 and 9, and item 3 in period 9 and 10.

The advantage of the PLSP over the other models discussed until now is that the setup state is preserved over idle periods and that two different items can be produced per period.

To solve the PLSP a randomized-regret-based biased sampling method[64] can be used. This method consists of three steps. The first part is that the lotsizing

[63] see footnote 61

[64] Drexl, Haase, 1995, p. 76

and scheduling is done backwards from the last period to the first. The second step of the model is that in every period at most two items can be added, and the third one is that the regrets are used to decide in which period which item is going to be produced. These three parts describe a Backward Add method.

Some extensions of the PLSP are:

- Proportional Lotsizing and Scheduling Problem with Setup Times (PLSPST)
- Proportional Lotsizing and Scheduling Problem with Sequence Dependent Setup Costs (PLSPSDSC)

5.6 General Lotsizing and Scheduling Problem (GLSP)

The General Lotsizing and Scheduling Problem is a large bucket problem. It is single-level multi-item capacitated model. The planning horizon is finite, the demand is dynamic and no backlogging is allowed. In the GLSP the periods, so called macro-periods, are divided into micro-periods with variable length, which allows sequencing of the products. The macro-periods indicate the demand and the holding costs, and the micro-periods consist of the changes referred to the decision variables. The "all-or-nothing"-assumption exists for the micro-periods[65].

The fundamental assumption[66] for the GLSP is that **the number of lots per period is restricted**. This is done with the new parameter N_t.

The decision variables for the GLSP are basically the same as for the DLSP.

They are:

I_{jt} = inventory for item j at the end of period t

q_{jn} = production quantity for item j at the position n

x_{jn} = variable which tells if a setup for item j occurs at the position n or not

y_{jn} = variable which tells if the machine is ready to produce item j at the position n or not

The parameters for the GLSP are the same as for the DLSP, and the new one is:

N_t = is the maximum number of lots in period t

(When $N_t = 1$ for all $t = 1,...,T$, the GLSP is the same as the CSLP.)

---

[65] Fleischmann, Meyr, 1997, p. 12

[66] Drexl, Kimms, 1999 (1997), p. 227

Further in this model:

$F_t = 1 + \sum_{\tau=1}^{t-1} N_\tau$ = shows the first position in period t

$L_t = F_t + N_t - 1$ = shows the last position in period t

$N = \sum_{t=1}^{T} N_t$ = total number of positions t

The mixed-integer model for the GLSP can be formulated as followed[67]:

$$Min \quad \sum_{j=1}^{J}\sum_{n=1}^{N} s_j x_{jn} + \sum_{j=1}^{J}\sum_{t=1}^{T} h_j I_{jt} \tag{5.41}$$

The objective (5.41) is to minimize the inventory costs and the sequence dependent setup costs.

$$I_{jt} = I_{j(t-1)} + \sum_{n=F_t}^{L_t} q_{jn} - d_{jt} \qquad j = 1,...,J \quad t = 1,...,T \tag{5.42}$$

The equation (5.42) shows the inventory balance. An item now can be produced at several positions in a period.

$$p_j q_{jn} \le C_t y_{jn} \qquad j = 1,...,J \quad t = 1,...,T \quad n = F_t,...,L_t \tag{5.43}$$

This restriction (5.43) shows that a lot for an item on a position can only be produced when the machine is set up for it.

$$\sum_{j=1}^{J}\sum_{n=F_t}^{L_t} p_j q_{jn} \le C_t \qquad t = 1,...,T \tag{5.44}$$

Here the capacity constraints (5.44) are mentioned.

$$\sum_{j=1}^{J} y_{jn} \le 1 \qquad n = 1,...,N \tag{5.45}$$

The restriction (5.45) makes sure that there is unique setup state.

$$x_{jn} \ge y_{jn} - y_{j(n-1)} \qquad j = 1,...,J \quad n = 1,...,N \tag{5.46}$$

The inequality (5.46) shows the position where the setup takes place.

[67] Drexl, Kimms, 1999 (1997), p. 228

$$x_{jn}, y_{jn} \in \{0,1\} \qquad j = 1,...,J \quad n = 1,...,N \qquad (5.47)$$

The setup state variables (5.47) are binary.

$$I_{jt} \geq 0 \qquad j = 1,...,J \quad t = 1,...,T \qquad (5.48)$$

$$q_{jn}, x_{jn} \geq 0 \qquad j = 1,...,J \quad n = 1,...,N \qquad (5.49)$$

Here the non-negativity conditions (5.48) + (5.49) are given.

The GLSP is a NP-hard problem to solve optimally, because of that exact solution methods are too complex. So heuristics have to be used, which only can provide near-optimal solutions. The GLSP can be solved for example with local search heuristics based on threshold accepting[68].

5.6.1 GLSP – CS

The GLSP – CS is a variant of the General Lotsizing and Scheduling Problem. The shortcut CS means "Conservation of Setup State".

The mixed-integer model for the GLSP-CS[69] is:

$$Min \quad \sum_{i,j,s} s_{ij} z_{ijs} + \sum_{j,t} h_j I_{jt} \qquad (5.50)$$

The objective (5.50) is to minimize the inventory costs and the changeover costs. In this function s_{ij} is the setup costs for a changeover from the product i to the product j, and z_{ijs} shows if the changeover takes place at the beginning of the micro-period s or not.

The difference to the General Lotsizing and Scheduling Problem are new constraints (the constraints (5.42) – (5.45) from the GLSP stay the same):

$$x_{js} \geq m_j (y_{js} - y_{j(s-1)}) \qquad j = 1,...,J \quad s = 1,...,S \qquad (5.51)$$

$$z_{ijs} \geq y_{j(s-1)} + y_{js} - 1 \qquad j = 1,...,J \quad i = 1,...,I \quad s = 1,...,S \qquad (5.52)$$

---

[68] Fleischmann, Meyr, 1997, p. 11

[69] Fleischmann, Meyr, 1997, p. 12

where x_{js} is the quantity that is produced from the product j in the period s, m_j is the minimum lotsize for the product j and y_{js} shows if the machine is setup for the product j or not.
The constraint (5.51) shows the usage of minimum batches. The production quantity of an item of the first lot has to bigger as the minimum batch so that production can take place. Then the production can go on if the production quantity is smaller than the minimum batch. The restriction (5.52) points out the connection of setup state and changeover.

$$\sum_{i,j,i\neq j} z_{ijs-1} \geq \sum_{i,j,i\neq j} z_{ijs} \qquad t = 1,...,T \quad s = (f_t + 2),...,l_t \qquad (5.53)$$

$$x_{js} \leq \frac{K_t}{a_j}(2 - \sum_i z_{ijs-1} - z_{ijs}) \qquad j = 1,...,J \quad t = 1,...,T$$

$$s = (f_t + 1),...,l_t \qquad (5.54)$$

where K_t is the available capacity in period t, a_j is the needed capacity to produce one unit of product j, f_t is the first micro-period s of period t and l_t is the last micro-period s of period t.
The constraint (5.53) tells that idle micro-periods s have to be placed at the end of periods t, and (5.54) that only the first micro-period s in a period t can be non-idle.
After an idle period the machine remains the setup for the item that was produced last. So the setup state is conservated[70].

An example for the GLSP – CS is a saw mill. In a saw mill different lengths of wooden boards can be cute, like 10, 15 or 20 centimetres. These are the different items. To adjust the cutting machine for a length a technician has to come and modify the machine. This is the setup. When an idle period occurs, like no working over night, the setup state stays conservated.

[70] Fleischmann, Meyr, 1997, p. 13

5.6.2 GLSP – LS

The GLSP – LS is also a variant of the General Lotsizing and Scheduling Problem. The shortcut LS means “Loss of Setup State”.

In this model a new product is introduced with $j = 0$, what stands for idle periods.

The mixed-integer model for the GLSP-LS[71] is:

$$Min \quad \sum_{i,j,s} s_{ij} z_{ijs} + \sum_{j \neq 0,t} h_j I_{jt} \tag{5.55}$$

The objective function is the same as for the GLSP-CS only that the product j cannot be null ($j \neq 0$).

The model includes the constraints (5.43) and (5.45) from the GLSP and (5.51) and (5.52) from the GLSP-CS.

The difference to the GLSP is the alteration of the constraint (7.4):

$$\sum_{j,s} a_j x_{js} = K_t \qquad t = 1,...,T \tag{5.56}$$

The constraint (5.56) shows that the capacity has to be equal the production quantity times the production needs to produce one unit of product j. As the product $j = 0$ has no inventory holding costs additional capacity will be used by idle periods.
The machine does not keep the setup state over idle periods, it is lost.

An example for the GLSP – LS is an industrial robot. An industrial robot can produce different electronic items. It is controlled by a software system. Over idle periods the software is shut down to reduce machine hours. So the setup is lost. In the next active period a new setup occurs due booting of the software system.

[71] Fleischmann, Meyr, 1997, p. 13

5.7 General Lotsizing and Scheduling Problem with Setup Times (GLSPST)

The General Lotsizing and Scheduling Problem with Setup Times is an extension of the General Lotsizing and Scheduling Problem. The model is like the GLSP a single-level multi-item capacitated model with dynamic demand and a finite planning horizon. Also no backlogging is allowed. The objective of the GLSPST is to minimize the inventory holding costs and the sequence-dependent setup costs[72].

The only difference to the GLSP is the alteration of constraint (5.44), where now setup times are included in the model.
This is shown by[73]:

$$\sum_{j,s} a_j x_{js} + \sum_{i,j,s} st_{ij} z_{ijs} \leq K_t \qquad t = 1,...,T \tag{5.57}$$

The restriction (5.57) includes setup times st_{ij} into the capacity constraint. So the capacity is further reduced.

The GLSPST is a model for simultaneous lotsizing and scheduling. It can be solved by local search heuristics, like simulated annealing or threshold accepting, in combination with dual reoptimization[74] (see chapter 10).

---

[72] Meyr, 2000, p. 311

[73] Meyr, 2000, p. 313

[74] Meyr, 2000, p. 312

5.8 Batching and Scheduling Problem (BSP)

The BSP is single-level multi-item capacitated lotsizing and scheduling model. The time scale of the Batching and Scheduling Problem is continuous, the planning horizon is finite and demand is dynamic. Every demand is looked at as a job and is characterized by a deadline and its size. The size shows the processing time of the job. The capacity is constant, and because of that the processing times do not depend on the schedule. Important is also that job splitting is not allowed, so certain demand have to be processed in one part. The demands for the same item can be put together to build a lot and to save setup costs. This leads to the name “Batching” and Scheduling Problem[75].

The decision variables for the BSP are:

r_n = completion time of job n

x_{nk} = variable which shows that the job n is scheduled before the job k (the precedence of the jobs)

The parameters for the BSP are:

B = a big number

f_n = deadline for the job n

h_n = holding costs for the job n

N = number of jobs

p_n = processing time for the job n

s_{nk} = sequence dependent setup costs for the jobs

[75] Drexl, Kimms, 1999 (1997), p. 229

The mixed-integer model for the BSP can be formulated as followed[76]:

$$Min \quad \sum_{n=0}^{N}\sum_{\substack{k=1\\k\neq n}}^{N} s_{j(n)j(k)}x_{nk} + \sum_{j=n}^{N} h_{j(n)}p_n(f_n - r_n) \tag{5.58}$$

The objective (5.58) is to minimize the setup costs and the holding costs.

There are number $1,\ldots,N$ of jobs and the jobs 0 and $N+1$ are the dummy jobs. They are the first and the last job to be scheduled.

$$\sum_{\substack{k=1\\k\neq n}}^{N+1} x_{nk} = 1 \qquad n = 0,\ldots,N \tag{5.59}$$

The equality (5.59) tells that every job has exactly one successor. Only the job $N+1$ has none.

$$\sum_{\substack{k=0\\k\neq n}}^{N} x_{kn} = 1 \qquad n = 1,\ldots,N+1 \tag{5.60}$$

The equality (5.60) tells that every job has exactly one predecessor. The job $k=0$ has not got one, and because of that k starts with k =1.

$$r_n + p_k \leq r_k + B(1 - x_{nk}) \qquad n = 0,\ldots,N \quad k = 1,\ldots,N+1 \tag{5.61}$$

This restriction (5.61) shows that jobs do not overlap.

$$r_n \leq f_n \qquad n = 1,\ldots,N \tag{5.62}$$

This inequality (5.62) means that no backlogging can occur.

$$x_{nk} \in \{0,1\} \qquad n = 0,\ldots,N \quad k = 1,\ldots,N+1 \tag{5.63}$$

This decision variable (5.63) is binary.

$$r_n \geq 0 \qquad n = 1,\ldots,N+1 \tag{5.64}$$

Here the non-negativity condition (5.64) is given.

$$r_0 = 0 \tag{5.65}$$

[76] Drexl, Kimms, 1999 (1997), p. 229

The equation (5.65) shows that the completion time of the dummy job 0 is zero.

The constraints $\sum_{\substack{k=1 \\ k \neq n}}^{N+1} x_{nk} = 1$, $\sum_{\substack{k=1 \\ k \neq n}}^{N} x_{kn} = 1$ and $r_n + p_k \leq r_k + B(1 - x_{nk})$ in combination show the order among the jobs (5.59) – (5.61).

A BSP model developed by Jordan and Drexl is equivalent to the DLSP, and so the solution procedures for the BSP can be used to solve the DLSP[77].

[77] Karimi et al., 2003, p. 367

6 Multi-Level Lotsizing and Scheduling Problems (MLLSP)

The lotsizing and scheduling problems mentioned above are all single-level models. In the real world there are more complex cases to look at.

6.1 MLLSP - Structure

In multi-level lotsizing and scheduling problems there are multi-level product structures. Generally there are four structures[78]:

- serial product structure (every item has at most one predecessor and one successor, see figure 13)

Figure 13: A serial product structure[79]

- assembly product structure (also called convergent structure, every item has at most one successor, but may have any number of predecessors, see figure 14)

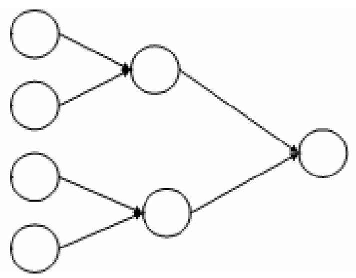


Figure 14: An assembly product structure[80]

[78] Salomon, 1991, p. 104

[79] Beamon, Chen, 2001, p. 3197

[80] see footnote 79

- arborescent product structure (also called divergent structure, every item has at most one predecessor, but may have any number of successors, see figure 15)

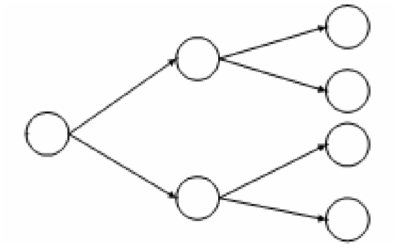


Figure 15: An aborescent product structure[81]

- general product structure (also called network structure, this structure does not fall into any of the preceding three structural classes, every item can have more than one successor or predecessor see figure 16)

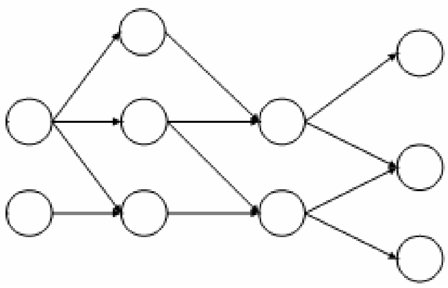


Figure 16: A general product structure[82]

A multi-level lotsizing and scheduling problem with a general product structure is NP-hard. In the MLLSP two types of demand occur. These are the dependent demand, where an item's demand does not depend on decisions of an other item's demand and the demand comes from outside (for example from the customer), and the independent demand, where the demand of one item depends on the demand of the previous item what means that there is a relationship between the demands. That is a difference to the single-level problem, where only independent demand is considered. The relevant costs are machine setup costs, inventory holding costs and production costs. The objective of the MLLSP is to minimize these costs[83].

[81] see footnote 79

[82] see footnote 79

[83] Salomon, 1991, p. 103-104

6.2 MLLSP - Example

Some examples for multi-level lotsizing and scheduling problems are:

- Multi-level Capacitated Lotsizing Problem (MLCLSP)
- Multi-level Discrete Lotsizing and Scheduling Problem (MLDLSP)
- Multi-level Proportional Lotsizing and Scheduling Problem (MLPLSP)

In the next part we look at the Multi-level Proportional Lotsizing and Scheduling Problem.

The decision variables for the MLPLSP are the same as for the single-level PLSP.

The parameters for the MLPLSP are:

a_{ji} = the "gozinto" factor, when the item i is no immediate successor of the item j then $a_{ji} = 0$, else it is the quantity of the item j that is needed to produce one item i

C_{mt} = capacity of the machine m that is available in period t

d_{jt} = independent demand for the item j in period t

h_j = holding costs for one unit of the item j to hold in one period in inventory

I_{j0} = inventory for the item j

Γ_m = set of all the items that use the same machine m

J = number of items

M = number of machines

m_j = machine on which the item j is produced

p_j = capacity that is needed to produce one unit of the item j

s_j = setup costs for the item j

S_j = the set of successors of the item j

T = number of periods

v_j = lead time of the item j

y_{j0} = unique setup state

The mixed-integer model for the MLPLSP can be formulated as followed[84]:

$$Min \quad \sum_{j=1}^{J}\sum_{t=1}^{T}(s_j x_{jt} + h_j I_{jt}) \tag{6.1}$$

The objective function (6.1) is the same as for the single-level PLSP.

$$I_{jt} = I_{j(t-1)} + q_{jt} - d_{jt} - \sum_{i \in S_j} a_{ji} q_{it} \qquad j = 1,...,J \quad t = 1,...,T \tag{6.2}$$

This equation (6.2) shows the inventory balance.

The inventory at the end of the period t consists of the inventory of the period $t-1$ plus the production in the period t minus the independent and the dependent demand.

$$I_{jt} \geq \sum_{i \in S_j} \sum_{\tau = t+1}^{\min\{t+v_j, T\}} a_{ji} q_{i\tau} \qquad j = 1,...,J \quad t = 0,...,T-1 \tag{6.3}$$

The restriction (6.3) takes lead times into account, so that the independent demand can be fulfilled.

$$p_j q_{jt} \leq C_{mjt}(y_{j(t-1)} + y_{jt}) \qquad j = 1,...,J \quad t = 1,...,T \tag{6.4}$$

$$\sum_{j \in \Gamma_m} p_j q_{jt} \leq C_{mt} \qquad m = 1,...,M \quad t = 1,...,T \tag{6.5}$$

$$\sum_{j \in \Gamma_m} y_{jt} \leq 1 \qquad m = 1,...,M \quad t = 1,...,T \tag{6.6}$$

$$x_{jt} \geq y_{jt} - y_{j(t-1)} \qquad j = 1,...,J \quad t = 1,...,T \tag{6.7}$$

$$y_{jt} \in \{0,1\} \qquad j = 1,...,J \quad t = 1,...,T \tag{6.8}$$

$$I_{jt}, q_{jt}, x_{jt} \geq 0 \qquad j = 1,...,J \quad t = 1,...,T \tag{6.9}$$

The other constraints (6.4) – (6.9) are the same as for the single-level PLSP.

The MLLSP is a NP-hard problem to solve, for example by Simulated Annealing or Tabu Search[85].

---

[84] Drexl, Kimms, 1999 (1997), p. 230

[85] Salomon, 1991, p. 121

7 Lotsizing and Scheduling Problem with Multiple Machines (LSPMM)

Lotsizing and scheduling problems can also occur on more than one machine. That case is shown in the following chapter.

7.1 Multiple Machine Types

There are two types of multiple machines, also called multiple facilities, which are machines in parallel and machines in series. In the case of parallel machines there is no interaction between the machines. Every machine satisfies its own demand. By serial machines the machines interact with each other. The output of one machine is the input for the next machine. The last machine then satisfies the demand. If an interaction takes place there occur transfer costs. These costs represent the quantities that have to be transferred from one machine to another during a period[86].

7.2 LSPMM - Example

Some examples for lotsizing and scheduling problems with multiple machines are:

- Capacitated Lotsizing Problem with Multiple Machines (CLSPMM)
- Discrete Lotsizing and Scheduling Problem with Multiple Machines (DLSPMM)
- Proportional Lotsizing and Scheduling Problem with Multiple Machines (PLSPMM)
- General Lotsizing and Scheduling Problem for Parallel Production Lines (GLSPPL)

[86] Brahimi et al., 2006, p. 7

In the next part we look at the General Lotsizing and Scheduling Problem for Parallel Production Lines.

The GLSPPL is an extension of the General Lotsizing and Scheduling Problem with Setup Times (GLSPST). The model is a multi-item capacitated model for non-identical parallel machines. It has dynamic demand and no backlogging is allowed. The periods t (macro-periods) are divided into sub periods s (micro-periods) with variable length.

The decision variables for the GLSPPL are:

I_{jt} = inventory for product j at the end of period t

x_{ljs} = production quantity for item j in micro-period s on the line l

y_{ljs} = binary variable which tells if the line l is setup for product j in micro-period s or not

z_{lijs} = shows if a changeover from product i to product j takes place or not on line l in micro-period s

The parameters for the GLSPPL are:

S_{lt} = set of micro-periods s on line l in period t

K_{lt} = capacity of the line l that is available in period t

d_{jt} = demand for the item j in period t

h_j = holding costs for one unit of the item j to hold in one period in inventory

I_{j0} = inventory for the item j

J = number of items

L = number of parallel production lines

a_{lj} = capacity that is needed to produce one unit of the item j on line l

m_{lj} = minimum lotsize of product j is produced when produced on line l

c_{lj} = production costs of product j on line l

s_{lij} = setup costs for the changeover from product i to product j on line l

st_{lij} = setup time for the changeover from product i to product j on line l

T = number of periods

y_{lj0} = tells if line l is setup for product j at the beginning of the period or not

The mixed-integer model for the GLSPPL can be formulated as followed[87]:

$$Min \quad \sum_{l,i,j,s} s_{lij} z_{lijs} + \sum_{j,t} h_j I_{jt} + \sum_{l,j,s} c_{lj} x_{ljs} \tag{7.1}$$

The objective function (7.1) of the GLSPPL minimizes the sequence-dependent setup costs, the inventory holding costs and the production costs.

$$I_{jt} = I_{j(t-1)} + \sum_{l,s} x_{ljs} - d_{jt} \qquad j = 1,...,J \quad t = 1,...,T \tag{7.2}$$

The equation (7.2) shows the inventory constraint. It is the only constraint that links the parallel lines together.

$$\sum_{j,s} a_{lj} x_{ljs} \leq K_{lt} - \sum_{i,j,s} st_{lij} z_{lijs} \qquad l = 1,...,L \quad t = 1,...,T \tag{7.3}$$

This constraint (7.3) is the capacity constraint. The capacity is reduced by the setup times.

$$x_{ljs} \leq \frac{K_{lt}}{a_{lj}} y_{ljs} \qquad l = 1,...,L \quad j = 1,...,J \quad s = 1,...,S \tag{7.4}$$

This restriction (7.4) tells that the production can only take place when the line is setup for the product.

$$x_{ljs} \geq m_{lj}(y_{ljs} - y_{lj(s-1)}) \qquad l = 1,...,L \quad j = 1,...,J \quad s = 1,...,S \tag{7.5}$$

The constraint (7.5) shows the usage of minimum lotsizes.

$$\sum_{j} y_{ljs} = 1 \qquad l = 1,...,L \quad s = 1,...,S \tag{7.6}$$

The equation (7.6) means that only one setup can be per line and micro-period.

$$z_{lijs} \geq y_{lj(s-1)} + y_{ljs} - 1 \qquad l = 1,...,L \quad j = 1,...,J$$
$$i = 1,...,I \quad s = 1,...,S \tag{7.7}$$

The restriction (7.7) points out the connection of setup state and changeover.

[87] Meyr, 2002, p. 280

$$y_{ljs} \in \{0,1\} \qquad l = 1,...,L \quad j = 1,...,J \quad s = 1,...,S \qquad (7.8)$$

This decision variable (7.8) is binary.

$$I_{jt}, x_{ljs}, z_{lijs} \geq 0 \qquad l = 1,...,L \quad j = 1,...,J$$

$$i = 1,...,I \quad s = 1,...,S \qquad (7.9)$$

Here the non-negativity conditions (7.9) are given.

The GLSPPL is a model for simultaneous lotsizing and scheduling. The machine assignment has to be done simultaneously, too. The model can be solved by local search heuristics, like simulated annealing or threshold accepting, in combination with dual reoptimization[88].

7.3 Multi-level, multiple machine example

An example for the multi-level and multi machine LSP is given by[89]:

There are two machines M = 2 and the number of items is J = 4. The time scale has T = 10 periods and the capacities of the machines in period t is C_{1t} = C_{2t} = 15. Figure 17 shows the gozintho-structure of the items.

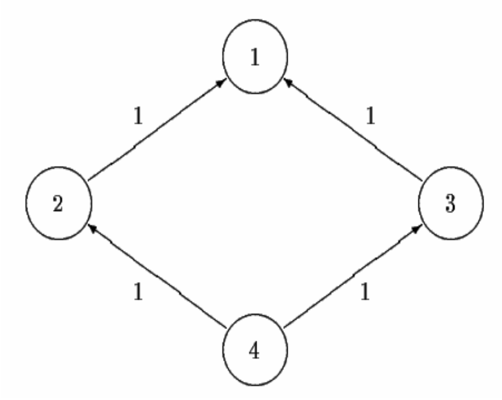


Figure 17: Gozintho-structure[90]

The structure gives information about the successors and predecessors. Item 4 is the predecessor for item 2 and 3, which are the predecessor for item 1.

[88] Meyr, 2002, p. 291

[89] Kimms, http://citeseer.ist.psu.edu/kimms96genetic.html, 10.05.2007

[90] see footnote 89

The demands d_{jt}, capacity needs p_j, lead times v_j and setup state y_{jo} are given in table 9 below.

t	1	2	3	4	5	6	7	8	9	10	p_j	v_j	y_{jo}
d_{1t}							20			20	1	1	1
d_{2t}											1	1	1
d_{3t}											1	1	0
d_{4t}								5			1	1	0

Table 9: Data for the multi-level, multi-machine LSP example[91]

The following Gantt chart (see figure 18) shows the schedules of the solution for the example above:

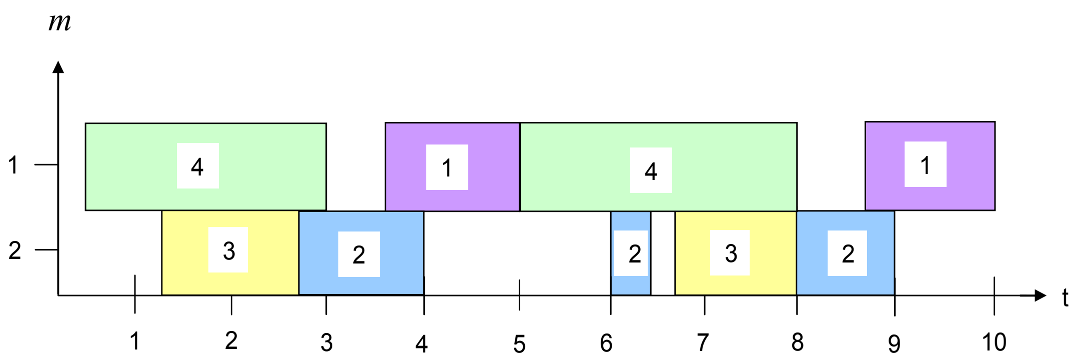


Figure 18: Gantt chart for the multi-level, multi-machine LSP example[92]

On machine m = 1 item 1 and 4 are scheduled and on m = 2 item 2 and 3. First item 4 has to be produced so that item 2 and 3 can be produced. After that item 1 can be produced.

[91] see footnote 89

[92] see footnote 89

8 Hierarchical Integration of Lotsizing and Scheduling

In the hierarchical integration lotsizing and scheduling are handled simultaneously.

1992 Lasserre integrated lotsizing and scheduling in a job shop problem. The model consists of two levels, the lotsizing level and the scheduling level. The lotsizing level is given with a fixed sequence of the jobs on the machines and the scheduling level is given with fixed lotsizes. The procedure starts with an initial production plan in the scheduling level. There with given lotsizes a new sequence is terminated. To solve the problem the procedure alternates between these levels, to find the best feasible plan. This procedure generates at any time a feasible plan, with at least one feasible schedule for the production plan. The ideal procedure, where both levels are solved optimal, is the local optimal solution of the model[93].

An efficient heuristic to solve the scheduling level is the modified shifting bottleneck procedure (MSB) from Dauzere-Peres and Lasserre (1993).

The MSB is a modification of the shifting bottleneck procedure (SB) introduced by Adams et al. in 1988. The MSB is an iterative procedure where for every machine the sequence of a one-machine scheduling (OMS) problem is solved. OMS has to minimize the makespan of the jobs that have to be produced on a machine. So in each step a one-machine scheduling problem is looked at for each machine which was not scheduled yet. To minimize the makespan, which is the objective, the bottleneck machine should be scheduled first. The bottleneck machine is the machine with the largest minimum makespan. The MSB stops when all machines are scheduled[94].

[93] Lasserre, 1992, p. 1201

[94] Dauzere-Peres, Lasserre, 1994, p.24

9 Comparison of the different modelling approaches

This chapter shows a summary of the lotsizing and scheduling problems mentioned in this work.

9.1 Single-Level Single-Item Uncapacitated Models

The model conditions are:

model conditions	**demand**	**planning horizon**
EOQ	constant	infinite
WW	dynamic	finite

Table 10: Single-level single-item uncapacitated model conditions

The Wagner-Within Model is an extension of the Economic Order Quantity Model with dynamic demand.

The model complexity is shown in the following table:

model complexity	**# non binary/ # binary**	**non binary variables**
EOQ	3/0	demand, setup costs, inventory costs
WW	2*T/0	production quantity, inventory at end of period

Table 11: Single-level single-item uncapacitated model complexity

Both models have no capacity constraint, they are so called uncapacitated models.
The models lead to an optimal solution. There are no complex algorithms needed.

9.2 Single-Level Multi-Item Capacitated Models

The model conditions are:

model conditions	demand	planning horizon	large or small bucket
CLSP	dynamic	finite	large bucket
CLSPL			combination of small and large bucket
DLSP			small bucket
CSLP			small bucket
PLSP			small bucket
GLSP			large bucket

Table 12: Single-level multi-item capacitated model conditions

The large bucket problems, the Capacitated Lotsizing Problem and the General Lotsizing and Scheduling Problem, as also the small bucket problems, the Discrete Lotsizing and Scheduling Problem, the Continuous Setup Lotsizing Problem and the Proportional Lotsizing and Scheduling Problem, have dynamic demand and a finite planning horizon. The CLSPL has also dynamic demand and a finite horizon, but is a combination of the small and large bucket problem.

model complexity	base	objective function	# non binary/ # binary	non binary variables	binary variables	comments
CLSP		$Min \sum_{j=1}^{J}\sum_{t=1}^{T}(s_j x_{jt} + h_j I_{jt})$	2*J*T/1*J*T	• inventory • production quantity	• setup variable	
CLSPL	CLSP	$Min \sum_{j=1}^{J}\sum_{t=1}^{T}\left[s_j\left(x_{jt} - z_{jt}\right) + h_j I_{jt}\right]$	2*J*T+1*T/ 2*J*T	• inventory • production quantity • variable if item is produced or not	• setup variable • variable if item is linked with next period or not	• 3 constraints: only one product can be produced at the end of a period and so be linked to the following period • 2 constraints when more than one item produced the setup can be either linked with the previous or the following period setup
DLSP	CLSP	$Min \sum_{j=1}^{J}\sum_{t=1}^{T}(s_j x_{jt} + h_j I_{jt})$	2*J*T/2*J*T	• inventory • production quantity	• setup variable • machine setup	• capacity constraint (all-or nothing-assumption)
CSLP	DLSP	$Min \sum_{j=1}^{J}\sum_{t=1}^{T}(s_j x_{jt} + h_j I_{jt})$	2*J*T/2*J*T	• inventory • production quantity	• setup variable • machine setup	• capacity constraint (all-or nothing-assumption given up)

PLSP	DLSP	$Min \sum_{j=1}^{J}\sum_{t=1}^{T}(s_j x_{jt} + h_j I_{jt})$	2*J*T/2*J*T	• inventory • production quantity	• setup variable • machine setup	• capacity constraint (setup at beginning or end of period) • capacity splitting
GLSP	DLSP	$Min \sum_{j=1}^{J}\sum_{n=1}^{N} s_j x_{jn} + \sum_{j=1}^{J}\sum_{t=1}^{T} h_j I_{jt}$	1*J*T+ 1*J*N/ 2*J*N	• inventory • production quantity	• setup variable • machine setup	• constraint: inventory (item can be produced at several places) • constraint for unique setup • constraint: position where setup takes place

Table 13: Single-level multi-item capacitated model complexity

The six models have inventory and production quantity as numeric decision variables and the setup state as binary decision variable. The objective function of the models is the same with the purpose to minimize the total costs. The difference between the models arises from the constraints.

The CLSPL differs from the CLSP by the variable z_{jt}, which tells if the setup of a period is linked with the following period or not, and the constraints (5.12) – (5.16) that make linking of lots possible.

The difference between the CLSP and the DLSP is the decision variable y_{jt}, the machine setup state. In the CLSP only one item can be produced per period. That is the same for the DLSP only with the all-or-nothing-assumption, what means that the item has to be produced over the full period or not at all. This is shown through the constraint (5.21) $p_j q_{jt} = C_t y_{jt}$.

The CSLP is based on the DLSP. The differential is that the item does not anymore have to be produced over the full period, shown by the constraint (5.28) $p_j q_{jt} \leq C_t y_{jt}$. The remaining capacity stays then unused.

For all three models the setup state only can occur at the beginning of the period and the setup state is lost over idle periods. That is the difference to the PLSP. There the setup can occur at the beginning or at the end of the period. That is illustrated by the constraint (5.35) $p_j q_{jt} \leq C_t (y_{j(t-1)} + y_{jt})$. Another difference is that two items can be produced per period and the setup state can be preserved over idle periods.

The GLSP has a restricted number of lots per period. For every lot a setup is possible, what means that more than one setup per period is allowed.

The model solvability is:

model solvability	optimal solvability	solution methods
CLSP	NP-hard	heuristics[95]
CLSPL		Backward Add[96], tabu search[97]
DLSP		Lagrangean Relaxation[98]
CSLP		Lagrangean Relaxation in combination with dynamic programming[99]
PLSP		Backward Add[100]
GLSP		local search based on threshold accepting[101]

Table 14: Single-level multi-item capacitated model solvability

All models are NP-hard to solve optimal. There are different solution methods for the models.

[95] Maes, Van Wassenhove, 1988, p. 993

[96] Suerie, Stadler, 2003, p. 1040

[97] Suerie, Stadler, 2003, p. 1041

[98] Fleischmann, 1990. p. 340

[99] Salomon, 1991, p. 35

[100] Drexl, Haase, 1995, p. 76

[101] Fleischmann, Meyr, 1997, p. 11

10 Solving Methods for Lotsizing and Scheduling Problems

This chapter gives a summary of the most common solving methods for the lotsizing and scheduling problems (in alphabetical order).

Backward Add Method (BACKADD)[102]:

The backward add method is a method where lotsizing and scheduling is done chronologically backwards. There are three steps. In the first step lotsizing and scheduling is done from the last period to the first. So the first period looked at is $t = T$, the second is $t = T - 1$ and so on until the last one which is $t = 1$. In this model at most two items can be added in every period, which is looked at the second step. Therefore regrets or priority rules are used to decide in which period which item is going to be produced. This is the third step of the backward add method. This method tries to find one feasible solution.

Lagrangean Relaxation[103]:

Relaxation is a technique for transforming hard constraints into easier ones. That means that constraints of the optimization model are either substituted by a more easily handled constraint or dropped completely to obtain an easier solvable model as the original one.

The Lagrangean Relaxation is a relaxation technique which works by moving capacity constraints with penalty costs, the so called lagrangean multipliers, for their violation into the objective function to become an easier problem to solve.

The Lagrangean Relaxation is s technique for finding lower bounds. It has to be extended with other heuristics or integrated in a branch and bound model to find upper bounds and through this a feasible solution[104].

It is used for single-item models and cannot be directly extended to multi-item models. For the standard single-item models the Lagrangean Relaxation outperforms metaheuristics[105].

[102] Drexl, Haase, 1995, p. 76

[103] Domschke, Scholl, Voß, 1993, p. 136

[104] Jans, Degraeve, 2007, p.1863

[105] Jans, Degraeve, 2007, p.1866-1868

Simulated Annealing (SA)[106]:

Simulated annealing is a technique belonging to the meta-heuristics.

It is a "search procedure that generalizes iterative improvement approaches to combinatorial optimization by sometimes accepting transitions to lower quality solutions to avoid getting trapped in local minima"[107].

SA gives optimal or near-optimal solutions for combinatorial optimizations, but often with intensive computational effort. It is flexible with respect to the evolution of problems and easy to implement. It gives very good results for problems of big size[108].

SA starts with generating an initial random or heuristic solution and initializing the temperature parameter *T*. This parameter controls the probability of accepting a movement to a solution with worse quality. The current solution is replaced by moving to random nearby solutions, while remembering the best solution found so far, until the termination condition is satisfied.

Termination conditions can be for example:

- the achievement of the end-temperature,
- the exceedance of the maximal number of iterations,
- that during a number of iterations no better solution is found,
- or that during the temperature reduction no better solution is found.

The temperature decreases during the search process. The motivation for lowering the temperature is to gain convergence. So at the beginning of the search the probability of accepting moves which result in solutions of worse quality than the current solution, is high and then decreases. The allowance of making moves to solutions of worse quality saves from becoming stuck in a local minimum. When the parameter temperature is below a given level the simulated annealing procedure stops.

An overview of the simulated annealing process is shown in figure 19.

---

106 Sadeh et al., p.279

107 Sadeh et al., p.278

108 Dreo et al., 2006, p. 7

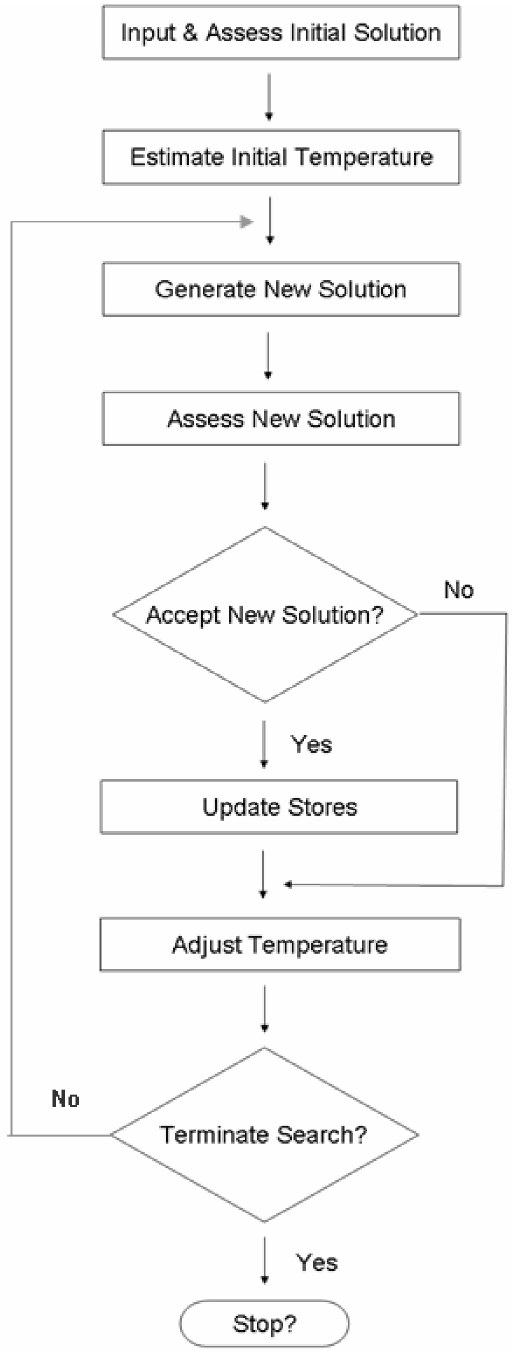


Figure 19: Simulated annealing flowchart[109]

[109] http://www.phy.ornl.gov/csep/CSEP/GIFFIGS/MOF217.GIF , 30.04.2007

Tabu Search (TS)[110]:

Tabu search is an optimization method belonging to the meta-heuristics.

TS is an algorithm based on adaptive memory and learning. It uses a local search or neighborhood search procedure to iteratively move from one solution to the next solution in its neighborhood, until a stopping criterion is satisfied. Stopping criterions can be for example the exceedance of the maximal number of iterations or the succession of the computational time. The already visited solutions are memorized in the tabu list, which is a short-term memory. The list gives the name to tabu search.

The intention of the list is not to prevent a previous move from being repeated, but rather to insure that it is not reversed. That means that no search cycling occurs and so the search cannot get stuck in a local optimum. These moves then have got a tabu status[111].

Other tabu lists prohibit solutions that have certain attributes. When a single attribute is tabu then more than one solution ends up being tabu. Some of these tabu solutions might be of excellent quality or might not have been visited. To avoid this problem an aspiration criteria is introduced. It allows forgetting the tabu state of a solution and then to include it. A common used aspiration criterion is to permit solutions which are better than the currently best solution.

The basic version of the TS can be extended through intensification and diversification. Intensification is the specific search of a defined solution space through holding on to advantageous solution attributes. Diversification is used to search the solution space to find not looked at new spaces[112].

An overview of the process is shown in figure 20.

[110] Vasquez et al., p. 370

[111] Dreo et al., 2006, p. 52

[112] Arnold et al., 2004, p. A2-23

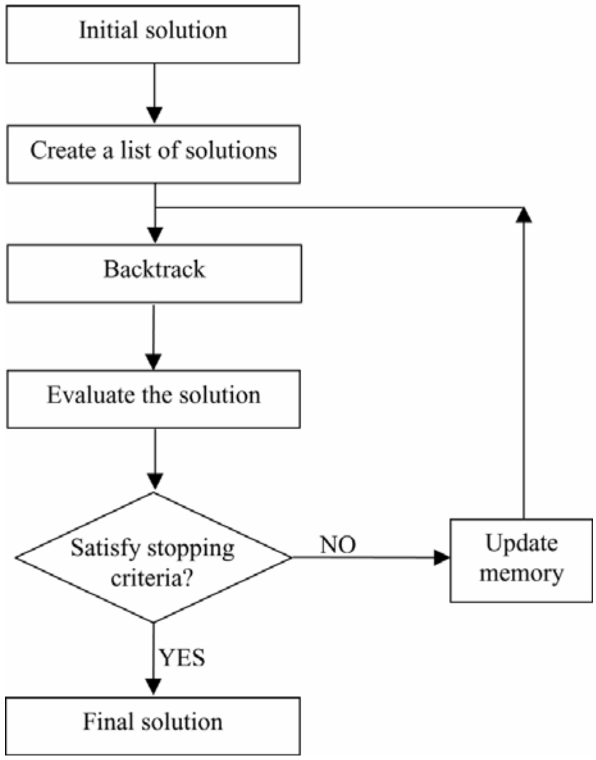


Figure 20: Tabu search flowchart[113]

Threshold Accepting (TA)[114]**:**

Threshold accepting is a modification of simulated annealing. TA also starts from an initial solution and iteratively moves then from one solution to the next solution in its neighborhood. The difference between simulated annealing and threshold accepting lies in the acceptance criterion of the solutions. TA always allows solutions of worse quality than the current solution to overcome local optima, as long the decline is not bigger than a given threshold. The threshold also decreases, like the temperature in the simulated annealing procedure, during the search process.

Metaheuristics are flexible schemes that can handle large and complex problems. They are used for extensions of the standard models (like multi-item models or sequence dependent models), where traditional algorithms fail or do not exist. However for the standard models they are outperformed[115].

---

[113] http://www.emeraldinsight.com/fig/0330210305005.png , 30.04.2007

[114] Dreo et al., 2006, p. 157

[115] Jans, Degraeve, 2007, p.1868

11 Summary

In this diploma thesis the current lotsizing and scheduling problems were presented. First of all the single-level uncapacitated models were introduced. These are the economic order quantity model with constant demand and the Wagner-Within model with dynamic demand. In the next chapter the single-level capacitated models were described. The first one was the Capacitated Lotsizing Problem (CLSP) which is a model with only one item production per period, and the setup can only occur at the beginning of a period or not at all. The next one was the Capacitated Lotsizing Problem with Linked Lotsizes (CLSPL). There a setup can be carried-over from one period to the next one. Then the Discrete Lotsizing and Scheduling Problem (DLSP) was shown. That model has a all-or-nothing-assumption. This assumption means that an item can only be produced over a full period or not at all. A similar model to the DLSP was the Continuous Setup Lotsizing Problem (CSLP). In this model the all-or-nothing-assumption was given up, what means that an item has not to be produced over a full period anymore. The next model described was the Economic Lotsizing and Scheduling Problem (ELSP), which is a model that shall find cyclical production schedules. In the model no two items can be produced at the same time. Then the Proportional Lotsizing and Scheduling Problem (PLSP), where the setup can take place at the beginning or the end of a period and capacity splitting is possible, and the General Lotsizing and Scheduling Problem (GLSP), which is a model were the periods are divided into sub-periods and because of that more than one setup per period is allowed, followed. Also some variants were looked at. These were the GLSP – Conservation of Setup State, the GLSP – Loss of Setup State and the GLSP with Setup Times. The last single-level capacitated model was the Batching and Scheduling Problem (BSP), where the demands for the same item can be put together to build a lot and so save setup costs. In the next chapter Multi-Level Lotsizing and Scheduling Problems (MLLSP) were introduced. The MLLSP-structures and an example, the MLPLSP, were shown. Lotsizing and Scheduling Problems with Multiple Machines Problems (LSPMM) were described in the following chapter, consisting of the multi machine types and an example, the GLSPPL. The last model chapter was the hierarchical integration of lotsizing and scheduling, where lotsizing and scheduling were handled simultaneously. Afterwards a comparison of selected problems was shown through model conditions, model complexity and model solvability. At last the definitions of the solving methods for the lotsizing and scheduling problems were given.

Lotsizing and scheduling problems are a challenging issue, because many are very difficult to solve optimal. The most of them are NP-hard, and can only be solved by difficult heuristics or metaheuristics. Here an optimal solution procedure cannot be guaranteed, but an adequate one in reasonable time. In the last years advanced planning systems (APS)[116] are used in operational practice for computer based support for the optimization problems. APS uses therefore mathematical models like linear programming (LP) or mixed-integer programming (MIP) models. These models are not always optimal solvable with Standard-Solvers, but can be used as source for heuristics.

---

[116] Meyr, 2004, p. 586

12 Appendix

Here in addition to chapter 9.2 the model complexity table with mathematical notations for the single-level multi-item capacitated models is shown:

model complexity	base	objective function	decision variables	binary variables	≥ 0	constraints	meaning of constraints
CLSP		$Min \sum_{j=1}^{J}\sum_{t=1}^{T}(s_j x_{jt} + h_j I_{jt})$	I_{jt}, q_{jt}, x_{jt}	x_{jt}	I_{jt}, q_{jt}, h_j	(1) $I_{jt} = I_{j(t-1)} + q_{jt} - d_{jt}$ (2) $p_j q_{jt} \le C_t x_{jt}$ (3) $\sum_{j=1}^{J} p_j q_{jt} \le C_t$	(1) inventory (2) item can be only produced when machine is setup (3) capacity constraint
CLSPL	CLSP	Min $\sum_{j=1}^{J}\sum_{t=1}^{T}\left[s_j\left(x_{jt} - z_{jt}\right) + h_j I_{jt}\right]$	I_{jt}, q_{jt}, x_{jt}, v_t, z_{jt}	x_{jt}, z_{jt}	I_{jt}, q_{jt}, h_j, v_t	(1) $I_{jt} = I_{j(t-1)} + q_{jt} - d_{jt}$ (2) $p_j q_{jt} \le C_t x_{jt}$ (3) $\sum_{j=1}^{J} p_j q_{jt} \le C_t$ (4) $\sum_{j=1}^{J} z_{jt} \le 1$ (5) $z_{jt} - x_{j,t-1} \le 0$	(1) inventory (2) item can be only produced when machine is setup (3) capacity constraint (4)-(6) only one product can be produced at the end of a period and so be linked to the following period

						(6) $z_{jt} - x_{jt} \leq 0$ (7) $1 - \sum_{j=1}^{J} x_{jt} + J * v_t \geq 0$ (8) $z_{jt} + z_{j,t-1} + v_{t-1} \leq 2$	(7)+(8) when more than one item produced the setup can be either linked with the previous or the following period setup
DLSP	CLSP	$Min \sum_{j=1}^{J}\sum_{t=1}^{T}(s_j x_{jt} + h_j I_{jt})$	I_{jt}, q_{jt}, x_{jt}, y_{jt}	x_{jt}, y_{jt}	I_{jt}, q_{jt}, x_{jt}, h_j	(1) $I_{jt} = I_{j(t-1)} + q_{jt} - d_{jt}$ (2) $\sum_{j=1}^{J} y_{jt} \leq 1$ (3) $p_j q_{jt} = C_t y_{jt}$ (4) $x_{jt} \geq y_{jt} - y_{j(t-1)}$	(1) inventory (2) only one item per period (3) capacity constraint (all-or nothing-assumption) (4) beginning of new lot
PLSP	DLSP	$Min \sum_{j=1}^{J}\sum_{t=1}^{T}(s_j x_{jt} + h_j I_{jt})$	I_{jt}, q_{jt}, x_{jt}, y_{jt}	x_{jt}, y_{jt}	I_{jt}, q_{jt}, x_{jt}, h_j	(1) $I_{jt} = I_{j(t-1)} + q_{jt} - d_{jt}$ (2) $\sum_{j=1}^{J} y_{jt} \leq 1$ (3) $p_j q_{jt} \leq C_t (y_{j(t-1)} + y_{jt})$ (4) $\sum_{j=1}^{J} p_j q_{jt} \leq C_t$ (5) $x_{jt} \geq y_{jt} - y_{j(t-1)}$	(1) inventory (2) only one item per period (3) capacity constraint (setup at beginning or end of period) (4) capacity splitting (5) beginning of new lot

GLSP	DLSP	Min $\sum_{j=1}^{J}\sum_{n=1}^{N} s_j x_{jn} + \sum_{j=1}^{J}\sum_{t=1}^{T} h_j I_{jt}$	I_{jt}, q_{jn}, x_{jn}, y_{jn}	x_{jn}, y_{jn}	I_{jt}, q_{jn}, h_j, N_t	(1) $I_{jt} = I_{j(t-1)} + \sum_{n=F_t}^{L_t} q_{jn} - d_{jt}$ (2) $p_j q_{jn} \le C_t y_{jn}$ (3) $\sum_{j=1}^{J}\sum_{n=F_t}^{L_t} p_j q_{jn} \le C_t$ (4) $\sum_{j=1}^{J} y_{jn} \le 1$ (5) $x_{jn} \ge y_{jn} - y_{j(n-1)}$	(1) inventory (item can be produced at several places) (2) production only when setup for item (3) capacity constraint (4) unique setup (5) position where setup takes place

Table 15: Single-level multi-item capacitated model complexity (mathematical notation)

VI. References

1. Literature in alphabetical order:

- Adams, Joseph, Balas Egon and Zawack Daniel, *The Shifting Bottleneck Procedure for Job Shop Scheduling*, Management Science, Vol. 34, No. 3, 1988, 391-401
- Arnold, D., Isermann, H., Kuhn, A. and Tempelmeier, H., *Handbuch Logistik*, Springer, Berlin, 2004
- Beamon, Benita M. and Chen, Victoria C.P., *Performance Analysis of Conjoined Supply Chains*, International Journal of Production Research Vol. 39, No. 14, 2001, 3195-3218
- Brahimi, Nadjib, Dauzere-Peres, Stephane, Najid, Najib M. and Nordli, Atle, *Single item lot sizing problems*, European Journal of Operational Research Vol. 168, 2006, 1-16
- Brucker, Peter, *Scheduling Algorithms*, Springer, Berlin et al., 1995
- Carreno, Jose Juan, *Economic lot scheduling for multiple products on parallel identical processors*, Management Science, Vol. 36, No. 3, 1990, 348-358
- Dauzere-Peres, Stephane and Lasserre, Jean-Bernard, *An Integrated Approach in Production Planning and Scheduling*, Springer, Berlin et al., 1994
- Domschke, Wolfgang, Scholl, Armin and Voß, Stefan, *Produktionsplanung: Ablauforganisatorische Aspekte*, Springer, Berlin et al., 1993
- Dréo, Johann, Petrowski, Alain, Siarry, Patrick and Taillard, Eric, *Metaheuristics for Hard Optimization: Methods and Case Studies*, Springer, Berlin et al., 2006
- Drexl, Andreas and Haase, Knut, *Proportional lotsizing and scheduling*, International Journal of Production Economics, Vol. 40, 1995, 73-87
- Drexl, Andreas and Kimms, A., *Lot sizing and scheduling – Survey and extensions*, European Journal of Operational Research Vol. 99, 1997, 221-235

- Elmaghraby, Salah E., *The Economic Lot Scheduling Problem (ELSP): Review and Extensions*, Management Science, Vol. 24, No. 6, 1978, 587-598
- Fleischmann, Bernhard, *The discrete lot-sizing and scheduling problem*, European Journal of Operational Research 44, 1990, 337–348
- Fleischmann, Bernhard and Meyr, Herbert, *The general lotsizing and scheduling problem*, OR Spektrum , Vol. 19, 1997, 11-21
- Haase, Knut, *Lotsizing and scheduling for production planning,* Springer, Berlin et al., 1994
- Harris, Ford Whitman, *How Many Parts to Make at Once*, Operations Research Vol. 38, No. 6, 1990, 947-950, reprint of Factory: The Magazine of Management Vol. 10, 1913, 135-136
- Hsu, Wen-Lian, *On the general feasibility test of scheduling lot sizes for several products on one machine*, Management Science, Vol. 29, No. 1, 1983, 93-105
- Jans, Raf, Degraeve, Zeger, *Meta-heuristics for dynamic lot sizing: A review and comparison of solution approaches*, European Journal of Operational Research Vol. 177, 2007, 1855-1875
- Karimi, B., Fatemi Ghomi, S.M.T. and Wilson, J.M., *The capacitated lotsizing problem: a review of models and algorithms,* Omega, Vol. 31, Nr. 5, 2003, 365-378
- Lasserre, Jean B., *An integrated model for job-shop planning and scheduling*, Management Science, Vol. 38, No. 8, 1992, 1201–1211.
- Maes, Johan and Van Wassenhove, Luk, *Multi-Item Single-Level Capacitated Dynamic Lot-Sizing Heuristics: A General Review,* Journal of the Operational Research Society, Vol. 39, No. 11, 1988, 991-1004
- Meyr, Herbert, *Simultaneous lotsizing and scheduling by combining local search with dual reoptimization*, European Journal of Operational Research, Vol. 120, 2000, 311-326
- Meyr, Herbert, *Simultaneous lotsizing and scheduling on parallel machines*, European Journal of Operational Research, Vol. 139, 2002, 277-292
- Meyr, Herbert, *Simultane Losgößen- und Reihefolgeplanung bei mehrstufiger kontinuierlicher Fertigung*, Zeitschrift für Betriebswirtschaft, Ausgabe 6, 2004, 585-610

- Quadt, Daniel, *Lot-Sizing and Scheduling for Flexible Flow Lines*, Springer, Berlin, 2004
- Sadeh, Norman, Nakakuki, Yoichiro and Thangiah, Sam, *Learning to Recognize (Un)Promising Simulated Annealing Runs: Efficient Search Procedures for Job Shop Scheduling and Vehicle Routing,* in: Osman, Ibrahim and Kelly, James, *Meta-Heuristics: Theory & Applications*, Kluwer Academic Publishers, Boston et al., 1996
- Salomon, Marc, Deterministic *lotsizing models for production planning*, Springer, Berlin et al., 1991
- Salomon, Marc, Kroon, Leo G., Kuik, Roelof and Van Wassenhove, Luk N., *Some extensions of the discrete lotsizing and scheduling problem*, Management Science, Vol. 37, No. 7, 1991, 801-812
- Skriptum: Grundzüge der BWL IV: Produktion und Logistik, O. Prof. Dr. Richard F. Hartl, Universität Wien, WS 1997/98
- Suerie, Christopher and Stadtler, Hartmut, *The Capacitated Lot-Sizing Problem with Linked Lot Sizes*, Management Science, Vol. 49 Nr. 8, 2003,1039-1054
- Vasquez, Michel, Dupont, Audrey and Habet, Djamal, *Consistent Neighbourhood in a Tabu Search*, in: Ibaraki, Toshihide, Nonobe, Koji and Yagiura, Mutsunori, *Metaheuristics: Progress as Real Problem Solvers*, Springer, New York, 2005

2. Internet:

- http://en.wikipedia.org/wiki/EOQ, 06.08.2006
- http://www.cms.uwe.ac.uk/~arclark/ifors.pdf, 05.09.2006,
  Staggemeier, Andrea and Clark, Alistair R., *A survey of lotsizing and scheduling models*, presented at the 23rd Annual Symposium of the Brazilian Operational Research Society, November 2001
- http://www.phy.ornl.gov/csep/CSEP/GIFFIGS/MOF217.GIF , 30.04.2007
- http://www.emeraldinsight.com/fig/0330210305005.png , 30.04.2007
- http://www.mathematik-21.de/statussem/poster/quadt/quadt.ppt#258,6,Lot Sizing and Scheduling-Problem , 08.05.2007
- http://citeseer.ist.psu.edu/kimms96genetic.html, 10.05.2007
  Kimms, Alf, *A Genetic Algorithm for Multi-Level, Multi-Machine Lot Sizing and Scheduling*, No. 415, 1996
- http://riot.ieor.berkeley.edu/~vinhun/machine.html, 20.05.2007

Abstract

Diese Magisterarbeit gibt einen Überblick über Losgrößen- und Reihefolgeplanungsprobleme. Losgrößenplanung und Reihefolgeplanung sind integrale Bestandteile der Produktionsplanung. Sie geben an wie viele Produkte und in welcher Reihenfolge diese produziert werden sollen, um die Gesamtkosten zu minimieren.

Der Hauptteil der Arbeit beschäftigt sich mit der Darstellung der unterschiedlichen Problemverfahren. Diese werden anhand von Definitionen, mathematischen Formulierungen und Beispielen dargestellt.

Zu Beginn der Arbeit wird das Produktionsplanungssystem (PPS) erläutert (in Kapitel 1.1). Dann werden die Definitionen von Losgrößenplanung und Reihefolgeplanung und ihr Zusammenhang erklärt (in Kapitel 2). Darauf folgen eine allgemeine Problembeschreibung und die verschiedenen Kriterien für Losgrößen- und Reihefolgeplanungsprobleme (in Kapitel 3).

Im Hauptteil werden die verschiedenen Problemtypen aufgelistet. Es werden einstufige unkapazitierte (in Kapitel 4) und einstufig kapazitierte Probleme (in Kapitel 5), genauso wie mehrstufige Probleme (in Kapitel 6) und Probleme auf mehreren Maschinen (in Kapitel 7) erklärt. Ebenfalls wird die hierarchische Integration von Losgrößen- und Reihefolgeplanungsproblemen beschrieben (in Kapitel 8).

Zuletzt werden ein Vergleich der unterschiedlichen Verfahren dargestellt (in Kapitel 9) und die möglichen Lösungsverfahren (in Kapitel 10) behandelt.

Curriculum vitae

Name:	Sally Tischer
Geboren:	5.11.1978, in Sydney, Australien
Staatsbürgerschaft:	Österreich
Familienstand:	ledig

Schulbildung:

1985-1989	Volksschule, Wien
1989-1997	Realgymnasium, Wien, 1997 Matura (mit gutem Erfolg abgeschlossen)

Studium:

1997-2007	Studium BWL Universität Wien (BWZ)
Diplom:	2004 Bakkalaureat der BWL (rer. soc. oec.) mit der Vertiefung Management
KFK:	KFK Produktionsmanagement am Lehrstuhl für Produktion und Logistik

Beruflicher Werdegang:

Juli 1999 + Juli 2000	Schindler AG, Wien Ferialtätigkeit in der Exportabteilung
1999–2001	Wiener Sozialdienste, Wien geringfügige Mitarbeiterin in der Personalabteilung
2003–2006	Wiener Sozialdienste, Wien Kfm. Mitarbeiterin in der Personalabteilung
2005-	Green Deluxe Wolfgang Axamit KEG (klassische Assistenzaufgaben, Buchhaltung und Finanz)