ECONSTOR
Make Your Publications Visible.

A Service of
ZBW
Leibniz-Informationszentrum Wirtschaft
Leibniz Information Centre for Economics

Haase, Knut

Working Paper — Digitized Version

Capacitated lot-sizing with sequence dependent setup costs

Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel, No. 340

Provided in Cooperation with:
Christian-Albrechts-University of Kiel, Institute of Business Administration

Suggested Citation: Haase, Knut (1994) : Capacitated lot-sizing with sequence dependent setup costs, Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel, No. 340, Universität Kiel, Institut für Betriebswirtschaftslehre, Kiel

This Version is available at:
https://hdl.handle.net/10419/155415

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.

WWW.ECONSTOR.EU

Mitglied der Leibniz-Gemeinschaft

No. 340

Capacitated Lot-Sizing with

Sequence Dependent Setup Costs

Knut Haase

February 94

Knut Haase, Institut für Betriebswirtschaftslehre, Christian-Albrechts-Universität zu Kiel,
Olshausenstr. 40, D-24118 Kiel

Abstract: A new model is presented for capacitated lot-sizing with sequence dependent setup costs. The model is solved heuristically with a backward oriented method; the sequence and lot-size decisions are based on a priority rule which consists of a convex combination of setup and holding costs. A computational study is performed where the heuristic is compared with the Fleischmann approach for the discrete lot-sizing and scheduling problem with sequence dependent setup costs.

Keywords: Lot-sizing, scheduling, sequence dependent setup costs, production planning and control

1. Introduction

For many production facilities the expenditures for the setups of a machine depend on the sequence in which different items are scheduled on the machine. Especially, if a machine produces items of different family types setup between items of different families can be substantially more costly than setups between items of the same family. Typically, the production quantities of the items are computed via lot-sizing. The objective of lot-sizing is to determine a schedule such that the sum of setup and holding costs are minimized with respect to demand (and capacity) constraints. Thus in the case of sequence dependent setup costs we have to integrate sequencing in lot-sizing in order to compute setup costs accurately.

Despite of the relevance there has been done only little research in the area of lot-sizing and scheduling with sequence dependent setup costs.

Some papers have been published which are related to the so-called discrete lot-sizing and scheduling problem (DLSP) [cf. Fleischmann 1990]. In the DLSP it is assumed that the production process always runs full periods without changeover, i.e. at most one item will be produced per period. First, in [Schrage 1982] a DLSP-like model with sequence dependent setup costs is considered. For the DLSP an exact branch-and-bound approach based on Lagrangean relaxation of the capacity constraints has been presented in [Fleischmann 1990] which is extended to the DLSP with sequence dependent setup costs (DLSPSD) in [Fleischmann, Popp 1989]. In [Fleischmann 1994] the DLSPSD is transformed to a travelling salesman problem with time windows; methods to determine a lower bound and a heuristic solution for the DLSPSD are presented.

Recently, a new type of model has been published in [Haase 1994] which is called the proportional lot-sizing and scheduling problem (PLSP). The PLSP is based on the assumption that at most one setup can occur within a period, i.e. at most two items are producible per period which allow a more efficient capacity usage than in the DLSP. Furthermore the setup state can be preserved over idle time. For the heuristic solution a method is introduced which can be extended for the PLSP with sequence dependent setup costs.

In [Dilts, Ramsing 1989] an uncapacitated lot-sizing model with sequence dependent setup costs is considered.

A heuristic for a static (i.e. constant demand per period) lot scheduling problem with sequence dependent setup costs and times is introduced in [Dobson 1992].

Thus, the capacitated, dynamic lot-sizing problem with sequence dependent setup costs where per period all items are producible has not been considered so far. In this paper this problem will be addressed.

We start in the next section with a mathematical formulation of the DLSPSD. Then we extend the so called capacitated lot-sizing problem with linking (CLSPL) [cf. Dillenberger et al. 1992, Haase 1994] for the case with sequence dependent setup costs, denoted by CLSPLSD. Linking means that production quantities of an item of adjacent periods can be linked to avoid a setup. As in the PLSP the setup state can be preserved over idle time. To illustrate the differences between the DLSPSD and the CLSPLSD we consider a simple example. For the heuristic solution we introduce a method which is backward oriented and relies on a priority rule (Section 3). A CLSPLSD can be approximated by a DLSPSD. A computational study is performed in Section 4 where we compare the method with the DLSPSD approach proposed in [Fleischmann 1994].

2. Capacitated Lot-sizing with Sequence Dependent Setup-Costs

We characterize the deterministic lot-sizing problem which is addressed: A single-stage system is considered, where a number of different items $j=1,\dots,J$ have to be manufactured on one machine (corresponding to a single capacity constraint) which is available with C_t capacity units. The time horizon T is segmented into a finite number of periods $t=1,\dots,T$. Producing one unit of item j absorbs p_j capacity units. The demand for item j in period t, d_{jt}, has to be satisfied without delay (shortages are disallowed). Setup costs of sc_{ij} are incurred when the setup of the machine changes from item i to item j. Inventory costs h_j per unit (holding costs coefficient) are incurred for the inventory of item j at the end of a period. The objective is to minimize the sum of setup and holding costs.

[Fleischmann, Popp 1989] and [Fleischmann 1994] propose the discrete lot-sizing and scheduling problem with sequence dependent setup costs (DLSPSD) where a period t is divided in S_t sub-periods $s=(t,1), (t,2), \dots, (t,S_t)$ of equal length. The following fundamental assumption is made:

The production process always runs full sub-periods without changeover.

Thus, at most one item is producible in a sub-period s and at most S_t different items in a period t, respectively. Since the sub-periods have the same length (capacity) lot-sizes are always multiples of a full period production $\tilde{p}_j$.

We use the following notation:

indices

j = item number; j=0,...,J where J denotes the number of items, and j=0 the "idle-item" or the non-setup state.

t = period number; t=1,...,T where T is the number of periods (planning horizon).

s = sub-period; s=(1,1),...,(1,S_t),...,(t,k),...,(T,1),...,(T,S_T) where (t,k) denotes the k-th sub-period in period t and S_t the number of sub-periods of period t.

data

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

h_j = inventory holding costs per unit of item j

$\tilde{p}_j$ = production quantity per sub-period of item j

c_{ij} = setup costs which are incurred when the setup changes from item i to item j

decision variables

I_{jt} = inventory of item j in period t (I_{jt} >0).

x_{ijs} = a continuous variable which indicates the change of the setup state. If $x_{ijs} \geq 1$, then item j is setup after item i in sub-period s.

y_{js} = a binary variable. If y_{js}=1, then the machine is setup for item j in sub-period s (y_{js}=0 otherwise).

Mathematically the DLSPSD can be stated as follows:

$$\text{minimize } Z_{DLSPSD} = \sum_{i=0}^{J}\sum_{j=0}^{J}\sum_{s=(1,1)}^{(T,S_T)} c_{ij}x_{ijs} + \sum_{t=1}^{T}\sum_{j=1}^{J} h_j I_{jt} \tag{1}$$

subject to

$$I_{j,t-1} + \tilde{p}_j \sum_{s=(t,1)}^{(t,S_t)} y_{js} - d_{jt} = I_{jt} \qquad (j=1,\ldots,J;\ t=1,\ldots,T) \tag{2}$$

$$I_{jT} < \tilde{p}_j \qquad (j=1,\ldots,J) \tag{3}$$

$$\sum_{j=0}^{J} y_{js} = 1 \qquad (s=(1,1),\ldots,(T,S_T)) \tag{4}$$

$$x_{ijs} \geq y_{js} + y_{i,s-1} - 1 \qquad (i=0,\ldots,J;\ j=0,\ldots,J;\ s=(1,1),\ldots,(T,S_T)) \tag{5}$$

$$y_{js} \in \{0,1\} \qquad (j=0,\ldots,J;\ s=(1,1),\ldots,(T,S_T)) \tag{6}$$

$$I_{jt} \geq 0 \qquad (j=1,\ldots,J;\ t=1,\ldots,T) \tag{7}$$

$$x_{ijs} \geq 0 \qquad (i=0,\ldots,J;\ j=1,\ldots,J;\ s=(1,1),\ldots,(T,S_T)) \tag{8}$$

(1) minimizes the sum of setup and holding costs. (2) comprises the ordinary inventory balance constraints. (3) avoids unnecessary production which might occur if the *triangle inequality*

$$c_{ij} \leq c_{ik} + c_{kj} \quad (i,j,k=0,\dots,J) \tag{9}$$

is *not* satisfied. (Note the setup state is not preserved over idle periods. Due to the basic assumption the final inventory of an item may be greater or equal than zero and less than the production quantity per sub-period, i.e. it is $0 \leq I_{jT} < \tilde{p}_j$ for j=1,...,J.) That one item per sub-period is produced only is stated in (4). (5) forces setup costs of item j to be added to the total costs if a batch starts for item j.

Note, if $S_t > 2$ then more than one batch can start for an item j in the period t (e.g. $y_{i(t,1)} = y_{j(t,2)} = y_{i(t,S_t)} = 1$ with $i \neq j$). However, a batch can be distributed over more than one period, i.e. a batch can be started in a period t and finished in a period $\tau > t$ (e.g. $y_{j(t,S_t)} = y_{j(t+1,1)} = 1$).

The DLSPSD focuses the case where a setup has to be performed at the beginning of a sub-period (e.g. shift, day). But often in practice there exists the possibility to perform a setup within a period and the setup state of a machine can be preserved over idle time. In such a case the DLSPSD is an approximation only.

In [Haase 1994] a so-called capacitated lot-sizing problem with linking, denoted by CLSPL, is considered where up to J items per period are producible, the lot-sizes are continuous, and the setup state can be preserved over idle time. As in the DLSP a batch can be distributed over more than one period. Linking can occur for production quantities which are produced in two adjacent periods. To extend the CLSPL for the case with sequence dependent setup costs, denoted with CLSPLSD, we assume the triangle inequality (9). Mathematically the CLSPLSD can be stated as follows:

$$\text{minimize } Z_{CLSPLSD} = \sum_{j=1}^{J} c_{0j} z_{j0} + \sum_{i=1}^{J}\sum_{j=1}^{J}\sum_{t=1}^{T} c_{ij} x_{ijt} + \sum_{j=1}^{J}\sum_{t=1}^{T} h_j I_{jt} \tag{10}$$

subject to

$$I_{j,t-1} + q_{jt} - d_{jt} = I_{jt} \qquad (j=1,\dots,J;\ t=1,\dots,T) \tag{11}$$

$$\sum_{j=1}^{J} p_j q_{jt} \leq C_t \qquad (t=1,\dots,T) \tag{12}$$

$$C_t \left(\sum_{i=1}^{J} x_{ijt} + z_{j,t-1} \right) - p_j q_{jt} \geq 0 \qquad (j=1,\dots,J;\ t=1,\dots,T) \tag{13}$$

$$\sum_{j=1}^{J} z_{jt} = 1 \qquad (t=0,1,\dots,T) \tag{14}$$

$$\sum_{i=1}^{J} x_{ikt} + z_{k,t-1} = \sum_{j=1}^{J} x_{kjt} + z_{kt} \qquad (k=1,\dots,J;\ t=1,\dots,T) \tag{15}$$

$$f_{jt} \geq f_{it} + 1 - J(1 - x_{ijt}) \qquad (i=1,\dots,J;\ j=1,\dots,J;\ t=1,\dots,T) \tag{16}$$

$$f_{jt} \geq 0;\ q_{jt} \geq 0;\ I_{jt} \geq 0; \qquad (j=1,\dots,J;\ t=1,\dots,T) \tag{17}$$

$$x_{ijt} \in \{0,1\}; \qquad (i=1,\dots,J;\ j=1,\dots,J;\ t=1,\dots,T) \tag{18}$$

$z_{jt} \in \{0,1\};$ (j=1,...,J; t=0,1,...,T) (19)

where

C_t = capacity in period t,

f_{jt} = a dummy variable used to eliminate sub-tours,

p_j = capacity which is required to produce one unit of item j,

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

c_{0j} = setup costs which occur if item j is produced as first item in period t=1,

x_{ijt} = a binary variable. If x_{ijt} =1, then item j is setup after item i in period t (x_{ijt} =0 otherwise),

z_{jt} = a binary variable. If z_{jt}=1, then the machine is setup for item j at the end of period t and at the beginning of period t +1 (z_{jt} =0 otherwise),

and the other symbols are defined as in the DLSPSD.

(10) minimizes the objective function value. (11) and (12) contain the ordinary inventory balance and capacity constraints, respectively. (13) forces setup costs of item j to be added to the total costs if a setup occurs for item j in period t. (14) and (15) secure that only one item can be produced at the end of a period and produced further (linking) in the following period. (16) eliminates sub-tours [cf. Dilts, Ramsing 1989].

For illustration of the models we consider the following example:

Example: Let J=3, T=3, C_1=C_2=75, C_3=100, and the other data as provided in Table 1. (Note, we have to assume integers $C_t/\tilde{p}_i=C_t/\tilde{p}_j=S_t$ for t=1,...,T and i,j=1,...,J.)

Table 1 Data of the example

	d_{j1}	d_{j2}	d_{j3}	$\tilde{p}_j$	p_j	h_j	c_{0j}	c_{1j}	c_{2j}	c_{3j}	c_{j0}
j=1	15	30	40	25	1	1	40	0	40	80	0
j=2	10	15	25	25	1	1	40	40	0	80	0
j=3	0	50	25	25	1	1	80	80	80	0	0

Thus, it is $C_1/\tilde{p}_j$=75/25=S_1=S_2=3, and $C_3/\tilde{p}_j$=S_3=4 for j=1,...,J. We solve this instance as a DLSPSD as well as a CLSPLSD to optimality. The solutions are given in Table 2 where Z^* denotes the corresponding optimal objective function values.

Table 2 DLSPSD and CLSPLSD solution

	t		1			2				3		Z^*
	s=(t,k)	(1,1)	(1,2)	(1,3)	(2,1)	(2,2)	(2,3)	(3,1)	(3,2)	(3,3)	(3,4)	
	$y_{1s}\tilde{p}_1$	25	25							25	25	
DLSPSD	$y_{2s}\tilde{p}_2$			25	25							295
	$y_{3s}\tilde{p}_3$					25	25	25				
	q_{1t}		20 (2)			25 (3)				40 (7)		
CLSPLSD[1)]	q_{2t}		25 (1)							25 (6)		300
	q_{3t}					50 (4)				25 (5)		

[1)] the numbers in braces give the sequence in which the items are scheduled

Since (9) is not satisfied, e.g. $(c_{31}=80)>(c_{30}+c_{01}=40)$, the optimal solution of the DLSPSD is less costly than the corresponding CLSPLSD solution. The final inventory of the item j=1 is greater than zero, i.e. $I_{13}=\sum_{s=(1,1)}^{(3,S_3)}\tilde{p}_1 y_{1s}-\sum_{t=1}^{T}d_{1t}=$ 100-85=15. In order to save holding costs the DLSPSD solution can be improved by reducing the last production quantity of the item j=1 to 10, which reduces the total costs from 295 to 280 ($=Z^*_{\text{reduced DLSPSD}}$). However, if (9) is valid, the following inequality can be stated:

$$Z^*_{\text{CLSPLSD}} \leq Z^*_{\text{reduced DLSPSD}} \leq Z^*_{\text{DLSPSD}}.$$

This can be reasoned as follows: Due to the triangle inequality there exists an optimal DLSPSD solution where each item will be produced at most once per period t. If a batch for an item starts at most once in a period t, i.e. for j=1,...,J and t=1,..,T $\sum_{s=(t,1)}^{(t,S_t)}\sum_{i=1}^{J}x_{ijs}\leq 1$, the corresponding solution space of the DLSPSD is a subset of the solution space of the CLSPLSD.

3. A Backward-Oriented Heuristic for the CLSPLSD (BACLSPLSD)

We describe a simple heuristic for the CLSPLSD which starts with scheduling at the planning horizon and steps backwards to the first period. The lot-size and sequence decisions are performed by a simple priority rule which consists of a convex combination of holding and setup costs.

To define the priority rule we have to introduce some additional notation (cf. Haase 1994).
The cumulative demand of item j from period t to the horizon T which still has to be satisfied in the periods t,...,1 is defined by

$$D_{jt} := \max\left\{0, \sum_{\tau=t}^{T}(d_{j\tau}-q_{j\tau})\right\} \quad \text{for } j=1,\ldots,J \text{ and } t=1,\ldots,T.$$

The total still required capacity is specified by

$$\mathrm{TRC} := \sum_{j=1}^{J} p_j D_{j1}.$$

The available capacity in period t will be computed as follows:

$$AC_t = C_t - \sum_{j=1}^{J} p_j q_{jt} \quad \text{for } t=1,\ldots,T.$$

The cumulative capacity from period $\tau=1$ to period $\tau=t$ is denoted by

$$CC_t := \sum_{\tau=1}^{t} C_\tau \quad \text{for } t=1,\ldots,T.$$

Scheduling will be done backward oriented, i.e. at first the lot-sizes are determined in period t=T, then in period t=T-1, and so on. We compute a sequence

$$SEQ=(0,seq_1^1,\ldots,seq_{n_t}^1,\ldots,seq_i^t,\ldots,seq_1^T,\ldots,seq_{n_T-1}^T,0)=(0,seq^1,\ldots,seq^T)$$

where

n_t = the number of items scheduled in period t,
seq_i^t = the item which is scheduled as i-th item in period t,
seq^t = the (sub-) sequence in which the items are scheduled in period t.

The last item in a period t<T is always the first item in period t+1 (setup preserving), i.e. $seq_{n_t}^t = seq_1^{t+1}$ for t=1,...,T-1. If the capacity which is required to schedule item j in period t is less than the available capacity then item i is placed at its cheapest insertion costs position, i.e. item j is scheduled at the position k+1 if

$$c_{jt}^{\min} := c_{seq_k^t,j} + c_{j,seq_{k+1}^t} - c_{seq_k^t,seq_{k+1}^t} \le c_{seq_m^t,j} + c_{j,seq_{m+1}^t} - c_{seq_m^t,seq_{m+1}^t} \quad \text{for } m=1,\ldots,n_t-1,$$

otherwise item j is scheduled as first item in period t and as last item in period t-1 (linking).

To derive a priority value which is based on "savings" we distinguish four cases:

1) There is unsatisfied demand of item j in period t and the available capacity in period t is greater or equal than the capacity which will be required if item j is scheduled in period t, i.e. $AC_t \ge p_j D_{jt} > 0$. Thus, to schedule item j after item i in period t and not in period t-1 saves holding costs $h_j D_{jt}$ and incurs setup costs $c_{jt}^{\min}$.

2) It is $AC_t < p_j D_{jt} > 0$. If a setup occurs for item j in period t item j will be scheduled in periods t and t-1 with linking, i.e. $z_{jt} = 1$. Thus, setup costs c_{j,seq_1^t} are incurred in period t-1 and holding costs $h_j D_{j,t-1}$ $(=h_j(D_{jt}+d_{j,t-1}))$ will be saved because the amounts AC_t/p_j and $D_{j,t-1}-AC_t/p_j$ are not scheduled before t or t-1, respectively.

3) Item j is scheduled *as first* item in period t, there exists a positive demand of item j in period t-1, i.e. $q_{jt} > 0$, and $d_{j,t-1}>0$. Thus a link in period t for item j avoids setup costs, estimated by

$$\bar{c}_j = 1/J \sum_{i=1}^{J} c_{ij},$$

and holding costs $h_j d_{j,t-1}$ because the amount is not scheduled in period t-2. After linking is performed between period t and period t-1 no more changes of the schedule from period t up to period T will be performed. Thus if we "leave" period t and it is $AC_t > 0$ we have to perform a feasibility check, that is, the total still required capacity must be less or equal the available capacity from period 1 up to period t-1, i.e. $TRC \le CC_{t-1}$. In the case where a link is feasible opportunity costs for AC_t arise which will be estimated by

$$\overline{hp} \cdot AC_t, \text{ where } \overline{hp} = 1/J \sum_{j=1}^{J} h_j / p_j$$

4) Linking in period t (for item $j = seq_1^t$) does not improve the solution quality (i.e. $D_{j,t} = D_{j,t-1} = 0$), linking leads to infeasibility, or item j is not the first item in the sub-sequence of period t (i.e. $j \ne seq_1^t$).

Thus, we define the following priority value for item j in period t

$$r_{jt} := \begin{cases} (1-\alpha) h_j D_{jt} - \alpha \cdot c_{jt}^{min} & \text{if } (AC_t \ge p_j D_{jt} > 0) \wedge (j \ne seq_1^t, \ldots, seq_{n_t}^t) \\ (1-\alpha) h_j D_{j,t-1} - \alpha \cdot c_{j,seq_1^t} & \text{if } (AC_t < p_j D_{jt} > 0) \wedge (j \ne seq_1^t, \ldots, seq_{n_t}^t) \\ (1-\alpha) h_j d_{j,t-1} + \alpha \cdot \bar{c}_j - \beta \cdot \overline{hp} \cdot AC_t & \text{if } j = seq_1^t \wedge (d_{j,t-1} > 0) \wedge (TRC \le CC_{t-1}) \\ -\infty & \text{otherwise} \end{cases}$$

where $0 \le \alpha \le 1$.

The larger the r_{jt} the more preferable it is to schedule the item j in period t. Therefore the item with the largest priority value will be scheduled (**priority rule**).

By the parameter $\alpha \in [0,1]$ we will control the expected lot-sizes, e.g. if $\alpha = 1$ we expect large lot-sizes for items with high setup costs. The parameter $\beta \in [0,1]$ allows to compute different opportunity costs for unused capacity.

A formal description of the backward oriented method, denoted by BACLSPLSD, is given in the following:

BACLSPLSD

INITIALIZATION:

for j=1,...,J, t=1,...,T **do begin** q_{jt}:=0; $D_{jt} := \sum_{\tau=t}^{T} d_{j\tau}$ **end;**

$TRC := \sum_{j=1}^{J} p_j D_{j1}$; SEQ:=();

for t=1,...,T **do** $CC_t := \sum_{\tau=1}^{t} C_\tau$;

SCHEDULING:

t:=T; seq^t:=(0,0); $AC_t := C_t$

while t ≥ 0 **do**

begin

for j=1,...,J **do** compute r_{jt};

(i,k) = (item with maximal priority value, position in sequence);

if $i \in \{seq_2^t, seq_3^t,..\}$ **then begin** SEQ:=((seq_2^t, seq_3^t,...), SEQ); t:=t-1; seq^t =(0,i); **end**

else if $p_i D_{it} > AC_t$ **then**

begin

q_{it}:= AC_t/p_i; TRC:=TRC - $p_i q_{it}$; SEQ:=(i,(seq_2^t, seq_3^t,...), SEQ);

t:=t-1; seq^t=(0,i); AC_t:=C_t;

for τ=1,...,t **do** $D_{i\tau}$:=$D_{i\tau}$-q_{it};

end else $seq^t := (\ldots, seq_{k-1}^t, i, seq_k^t, \ldots)$;

q:=min{C_t/p_i,D_{it}}; AC_t:=AC_t-$p_i q_{it}$; TRC:=TRC-$p_i q_{it}$;

for τ=1,...,T **do** $D_{i\tau}$:=$D_{i\tau}$-q_{it};

end;

if $\sum_{j=1}^{J} D_{j1} > 0$ **then** Z:=∞;

Note, if for an instance a feasible solution exists, i.e. $\sum_{\tau=1}^{t}\sum_{j=1}^{J} p_j d_{j\tau} \le CC_t$ for t=1,...,T, BACLSPLSD computes a feasible solution due to the feasibility check.

The solution quality depends on the choice of the parameter values α and β. Therefore, we apply a search procedure for the parameter values $\alpha \in [\underline{\alpha}, \overline{\alpha}] \subseteq [0,1]$ and $\beta \in [\underline{\beta}, \overline{\beta}] \subseteq [0,1]$. We start with $\alpha=\underline{\alpha}$ and $\beta=\underline{\beta}$. Then β will be increased by $\Delta\beta:=(\overline{\beta}-\underline{\beta})/b$, where b is an integer greater than 2, as long as $\beta = \overline{\beta}$. If $\beta=\overline{\beta}$ α will be increased by $\Delta\alpha:=(\overline{\alpha}-\underline{\alpha})/b$ and β will be reset to $\underline{\beta}$, and so on until $\alpha=\overline{\alpha}$ and $\beta=\overline{\beta}$. Let α^{min} and β^{min} denote the respective parameter values of α and β where BACLSPLSD has computed the best solution. A more detailed search will be started with

$\underline{\alpha}$:=max{Δα/b, α^{min} - Δα(b-1)/b},

$\overline{\alpha}$ = min{1-Δα/b,α^{min} + Δα(b-1)/b},

$\underline{\beta} := \max\{\Delta\beta/b, \beta^{min} - \Delta\beta(b-1)/b\}$, and

$\overline{\beta} = \min\{1-\Delta\beta/b, \beta^{min} + \Delta\beta(b-1)/b\}$.

If no improvement around α^{min} and β^{min} has been achieved the search procedure stops, otherwise a further more detailed search will be started.

Note, if a solution exists, at least b^2 schedules will be computed. Thus the computation time for all BACLSPLSD executions depends on the choice of b.

4. Computational Study

The computational performance of BACLSPLSD is compared with the heuristic presented in [Fleischmann 1994], in the following denoted with H-DLSPSD. In [Fleischmann 1994] the instances

- TV11/S0 to TV11/S6
- TV11/S0/h0 to TV11/S6/h0
- TV12/S0 to TV12/S6
- TV13/S0 to TV13/S6
- TV14/S0 to TV14/S6
- PR1 to PR4

are considered. PR1 to PR4 relate to cases of the food industry, also considered in [Fleischmann, Popp 1989]. The instances TV11 to TV14 correspond to the data sets 11-14 of [Thizy, Van Wassenhove 1985] with J=8 and T=8. For the DLSP the periods have been divided into sub-periods of equal capacity (50 units per sub-period). The 4 problems TV11 to TV14 differ only in the capacity utilization (97,95,76 and 64%, respectively). The extensions /S0 to /S6 denotes different setup costs matrices and /h0 indicates that the holding costs for all items are zero. The matrix S0 denotes the original sequence independent setup costs of [Thizy, Van Wassenhove 1985].

The maximum in S1 is less than 1000. To provide a more "accurate" (or fair) comparison of BACLSPLSD and H-DLSPSD, we define a new setup costs matrix S7 by modification of S1 which satisfies the triangle inequality (9), i.e.

$$S7_{ij} := S1_{ij} + \begin{cases} 1000 & \text{if } i,j > 0 \ \wedge \ i \neq j \\ 0 & \text{if } i = j \ \vee j = 0 \\ 2000 & \text{otherwise, i.e. } i = 0 \end{cases}$$

Furthermore, we transform the demands of TV11 to TV14, denoted with /dt, to multiples of a full period production, i.e.

$$d'_{jt} := \left\lceil \sum_{\tau=1}^{t} d_{jt} / p_j \right\rceil p_j - \sum_{\tau=1}^{t-1} d'_{jt}.$$

where $\lceil a \rceil$ denote the smallest integer greater or equal a. Note, if BACLSPLSD solves an instance with /dt all production quantities are multiples of a full period production as in a DLSP solution.

The instances are solved with H-DLSPSD and with BACLSPLSD. The parameter b which has to be initialized for the search of well suited parameter values α and β is in all instances 4. For a given H-DLSPSD solution holding costs are ***reduced*** a posterior, by reducing the last production quantities as shown in the example [cf. Fleischmann 1994]. Let $\lfloor a \rfloor$ denotes the greatest integer less or equal a. In all instances (except S7) minimal sequence dependent setup costs for an item are incurred if the item is produced after an idle sub-period (i.e. idle-item j=0). Thus for a given CLSPLSD solution setup costs are reduced a posterior, by inserting $\left\lfloor \left(C_t - \sum_{j=1}^{J} p_j q_{jt} \right) / p_1 \tilde{p}_1 \right\rfloor$ idle-items in period t=1,...,T, if there is a choice between two items which reduces the setup costs at most. The corresponding objective function values are denoted with $\overline{Z}_{\text{reduced H-DLSPSD}}$ and $\overline{Z}_{\text{reduced BACLSPLSD}}$, respectively. In Table 3 the relations of the objective function values of instances TV11 to TV14 are entered.

Table 3 $\overline{Z}_{\text{reduced BACLSPSD}} / \overline{Z}_{\text{reduced H-DLSPSD}}$

	TV11	TV12	TV13	TV14
/S0	.938	.958	.925	.866
/S1	.898	1.03	1.05	1.15
/S2	.843	1.01	1.02	1.11
/S3	.739	.915	.844	.921
/S4	.999	.975	.911	.893
/S5	.896	.798	.838	.902
/S6	.834	.799	.829	.800
/S1/h0	.880	1.06	1.10	1.26
/S7/h0	1.02	.935	.972	.755
/S7/h0/dt	1.14	1.07	.989	.810

In most instances BACLSPLSD has computed a solution which is less costly than the corresponding DLSP solution. If the setup costs are sequence independent, i.e. $s_{ij}=S0_{ij}$, BACLSPLSD has computed for the four instances always a solution which is less costly than the corresponding H-DLSPSD solution. The difference increases with decreasing capacity utilization. For the instances TV12/S1/h0 to TV14/S1/h0 the solution quality of H-DLSPSD dominates that of BACLSPLSD whereas for TV12/S7/h0 to TV14/S7/h0 the result is vice versa. One reason for this result may be that in the DLSP solution items are produced more than once per period which reduces the setup costs. The instance TV11/S7/h0/dt indicates that the deviation between the optimum solution of CLSPLSD and the solution computed by BACLSPLSD can be large, i.e. more than 14%. The results do not indicate that BACLSPLSD in general is superior, but the small investigation shows that on the average BACLSPLSD provides a higher solution quality than H-DLSPSD.

In Table 4 the objective function values $\overline{Z}_{\text{reduced H-DLSPSD}}$ and $\overline{Z}_{\text{reduced BACLSPLSD}}$ for the instances PR1 to PR4 are given.

Table 4 Comparison of (the reduced costs of) H-DLSPSD and BACLSPLSD for the instances related to cases of the food industry

	T	number of sub-periods	J	capacity utilization	H-DLSP	BACLSPLSD
PR1	8	120	9	66	21981.55	7395.24
PR11	8	240	9	66	16309.93	7395.24
PR2	8	80	9	99	40353.94	27248.73
PR3	26	230	3	91	36090.31	28766.05
PR4	26	225	4	95	67279.60	60608.98

For the instances PR1 to PR4 the solution quality of BACLSPLSD is substantially better than the modified H-DLSPSD solution (cf. Table 4). Especially, for PR1 the H-DLSPSD solution is very poor. To compute a better approximation of a CLSPLSD solution with H-DLSPSD the number of sub-periods in PR1 per macro period are duplicated; the corresponding instance is denoted with PR11. We see that such a modification improves the solution quality substantially; however, the solution is still very poor with respect to the BACLSPLSD solution. The solution may be improved additionally if the number of sub-periods will be increased a second time, but this is not very attractive because the computation time increase enormously (cf. Table 5).

BACLSPLSD has been coded in Turbo Pascal 6.0 from Borland. H-DLSPSD has been implemented in MS FORTRAN 5.1. Table 5 gives the computation times on a PS/2 Model P70 with 80386 processor and 80387 co-processor.

Table 5 Computation times of BACLSPLSD and H-DLSPSD in seconds

instances	H-DLSPSD	BACLSPLSD
a) PR1	979	40
b) PR11	7113	40
c) average PR2 to PR4	542.3	17.3
d) average TV11/S0 to TV14/S6	651.1	24.5
e) max. of TV11/S0 to TV14/S6	2723	35
f) min. of TV11/S0 to TV14/S6	55	17
g) average TV11/S1/h0 to TV14/S1/h0	481.5	20.5
h) average TV11/S7/h0 to TV14/S7/h0	248.75	16.8

In general, BACLSPLSD is much faster than H-DLSPSD. An increase of the number of subperiods increases the computation time of H-DLSPSD substantially (cf. rows a) and b) in Table 5). There is a high (small) variability in the cpu times of H-DLSPSD (BACLSPLSD) (cf. rows e) and f) in Table 5). If the triangle inequality (9) is satisfied H-DLSPSD seems to become faster (cf. rows g) and h) in Table 5).

Thus BACLSPLSD is more efficient (on the average) than H-DLSPSD.

5. Summary

A mixed-integer programming formulation is presented for capacitaed lot-sizing with sequence dependent setup costs denoted by CLSPLSD. In the CLSPLSD it is assumed that the setup state can be preserved over idle time between adjacent periods (linking). The CLSPLSD can be solved efficiently by a backward oriented approach where lot-sizing and linking depends on a priority rule. The CLSPLSD is compared with the so-called discrete lot-sizing and scheduling problem with sequence dependent setup costs (DLSPSD). A heuristic for the DLSPSD is introduced in [Fleischmann 1994]. A computational study shows, that in the case where continuous lot-sizes can be computed as well as the setup state can be preserved over idle time, the backward oriented approach is more efficient than the DLSPSD heuristic.

Acknowledgement

The author is grateful to Bernhard Fleischmann for providing the code H-DLSPSD and to Andreas Drexl for helpful support.

References

Dobson, G., "The cyclic lot scheduling problem with sequence-dependent setups", Operations Research, Vol. 40 (1992), pp. 736-749.

Dillenberger, C., Escudero, L.F., Wollensak, A., and Zhang, W., "On solving a large-scale resource allocation problem in production planning", in: Operation research in production planning and control, Fandel, G. et al. (eds.), Springer (Berlin) 1992, pp. 105-119.

Dilts, D. M., Ramsing, K. D., "Joint lot sizing and scheduling of multiple items with sequence dependent setup costs", Decision Sciences, Vol. 20 (1989), pp. 120-133.

Fleischmann, B., "The discrete lot-sizing and scheduling problem", European J. of Operational Research", Vol. 44 (1990), pp. 337-348.

Fleischmann, B., "The discrete lot-sizing and scheduling problem with sequence-dependent setup costs", European J. of Operational Research (to appear), (1994).

Fleischmann, B., Popp, Th., "Das dynamische Losgrößenproblem mit reihenfolgeabhängigen Rüstkosten", in: Pressmar, D. et al. (eds.): "Operations Research Proceedings 1988", Springer (Berlin) 1989, pp. 510-515.

Haase, K., "Lotsizing and scheduling for production planning", Springer (Berlin) 1994.

Schrage, L., "The multiproduct lot scheduling problem", in: Dempster, M.A.H., et al. (eds.), "Deterministic and stochastic scheduling", Dordrecht/Holland, 1982, pp. 233-244.

Thizy, J.M., Van Wassenhove, L.N., "Lagrangean relaxation for the multi-item capacitated lot-sizing problem: A heuristic implementation", IIE Transactions, Vol. 17 (1985), pp. 308-313.