ELSEVIER

European Journal of Operational Research 124 (2000) 511–528

EUROPEAN JOURNAL OF OPERATIONAL RESEARCH

www.elsevier.com/locate/dsw

Theory and Methodology

The discrete lot-sizing and scheduling problem: Complexity and modification for batch availability

Wolfgang Brüggemann [a,*], Hermann Jahnke [b,1]

[a] Institut für Logistik und Transport, Universität Hamburg, Von-Melle-Park 5, 20146 Hamburg, Germany

[b] Fakultät für Wirtschaftswissenschaften, Universität Bielefeld, Universitätsstraße 25, Postfach 10 01 31, 33501 Bielefeld, Germany

Received 1 September 1996; accepted 1 March 1999

Abstract

The discrete lot-sizing and scheduling problem (DLSP) has been suggested for the simultaneous choice of lot sizes and production schedules. In the context of computational complexity, it turns out that literature results for the DLSP are incorrect. Therefore, we prove that the decision version of the DLSP is NP-hard in the strong sense. The common assumption of instantaneous availability of the manufactured units is not satisfied in practice if the units arrive in inventory only in one batch after the whole lot has been completed. Therefore, additional constraints are presented for this case of batch availability on a single machine. The resulting modified DLSP is formulated as a mixed-integer linear program. This problem can be shown to be NP-hard again using ideas similar to the item-availability case. Hence, a two-phase simulated-annealing (SA) heuristic is suggested for solving the DLSP in the case of batch availability. Numerical results are presented for different problem classes. © 2000 Elsevier Science B.V. All rights reserved.

Keywords: Manufacturing; Scheduling; DLSP; Batch availability; Complexity; Simulated annealing

1. Introduction

In production planning, different situations arise depending on the level of aggregation. For highly aggregate, medium-term planning, the capacitated lot-sizing problem (CLSP) is suggested. For example, Chen and Thizy (1990) discuss a classical model formulation, its complexity, and several solution methods for this problem. A coarse subdivision of the finite planning horizon yields periods in which all of the items can be produced subject to common capacity restrictions.

On a more detailed level of production planning with relatively short horizons, the scheduling aspect becomes more prominent. Considering the vast literature in the fields of lot-sizing and batching decisions on the one hand and scheduling aspects on the other, Potts and Van Wassenhove (1992) stress the importance of an integrated decision making in these fields because of their high degree of interrelation. In this context, Dinkelbach (1964, Part 3), already proposes a mixed-integer

* Corresponding author. Tel.: +49 40 42838 2950; fax: +49 40 42838 6283.

[1] Tel.: +49 521 106 3927; fax: +49 521 106 6036.

E-mail addresses: na.brueggemann@na-net.ornl.gov (W. Brüggemann), jahnke@wiwi.uni-bielefeld.de (H. Jahnke).

PII: S 0 3 7 7 - 2 2 1 7 (9 9) 0 0 1 9 0 - 3

linear model to address the simultaneous lot-sizing and scheduling problem. The principal idea is to divide the finite time horizon into (small) time intervals in which the machine can be used either for production of at most a single item, or can be setup for such a production. Later, Lasdon and Terjung (1971), Schrage (1981) and Fleischmann and Popp (1989) use similar ideas for different applications. Fleischmann (1990) proposes a standard model formulation for this problem which is now known as the discrete lot-sizing and scheduling problem (DLSP). Salomon et al. (1991) introduce a six-field classification scheme for different DLSP variants. For example, Fleischmann and Popp (1989) and Fleischmann (1994) examine sequence-dependent setup costs for the DLSP while Salomon et al. (1997) additionally consider sequence-dependent setup times. Drexl and Haase (1992) explore the similarities of different model types suggested for the lot-sizing and scheduling problem. Comprehensive literature reviews for the DLSP are given for example by Salomon et al. (1997), Van Hoesel et al. (1994) and Brüggemann (1995, Section 3.2.4).

The DLSP-kind of problem has been thought to be difficult without formal proof until Kroon and Salomon (1988), Salomon (1991, Section 3.3) and Salomon et al. (1991) claim the NP-completeness of the standard DLSP and certain variants. Although some of the relevant proofs given in these references are incorrect (Brüggemann and Jahnke, 1997), the DLSP indeed turns out to be a computationally difficult problem (Section 3). Despite the computational complexity, Fleischmann (1990) proposes a branch-and-bound procedure based on Lagrangean relaxation for the single-machine, multi-item DLSP and reports numerical advantages over CLSP solution methods. A 0–1 structure similar to that of the DLSP is first exploited by Glassey's (1968) network algorithm for a lot-sizing and scheduling problem. Different solution methods for the single-item DLSP can be found in Van Hoesel et al. (1994). A heuristic for the single machine, multi-item DLSP is suggested for example by Salomon (1991, Section 5.4) and similarly by Cattrysse et al. (1993).

All the modelling approaches and solution methods mentioned above pertain to the case of instantaneous availability of the manufactured units prior to the completion of the lot. However, the situation in practical short-term production planning is often different. Due to logistical, organizational, or technical circumstances, units may arrive in inventory as a complete batch only after the last of its items is processed. This mode of production is found for example if demand quantities accrue from downstream departments (Dobson et al., 1987) and is often assumed in material requirement planning (MRP) systems (e.g., Silver et al., 1988, pp. 602–610). Batch availability will be characteristic for the manufacturing process if, for example, the transport between different work centers uses pallets with a maximum capacity which defines a maximal size of the production batches (Dobson et al., 1987). For both modes of manufacturing, we use the terms *item* and *batch availability*, respectively, following the nomenclature of Santos and Magazine (1985), Dobson et al. (1987) and Webster and Baker (1995) in the context of job family scheduling.

The standard single-machine multi-item DLSP with sequence-independent setup times is described in Section 2 followed by a discussion of complexity issues for this item-availability DLSP in Section 3. The case of batch availability is treated in Section 4: To extend the DLSP model to this situation, additional constraints are needed which are presented in Section 4.1. The resulting mixed-integer linear program for the batch-availability DLSP is shown to be NP-hard in Section 4.2 using the approach from Section 3. For the solution of this difficult problem, a two-phase simulated-annealing (SA) based heuristic is suggested in Section 4.3 because of its general applicability. The numerical experiments are described in Section 5. First, the choice of problem parameters, demand data, and target inventories is discussed followed by an explanation of how the cooling schedule parameters are selected in Section 5.2 while the numerical results are presented in Section 5.3. Finally, the results are summarized in Section 6.

2. The single-machine multi-item DLSP

Ordinarily, the DLSP is formulated for N items and a finite time horizon of T time units. Here, production scheduling and demand should be considered on different time scales. Typically, demand can be estimated as demand per day or week, while the DLSP requires a finer discretization of the time axis due to the underlying idea that during one time unit (e.g., hours or 30-minutes intervals), the machine can be used at most for setup or production of only one item (see constraints (4) below). Hence, the time units for the DLSP will be chosen to be the greatest common divisor of setup times and minimal production times for all items. The different time scales of demand and scheduling yield a division of the planning period T into M (demand) subintervals of lengths $T_m - T_{m-1}$ ($m = 1, \ldots, M$; $T_0 = 0$; $T_M = T$) where demand d_{it} for product i is positive only at times T_m ($d_{i,T_m} \geqslant 0$ and $d_{i,t} = 0$ for $t = 1, \ldots, T$; $t \neq T_m$; $m = 1, \ldots, M$; $i = 1, \ldots, N$). Approximating CLSP instances by corresponding DLSP instances yields a similar demand structure (Fleischmann, 1990).

As an example, the demand schedule for a problem instance with $N = 6$ products, $T = 60$ periods and $M = 6$ equidistant demand instances ($T_1 = 10, T_2 = 20, \ldots, T_6 = 60$) is given in Table 1 where demand is measured in how many units can be produced during one production period, i.e. the production speeds are normalized to one unit per period.

This distribution of the demand over time is visualized more directly in Table 2 where the rows and columns correspond to the individual items and time periods, respectively, and the demand data which are not shown explicitly are zero. Additionally, initial inventories are given in the second column, I of Table 2.

Table 1
Demand structure for a problem with 6 products, 60 periods and 6 demand instances

Demand	Demand instances					
Product	10	20	30	40	50	60
1	3	3	2	1	2	5
2	3	1	0	1	0	3
3	0	2	0	0	2	3
4	2	0	0	1	3	1
5	0	1	0	0	2	4
6	0	0	0	1	0	1

The following notation is used to model the DLSP:

Parameters:
- M number of demand instances
- N number of products
- T number of periods

Data:
- a_i number of setup periods required before production of product i
- $d_{i,t}$ demand for product i in period t
- h_i inventory holding cost per unit of product i and period
- $I_i^{(0)}$ initial inventory for product i
- o_i production speed for product i
- r_i setup cost per setup period for product i
- T_m period number of the mth demand instance

Variables:
- $I_{i,t}$ amount of inventory at the end of period t of product i
- $v_{i,t}$ zero–one variable for setup in period t for production of product i
- $y_{i,t}$ zero–one variable for production of product i in period t

Then the DLSP can be modelled as to minimize the sum of setup and inventory holding cost:

$$\text{minimize } C(v,y) := \sum_{i=1}^{N}\sum_{t=1}^{T} r_i v_{i,t} + h_i I_{i,t} \tag{1}$$

subject to

$$I_{i,t-1} + o_i y_{i,t} - d_{i,t} = I_{i,t}, \quad i = 1, \ldots, N; \quad t = 1, \ldots, T, \tag{2}$$

$$v_{i,t-a_i+\tau} \geqslant y_{i,t} - y_{i,t-1}, \quad i = 1, \ldots, N; \quad t = a_i + 1, \ldots, T; \quad \tau = 0, \ldots, a_i - 1, \tag{3}$$

$$y_{i,t} = 0, \quad i = 1, \ldots, N; \quad t = 1, \ldots, a_i, \tag{3a}$$

$$v_{i,t} \geqslant y_{i,t} - y_{i,t-1}, \quad i = 1, \ldots, N; \quad t = 1, \ldots, T; \quad a_i = 0, \tag{3b}$$

$$\sum_{i=1}^{N}(y_{i,t} + v_{i,t}) \sum_{\substack{i=1 \\ a_i>0}}^{N} v_{i,t} \leqslant 1, \quad t = 1, \ldots, T, \tag{4}$$

Table 2
Explicit demand structure for the problem in Table 1

```
  |   |         1         2         3         4         5         6
  | I |123456789012345678901234567890123456789012345678901234567890
--|---|------------------------------------------------------------
1 | 3 |         3         3         2         1         2         5
2 | 1 |         3         1         0         1         0         3
3 | 3 |         0         2         0         0         2         3
4 | 1 |         2         0         0         1         3         1
5 | 1 |         0         1         0         0         2         4
6 | 1 |         0         0         0         1         0         1
```

$$y_{i,t} \in \{0,1\}, \quad I_{i,t}, v_{i,t} \geqslant 0, \quad i = 1,\ldots,N; \quad t = 1,\ldots,T,$$

$$I_{i,0} = I_i^{(0)}, \quad i = 1,\ldots,N.$$

The objective (1) and constraints (2), (3) and (4) are taken from the standard model for the DLSP (Salomon, 1991, pp. 37 and 43). Parameters r_i in (1) are sequence-independent setup costs per setup period and the inventory holding costs are given by the product of parameter h_i (costs per unit and period) and nonnegative inventories $I_{i,t}$. The demand-satisfaction or inventory-balance constraints (2) describe the dependence of current inventories on the inventories of the preceding period, the quantity produced (where o_i is the production speed, i.e., the constant number of units manufactured in one period) and the demand. The correct sequence of setup and production periods is modelled in Eq. (3) where $a_i > 0$ is the corresponding number of setup periods; constraints (3a) prohibit production of item i in periods $1,\ldots,a_i$ with no preceding setup while the case of vanishing setup times is modelled in (3b). Constraints (4) are used to prevent simultaneous action (setup or production) on the machine. The setup variables do not need to be restricted to binary values explicitly because they assume any value larger than zero in an optimal solution only if required by constraints (3). In this case, however, they will not be larger than one due to their cost contribution. The variables $I_{i,0}$ are set to the given initial inventories $I_i^{(0)}$.

A possible solution for the example above with setup times of one period for each item is given in Fig. 1 where the `R` stands for a setup period while production is indicated by `1`.

This solution is feasible for the item-availability DLSP. For example, the demand for product 1 in period 10 can be satisfied by stocked units. In the second demand instance, units have to be taken from the current production. This is possible in the case of item availability considered here (but not for batch availability as introduced below in Section 4).

3. Complexity results for the single-machine multi-item DLSP

The DLSP problem type has been thought to be difficult to solve without formal proof until Salo-

```
  |   |         1         2         3         4         5         6
  | I |123456789012345678901234567890123456789012345678901234567890
--|---|------------------------------------------------------------
1 | 3 |               R1111111111111
2 | 1 |R1111111
3 | 3 |                                    R1111
4 | 1 |        R111111
5 | 1 |                             R111111
6 | 1 |                                         R1
```

Fig. 1. Feasible production schedule for the demand structure of Table 1.

mon (1991, Section 3.3) and Salomon et al. (1991) claim the NP-completeness of the DLSP and certain of its variants. However, some of the proofs given there are incorrect (Brüggemann, 1995; Brüggemann and Jahnke, 1997).

The complexity of the DLSP with sequence-independent setup costs and no setup times (this is the 1/*/SI/G/A in the six-field notation by Salomon et al. (1991)) is considered first. Salomon et al. suggest to reduce the set partition problem (SPP)

DATA: Given K integers $A_1, \ldots, A_K$.

QUESTION: Does there exist a subset $\Theta \subseteq \{1, \ldots, K\}$ such that

$$\sum_{n\in\Theta} A_n = \sum_{n\notin\Theta} A_n =: B \quad \text{holds?}$$

to the DLSP by the following transformation:

For an arbitrary SPP instance, the number of items N in the resulting DLSP instance equals $K + 1$ while the number of periods T is set to $2B + 1$. Further details of the transformation by Salomon et al. are not important for the following argument.

The data of a DLSP instance is given by the number of products N and the number of periods T, demand $d_{i,t}$, setup cost r_i, setup times a_i, initial inventories $I_i^{(0)}$, inventory holding cost h_i and production speed o_i. Assuming that all parameters are chosen to be bounded by a constant, the length of this DLSP instance is essentially given by the size of the demand matrix, i.e., $O(NT)$ or equivalently using the SPP symbols $O(KB)$. But this value is not bounded by a polynomial in $O(K \log_2 B)$, i.e., in the length of the SPP instance. Hence, the data of the resulting DLSP instance cannot even be written in polynomially bounded time and the transformation is not computable by a polynomial-time algorithm. Therefore, the first of the two necessary conditions for polynomial transformations in NP-completeness proofs, which can be found in Garey and Johnson (1979, pp. 34–35), is violated.

Consider the following example as an illustration for the argument. Let the data of a particular (pathological) SPP instance be given by

$$A_k = 2^k \quad \text{for } k = 1, 2, \ldots, K.$$

Then

$$B = \frac{1}{2}\sum_{k=1}^{K} A_k = \frac{1}{2}\sum_{k=1}^{K} 2^k = 2^K - 1 \approx O(2^K).$$

The corresponding SPP length yields $O(K \log_2 B) = O(K \log_2 2^K) = O(K^2)$. In the resulting DLSP instance, there are

$$T = 2B + 1 = 2^{K+1} - 1$$

time periods, the size of the demand matrix is

$$NT = (K + 1)(2^{K+1} - 1)$$

and the DLSP length $O(K2^K)$ is essentially exponential in the SPP length.

If the data in the DLSP model formulation (1)–(4) are commonly bounded by a constant, the length of a DLSP instance is essentially determined by the way of storing the demand. In the natural encoding scheme, the full demand matrix is stored including the zero elements (compare the demand matrix in Table 2 for the example in Section 2). However, this turns out not to be ‘reasonable’ in the terminology of Garey and Johnson (1979). A ‘reasonable’ encoding scheme can be constructed by neglecting the zero elements of the demand matrix. In this case, the transformation by Salomon et al., from the SPP to the DLSP becomes polynomial again. Considering this combination of ‘reasonable’ encoding and standard model formulation, however, it is questionable whether the decision version of the DLSP can be solved in polynomially bounded time on a non-deterministic Turing machine, and hence whether it is contained in the class NP. (Brüggemann (1995, Section 3.3) and Brüggemann and Jahnke (1997); Webster (1999) points out that these last two observations suffice to conclude that the DLSP decision version is at least NP-hard.)

This can be illustrated by the following (again pathological) example. Consider the single-item ($N = 1$) DLSP instance where demand occurs only in the last period T. The cost coefficients are bounded by a constant and are therefore irrelevant; the initial inventory is zero ($I_1^{(0)} = 0$). Hence, the length of this instance in the ‘reasonable’ encoding is essentially $O(\log_2 T)$ whereas its length using the standard encoding scheme is at least

O(T). This, however, grows exponentially in the 'reasonable' length. Consequently, the standard encoding is not 'reasonable'. Moreover, the number of decision variables is T and the verification of a possible solution by assigning values to the decision variables and evaluating the constraints may require exponential time.

The two classes of encoding schemes yield different results only for pathological instances with specific demand structures while the complexity is independent of the encoding scheme, if the demand is distributed more uniformly. In this context, the complexity of the DLSP using the 'unreasonable' standard encoding scheme can be interpreted as the difficulty of typical instances. Moreover, it is possible to derive complexity results for the DLSP which hold for both encoding schemes. Since the corresponding proof can be applied analogously to the DLSP for the case of batch availability below we consider the standard case of item availability first.

Theorem. *The decision version of the DLSP with setup cost is NP-hard in the strong sense.*

Proof. Here, the DLSP with setup times as in the model (1)–(4) is considered while the proof can easily be modified for vanishing setup times. In order to show that the DLSP is NP-hard in the strong sense, a pseudo-polynomial transformation from the strongly NP-complete 3-partition problem (3PP) (Garey and Johnson, 1979, p. 96) to the DLSP decision problem is constructed. For a given collection of $3J$ integers $A_1, \dots, A_{3J}$ and a bound B with

$$\frac{B}{4} < A_j < \frac{B}{2} \quad \text{for } j = 1, \dots, 3J \quad \text{and} \quad \sum_{j=1}^{3J} A_j = JB$$

the 3PP can be stated as the question whether there exists a collection of disjoint subsets

$$\Theta_1, \dots, \Theta_J \subseteq \{1, \dots, 3J\},$$

which satisfies

$$\sum_{k \in \Theta_j} A_k = B \quad \text{for } j = 1, \dots, J.$$

An arbitrary instance of the 3PP is mapped to a DLSP instance with $N := J + 1$ items and

$$T := \sum_{j=1}^{3J} A_j + 6J - 1$$

periods. This demand structure is shown in Table 3. A positive demand occurs for products $i = 1, \dots, J$ in the last period T only while there is demand of one unit for product N in certain demand instances.

The remaining data of the resulting DLSP instance are given by

$$\begin{aligned}
o_i &:= 1 \quad \text{for } i = 1, \dots, N, \\
I_i^{(0)} &:= 0 \quad \text{for } i = 1, \dots, N, \\
r_i &:= 1 \quad \text{for } i = 1, \dots, J, \\
r_N &:= 0, \\
h_i &:= 0 \quad \text{for } i = 1, \dots, J, \\
h_N &:= T, \\
a_i &:= 1 \quad \text{for } i = 1, \dots, J, \\
a_N &:= 0.
\end{aligned}$$

Table 3
Demand structure for resulting DLSP instance

	A_1+1	1	A_2+1	1	A_3+1	1			$A_{3J}+1$
	123456...		...		...		...		T
1	00 ... 0	0	00 ... 0	0	00 ... 0	0		0	0 ... 0B
2	0 ... 0	0	0 ... 0	0	0 ... 0	0		0	0 ... 0B
⋮	⋮		⋮		⋮		⋱	⋮	⋮
J	0 ... 0	0	0 ... 0	0	0 ... 0	0	...	0	0 ... 0B
N	0 ... 0	1	0 ... 0	1	0 ... 0	1		1	0 ... 00

The upper bound on the objective function of the decision problem is chosen to be $3J$. In order to meet this bound, the last item has to be produced exactly during the periods in which there is positive demand for it. In this way, there arise $3J$ sets of consecutive idle periods which can be used for production of the remaining items. During each of these sets, only a single item can be manufactured due to capacity restrictions. Thus, the 3PP instance is a 'yes' instance iff the resulting DLSP instance has a feasible solution with objective function value $3J$ and the first condition for a pseudo-polynomial transformation is satisfied (Garey and Johnson, 1978).

Moreover, the transformation can be computed within a pseudo-polynomially bounded time, because the size of the demand matrix is given by

$$NT = (J+1)\left(\sum_{j=1}^{3J} A_j + 6J - 1\right) < (J+1)^2(B+6).$$

Hence, the time required for the computation of this transformation is essentially bounded by $O(J^2B)$ which is a two variable polynomial in the length and maximum value of the 3PP instance and the second condition for pseudo-polynomial transformations holds.

The verification of the remaining two conditions for a pseudo-polynomial transformation is straightforward. The resulting DLSP instance is larger than the corresponding 3PP instance, and its maximum value is given by T and hence bounded again by $O(J^2B)$. Therefore, the transformation is pseudo-polynomial in the sense of Garey and Johnson (1978) and the decision version of the DLSP is NP-hard in the strong sense. □

As a consequence of this theorem, the optimization version of the DLSP with setup costs and times is NP-hard in the strong sense, too. Since the model considered here is a special case of the DLSP with sequence-dependent setup costs and setup times these variants are also NP-hard in the strong sense.

Even the feasibility problem of the DLSP with non-vanishing setup times is NP-hard (Webster, 1999). The fact that this feasibility problem and all of the decision problems mentioned above are NP-hard (the latter even in the strong sense) can be extended to corresponding NP-completeness results by showing that these models are contained in the class NP. This can be done in two ways.

First, if the standard encoding is assumed for the demand data, the length of an instance is essentially given by the size of the demand matrix, i.e., $O(NT)$. It is possible to read suitably guessed values for the DLSP decision variables, $y_{i,t}$ and $I_{i,t}$, and to evaluate the objective and constraints (1)–(4) in polynomially (depending on this length) bounded time. Therefore, the decision version of the DLSP – a similar argument pertains to the feasibility problem – is a member of the class NP and is hence NP-complete in the strong sense. On the other hand, the same result for a 'reasonable' encoding of the demand data can be achieved if a completely different model is used (Brüggemann et al., 1999).

4. The single-machine multi-item DLSP in the case of batch availability

4.1. Modification of the DLSP model for batch availability

Under the assumption of batch availability, units of a product being processed can be used to satisfy demand only after the whole lot has been completed. Constraints (2)–(4) alone are not sufficient to model this mode of manufacturing. Therefore, a model formulation for the batch-availability DLSP is given here. The inventory including work in process $I_{i,t}$ (as described above) is still required for computing holding costs in the objective (1). However, the last batch begun before time t is not ready for satisfying demand if the production process of this batch is not finished by t. Hence, auxiliary inventory $\tilde{I}_{i,t}$ is introduced accordingly as the number of units which remain available for demand satisfaction at the end of period t.

In the example of Table 1 with the solution in Fig. 1, the units of item $i=1$ which are produced in the batch starting in period 16 only become available in period 29 in this case. Since these units are needed for demand satisfaction in period $t = 20$ this solution is not feasible anymore under batch availability. The available inventory for the first item in the solution of Fig. 1 for the example is $\tilde{I}_{1,10} = 0$, $\tilde{I}_{1,20} = -3$, $\tilde{I}_{1,30} = 8$, $\tilde{I}_{1,40} = 7$, $\tilde{I}_{1,50} = 5$ and $\tilde{I}_{1,60} = 0$, where the negative value for $\tilde{I}_{1,20}$ indicates the infeasibility.

In order to model the available inventory, additional variables

$e_{i,t}$ zero–one variable for the end of production of product i in period t and

$\tilde{I}_{i,t}$ amount of inventory of product i which is available in period t

are needed and the DLSP model (1)–(4) is extended by

$$e_{i,t} \geqslant y_{i,t} - y_{i,t+1}, \quad i = 1, \ldots, N; \quad t = 1, \ldots, T, \quad (5)$$

$$e_{i,t} \leqslant y_{i,t}, \quad i = 1, \ldots, N; \quad t = 1, \ldots, T, \quad (6)$$

$$e_{i,t} \leqslant (1 - y_{i,t+1}), \quad i = 1, \ldots, N; \quad t = 1, \ldots, T, \quad (7)$$

$$\tilde{I}_{i,t} \leqslant \tilde{I}_{i,t-1} - d_{i,t} + Le_{i,t}, \quad i = 1, \ldots, N; \quad t = 1, \ldots, T, \quad (8)$$

$$\tilde{I}_{i,t} \geqslant \tilde{I}_{i,t-1} - d_{i,t} - Le_{i,t}, \quad i = 1, \ldots, N; \quad t = 1, \ldots, T, \quad (9)$$

$$\tilde{I}_{i,t} \leqslant I_{i,t} + L(1 - e_{i,t}), \quad i = 1, \ldots, N; \quad t = 1, \ldots, T, \quad (10)$$

$$\tilde{I}_{i,t} \geqslant I_{i,t} - L(1 - e_{i,t}), \quad i = 1, \ldots, N; \quad t = 1, \ldots, T, \quad (11)$$

$$e_{i,t}, I_{i,t}, \tilde{I}_{i,t}, v_{i,t} \geqslant 0, \quad i = 1, \ldots, N; \quad t = 1, \ldots, T, \quad (12)$$

$$y_{i,t} \in \{0, 1\}, \quad i = 1, \ldots, N; \quad t = 1, \ldots, T, \quad (13)$$

$$Y_{i,T+1} = 0; \quad I_{i,0} = \tilde{I}_{i,0} = I_i^{(0)}, \quad i = 1, \ldots, N. \quad (14)$$

Here, zero–one end-of-batch variable $e_{i,t}$ indicates the production of the last unit of the current batch in t by $e_{i,t} = 1$. End-of-batch constraints (5)–(7) are similar to setup constraints (3). It should be pointed out that the $e_{i,t}$ again assume only values of zero or one by construction without explicit binary constraints. The available inventory $\tilde{I}_{i,t}$ is defined in constraints (8)–(11) using a large number L. These conditions are binding alternatively in pairs depending on the value of end-of-batch variables $e_{i,t}$. Restrictions (8) and (9) are the inventory balance conditions for the available inventories if no processing of a batch is completed in the same period for the same product ($e_{i,t} = 0$). If on the other hand a batch of product i is completed in period t ($e_{i,t} = 1$), these constraints are not restrictive but the standard inventory balance constraints (2) still guarantee demand satisfaction. In this case, available inventories of the corresponding product are adjusted to be equal to the inventories including work in process $I_{i,t}$ as enforced by (10) and (11). Constraints (12) and (13) are the standard non-negativity and binary constraints, respectively. Constraints (14) are the technical conditions to define the variable $y_{i,T+1}$ needed for the end-of-batch constraints and to initialize the inventory variables.

4.2. Complexity of the batch-availability DLSP

If a solution for the DLSP instance resulting from the transformation in the proof of the theorem in Section 3 is feasible, no demand needs to be satisfied prior to completion of the corresponding batch. Therefore, the batch-availability DLSP is also NP-hard, because the transformation given above pertains to the case of batch availability, too. Because of these complexity results, the existence of an efficient and exact algorithm for the DLSP is unlikely.

For the case of batch availability, it is intuitively clear that it can be difficult to find a solution which satisfies conditions (5)–(14) for instances with high capacity utilization. Because of this numerical difficulty, it is advantageous to construct a two-phase solution procedure with an explicit search for a feasible solution. It is thereby possible to control the numerical effort for the feasibility search individually.

4.3. A two-phase simulated-annealing based heuristic for the batch-availability DLSP

The proposed model for the case of batch availability is innovative and no solution approaches are available. It is very unlikely to find an efficient exact algorithm due to the complexity results. In addition to the theoretical aspects discussed so far, one possible heuristic to actually solve this computationally difficult problem is hence proposed in the following.

There exist several general purpose heuristics, e.g., genetic algorithms, tabu search, and simulated annealing (SA). SA can easily be adapted to minor modifications of an original problem. Such model modifications are often necessary in practical applications. Since it is usually far more complicated to represent a solution as a genotype for a genetic algorithm and since tabu search lacks any theoretical convergence guarantee, we choose SA which has been applied very successfully to the related job-shop scheduling problem (Van Laarhoven et al., 1992) as the basis for our two-phase solution procedure.

Simulated annealing, first suggested by Kirkpatrick et al. (1983) and Černý (1985), is a multipurpose heuristic for the solution of combinatorial optimization problems. A neighbourhood structure is superimposed on the usually finite but large space of feasible solutions (configurations or in this context production schedules). Given a current feasible configuration $(v_{\text{cur}}, y_{\text{cur}})$ a candidate solution $(v_{\text{can}}, y_{\text{can}})$ is drawn randomly from the corresponding neighbourhood. This new configuration will be accepted subject to either improvement of the objective function or another random experiment with acceptance probability given by $e^{-\Delta C/\gamma}$, where $\Delta C = C(v_{\text{can}}, y_{\text{can}}) - C(v_{\text{cur}}, y_{\text{cur}})$ is the difference of the cost function values of the candidate and the current configuration. γ is a control parameter corresponding to temperature in the original physical analogue in thermodynamics (Metropolis et al., 1953). Infinite repetition of this procedure with a fixed value of control parameter γ can be viewed as one realization of a homogeneous Markov chain where the current state of the Markov chain is the last accepted configuration. Iterative reduction of the temperature (i.e., γ) yields a sequence of such Markov chains.

The main advantage of SA relative to tailored solution methods is its general applicability. Solving a specific problem with SA requires only determination of a cooling schedule (i.e., choice of the sequence of control parameters γ and number and length of the finite approximations of the homogeneous Markov chains) and a neighbourhood structure. Different cooling schedules are discussed for example by Van Laarhoven (1988). The neighbourhood choice is usually based on the set of feasible configurations only. Feasibility in the context of the batch-availability DLSP is mainly given by the demand-satisfaction under batch production (8)–(11) and setup constraints (3)–(4). For a general problem instance, it is therefore necessary to construct an initial feasible solution disregarding costs and to optimize with respect to the objective function in a second phase. The overall structure of the two-phase algorithm outlined in the following is similar to the two-phase simplex method for the solution of linear programs.

The neighbourhood structure for a given production schedule (v, y) is defined by reducing (v, y) to a 'pure' production schedule y^{pure} by eliminating setup periods. For the solution of the example in Section 2, this reduction is shown in Fig. 2.

An element $(v_{\text{can}}, y_{\text{can}})$ from the neighbourhood of (v, y) is then obtained by exchanging the activities of two arbitrary periods in y^{pure} and expanding (i.e., inserting setup periods in front of each

```
       |    |          1         2         3         4         5         6
       | I  |123456789012345678901234567890123456789012345678901234567890|
  ___  |____|____________________________________________________________|___
  1    | 3  |             1111111111111                                  |
  2    | 1  |1111111                                                     |
  3    | 3  |                                1111                        |
  4    | 1  |       111111                                               |
  5    | 1  |                          111111                            |
  6    | 1  |                                    1                       |
  ___  |____|____________________________________________________________|___
```

Fig. 2. Pure production plan y^{pure} for the solution in Fig. 2.

production batch and shifting later production by the corresponding number of time units) $y_{\text{can}}^{\text{pure}}$ to $(v_{\text{can}}, y_{\text{can}})$; see Figs. 3 and 4 for an exchange of the active periods 16 and 30 and the subsequent expansion, respectively. During expansion of $y_{\text{can}}^{\text{pure}}$ to $(v_{\text{can}}, y_{\text{can}})$ a necessary condition for feasibility (the last batch must be finished in or before period T) is checked and if this condition is violated another configuration is drawn out of the neighbourhood of (v, y). This procedure does not ensure feasibility in the sense of (2) to (14) since the demand-satisfaction constraints might not be satisfied. Therefore, search for a feasible solution is subject of the first phase while feasibility is always maintained during the second phase of the heuristic.

Phase 1: A first, possibly infeasible, production schedule is chosen to consist of a single batch for each item where the batch size is determined to satisfy the cumulated demand in T and production is carried out as soon as possible in the chronological order of cumulated demand exceeding initial inventory. Fig. 1 in Section 2 shows the initial production schedule for the example instance. For this first (perhaps infeasible) production schedule, hypothetical inventories are computed which would be needed to fulfill the demand satisfaction constraints (8)–(11) in each demand instance. Afterwards, the sum of the positive deviations of these hypothetical inventories from the actual inventories form the objective function and are minimized in phase 1. Similar to the simplex method, a feasible production schedule is found if this sum vanishes. SA will yield more feasible (in the sense that production is finished by T) production schedules by restricting the exchange to active production periods only and neglecting the final idle periods.

A first feasible production schedule for the example of Table 1 can be found in Fig. 5 where the cost parameters are as defined in Section 5. Note the rather intricate structure of the solution with many small batches produced by phase 1. This is due to the fact that the optimization in phase 1 is not cost oriented.

Phase 2: The actual optimization with respect to the cost function is carried out in phase 2. Here, the same neighbourhood structure is used, but exchanges are now permitted between arbitrary (active and idle) periods of y^{pure}. Production schedules that are not feasible in the sense of (8)–(11) are not considered as candidates. Thus, feasibility is preserved before each acceptance in

```
  |   |         1         2         3         4         5         6
  | I |123456789012345678901234567890123456789012345678901234567890
--|---|------------------------------------------------------------
1 | 3 |             11 1111111111   1
2 | 1 |1111111
3 | 3 |                                1111
4 | 1 |       111111
5 | 1 |               1          111 11
6 | 1 |                                    1
```

Fig. 3. Pure production plan $y_{\text{can}}^{\text{pure}}$ after the activities in periods 16 and 30 in Fig. 2 have been exchanged.

```
  |   |         1         2         3         4         5         6
  | I |123456789012345678901234567890123456789012345678901234567890
--|---|------------------------------------------------------------
1 | 3 |               R11  R1111111111    R1
2 | 1 |R1111111
3 | 3 |                                        R1111
4 | 1 |        R111111
5 | 1 |                  R1           R111  R11
6 | 1 |                                              R1
```

Fig. 4. Production schedule after expanding the pure production plan in Fig. 3.

```
   |   |         1         2         3         4         5         6
   | I |123456789012345678901234567890123456789012345678901234567890
---|---|------------------------------------------------------------
 1 | 3 |          R1  R1111      R11111           R111
 2 | 1 |     R11    R1                 R1       R1        R1      R1
 3 | 3 |R11                    R1           R1
 4 | 1 |        R1                       R11  R1        R1  R1
 5 | 1 |   R1                R1                       R1      R111
 6 | 1 |                   R1
---|---|------------------------------------------------------------

                 Used Capacity:          100.0 %
                 Objective Function:    2151.0
```

Fig. 5. First feasible production schedule found by phase 1 for the demand structure of Table 1.

phase 2 subject to the acceptance criterion corresponding to the original cost function (1). Typically, the optimized production schedules are more plausible than the first feasible ones. Depending on cost parameters, they exhibit fewer and larger batches with a resulting small total of setup costs and less used machine capacity. Production is carried out as late as possible in order to save inventory holding costs. See Fig. 6 for the example and note the improvement of the objective function value. The structural improvements make the final production schedule preferable to the first feasible solution even if cost parameters are not known precisely.

5. Numerical experiments

5.1. Demand data and target inventories

For the numerical evaluation of the batch-availability DLSP solution procedure, we consider problems with six products and 60 periods (6/60-problems), as well as 8/80- and 10/100-problems. Every tenth period is a demand instance for all products. Demand for each product is measured in how many units can be produced during one production period. Thus, all production speeds are equal to one. Inventory costs for all products are assumed to be equal and unity. The final inventory for each item is assumed to be equal to the given initial inventory and added to the demand of the last period. A setup of one period is required for each batch and the corresponding costs are set to 60.

During the numerical experiments, it turns out to be necessary to differentiate between hard and easy problems. As an elementary indicator of difficulty, we use the ratio of the total demand plus one setup period per product and the length of the planning period. The elementary degree of difficulty (DOD) is the capacity utilization of the first (possibly infeasible) solution. Note that two problems with the same DOD can be quite

```
   |   |         1         2         3         4         5         6
   | I |123456789012345678901234567890123456789012345678901234567890
---|---|------------------------------------------------------------
 1 | 3 |                R111  R11111                          R11111
 2 | 1 |      R111                  R1111
 3 | 3 |                                            R1111
 4 | 1 |    R1                           R11111
 5 | 1 |                                         R11     R1111
 6 | 1 |                                       R1
---|---|------------------------------------------------------------

                 Used Capacity:           80.0 %
                 Objective Function:    1221.0
```

Fig. 6. Final phase-2 production schedule for the demand structure of Table 1.

different with respect to actual difficulty due to different demand distributions over time.

The DOD intervals in the 6/60-, 8/80-, and 10/100-problem classes are chosen according to similar computation times of the corresponding problems. The intervals found for the 10/100-problem class are separated by 70%, 80%, 85% and 88% and reveal the most accentuated distinction of average computation times. This structure can be projected onto the 6/60- and 8/80-problem classes. Here, it is not possible, however, to find substantial differences in computation time for the two DOD intervals from 70% to 80% and from 80% to 85%. Therefore, these two intervals can be joined and we use the resulting coarser interval structure for the smaller problem classes. All DOD intervals contain their left boundary value with the exception of the 85–88% interval which is open. Problems with an 88% or higher DOD are considered to be toughest and the highest DOD considered is 92% since the set of feasible solutions becomes very small beyond this limit (it might even be empty) and hence there is no decision problem of any interest left. Since Banker et al. (1988) find in their empirical study that capacity utilization ranges from 45% to 88% in different branches of the US industry for the period from 1974 to 1984, the data of our numerical experiments should cover a representative range of practical situations with respect to this aspect.

For each DOD interval, 15 demand schedules are generated randomly. The items can be distinguished with respect to their demand expectations and variances. For example, there are products with high expected values and low variances, intermediate moments, and products with low expected demand and high variances.

In many practical situations, there is no natural time limitation for production processes and demand will be stochastic. Hence finite-time-horizon models are applied repeatedly in order to solve approximately the underlying infinite planning problem and to incorporate new estimates based on more available data in each planning instance. In this context, a special emphasis is put on the final inventories. Inventory parameters have to be determined which can be used as target (final) inventories for production planning. In the following numerical experiments, initial inventories are assumed to be equal to these target inventories. Reasonable values for these inventory parameters can be found for example by solving the underlying problem with the expected demand and parametrically altered initial inventories. Since demand will be stochastic in practical production planning, the data used for solving such problems are estimates of the future demand. Ex post, actual demand deviates from these estimates. Therefore, initial inventories are defined as the optimal inventories with respect to the expected demand plus an additional safety stock component. The target inventories used for the following numerical experiments are given in the example of Table 2. They are based on the more detailed numerical analysis by Brüggemann and Jahnke (1993).

5.2. Cooling schedules

Phase 1 is concerned with finding a feasible solution rather than minimizing cost. Here, we use a fairly large number of repetitions at high temperature with slow cooling. By the resulting increase of the acceptance probability the search for a feasible solution is less restricted. Usually, such a cooling schedule yields large computation times. However, this is not necessarily the case here, since phase 1 terminates immediately after a feasible solution is found. In most practical applications, finding a feasible production plan will be of primary interest when cost parameters are not easily estimated. Therefore, the cooling schedule for phase 2 (optimization) is chosen to be coarser than for phase 1 (feasibility). The main task of phase 2 is to improve the initial feasible solution and to generate sensible production plans in reasonably short computation times. Suboptimal solutions obtained by this rough procedure are improved by shifting batches to the right in order to fill unnecessary gaps.

In our numerical experiments, a geometric cooling schedule is applied, the number of repetitions is given by ‘acceptances max constant’ and no acceptances at one temperature stage ‘max constant’ is used as the stopping criterion (for notation see Collins et al. (1988)). This means that the temperature control parameter is kept constant for a fixed number of iterations unless the number of acceptances exceeds a prescribed bound. The value of the control parameter is then multiplied by a positive constant factor less than one until no acceptances occur during the fixed number of iterations on one stage or a maximal number of temperature reductions is reached.

For the DLSP with 6 items and 60 periods, appropriate cooling schedule parameters are found empirically by a test sequence for selected instances. These values are extrapolated and additionally adjusted for the larger problems depending on their size. The initial temperature of phase 1 is determined according to the expected changes in the hypothetically needed inventories during the neighbourhood search. Similarly, phase 2 cooling starts at a temperature which depends on the setup costs per period and inventory holding costs, since these two parameters determine the magnitude of change in the objective function. The maximal numbers of repetitions and acceptances for phase 1 are chosen proportional to NT while these parameters of the cooling schedule for phase 2 grow linearly in the number of products N only. The cooling schedules used in the numerical experiments are shown in Table 4. The actual maximal number of repetitions and acceptances are rounded to smooth values.

Table 4
Simulated annealing cooling schedules

Parameter	Phase 1	Phase 2
Initial temperature	10	100
Reduction	0.94	0.9
Max number of acceptances	$0.24NT + 30$	$15N - 20$
Max number of repetitions	$19NT + 3000$	$375N - 1125$
Number of reductions	80	50

5.3. Numerical results

The two-phase SA solution procedure is programmed in FORTRAN 77 and implemented on an IBM compatible NEC PC with an 80 486/20 MHz processor. It is applied to each of the problem realizations ten times with new seeds for the random number generator. The results for the different problem and DOD classes are summarized in Tables 5–7, respectively. The first column (after the DOD column) in each of these tables contains the average CPU times for phase 1 where the averages are taken over all ten runs for the 15 instances in a DOD class. Analogously, the average CPU times for complete two-phase SA runs are listed in the second column. Afterwards the coefficient of variation (CV) of the average (taken

Table 5
Results for the 6-product/60-period DLSP

Degree of difficulty (%)	Average CPU time Phase 1 (s)	Average CPU time (s)	CV of average CPU time (%)	Average deviation from LB (%)	Average CV of cost (%)	Unsuccessful trials	Unsolved problems
60–70	2	202	6.15	10.70	2.53	–	–
70–85	22	215	12.44	12.29	3.41	–	–
85–88	124	292	22.94	13.80	3.02	–	–
88–92	263	375	24.81	13.08	3.12	–	–

Table 6
Results for the 8-product/80-period DLSP

Degree of difficulty (%)	Average CPU time Phase 1 (s)	Average CPU time (s)	CV of average CPU time (%)	Average deviation from LB (%)	Average CV of cost (%)	Unsuccessful trials	Unsolved problems
60–70	3	699	5.01	10.97	2.37	–	–
70–85	48	803	8.46	12.01	3.03	–	–
85–88	609	1226	8.29	11.92	3.06	–	–
88–92	1742	1788	10.96	14.85	1.42	22	5

Table 7
Results for the 10-product/100-period DLSP

Degree of difficulty (%)	Average CPU time Phase 1 (s)	Average CPU time (s)	CV of average CPU time (%)	Average deviation from LB (%)	Average CV of cost (%)	Unsuccessful trials	Unsolved problems
60–70	19	1437	3.49	12.83	1.89	–	–
70–80	136	1698	8.30	16.30	2.52	–	–
80–85	676	2171	15.24	16.83	2.43	–	–
85–88	1650	2963	17.87	15.90	2.08	3	–
88–92	3611	3988	13.43	15.20	1.87	11	5

over the ten runs) CPU times for the different instances is given in the third column.

Considering the complexity of the DLSP, the CPU times are still reasonable even for the large instances. However, they tend to grow with problem size and DOD. A comparison of average computation times with respect to problem class and level of difficulty is given in Fig. 7 which suggests that the rate of change of the CPU-time graphs increases with problem size. In particular, phase 1 can require a significant amount of computation time for the more difficult problems. The small CVs of the average CPU times demonstrate that the average (and hence the expected) CPU time needed to solve the problem remains at a comparable level for different instances of the same problem class.

Visual inspection of the solutions shows that the results which are obtained in a reasonable amount of time exhibit a plausible structure similar to that of Fig. 6. However, it is difficult to assess the quality of the solutions rigorously since the problem classes considered here are in general not amenable to exact solution procedures because of the computational complexity of the DLSP. Therefore, 3/30-problem realizations of the different DOD classes are solved exactly using XPRESS-MP (version 10). In many cases, an optimal solution is found also by the two-phase SA algorithm (with cooling schedule parameters as for the 6/60-problems). The overall average deviation between the best solution found by SA and the exact optimum is 1.8% and the CV of the objective function values is 5.24% for these problems.

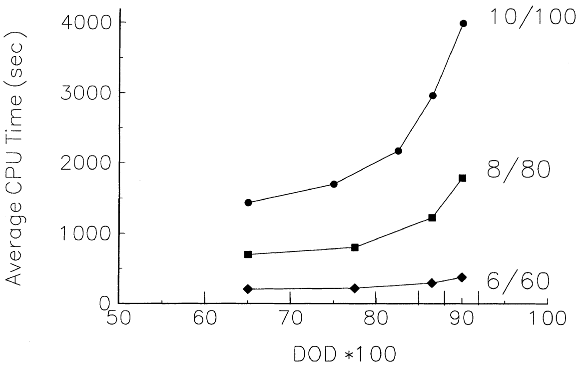

Fig. 7. Average CPU times.

A similar approach to analyse the solution quality cannot be applied successfully to the larger problem classes because the SA solutions (found within the CPU times listed) are in general better than the best solution found by XPRESS-MP in 24 hours on a much faster computer (233 MHz Pentium II). For example, the SA solutions of the hard 6/60-problem subclass are on average 3.45% better than the XPRESS-MP solutions while this difference grows to 6.51% and 6.75% for the hard 8/80- and 10/100-problem subclasses, respectively. For some of these hard problems, XPRESS-MP even fails to find a feasible solution within two days.

Instead, different lower bounds are calculated either by XPRESS-MP or by the column generation heuristic of Cattrysse et al. (1993) for the item-availability DLSP with setup times. While the lower-bound computation using XPRESS-MP is quite time consuming, the latter solution method yields lower bounds in relatively short CPU times. Note, that these bounds also pertain to the batch-availability DLSP because every solution for the situation of batch availability is also a feasible solution for the same problem data under item availability. Moreover, the item-availability bounds for the hard 6/60 (8/80, 10/100) subclass are on average 2.24% (8.10%, 12.95%) larger than the batch-availability lower bounds obtained by XPRESS-MP. Hence, the average relative deviation of the SA solutions from the item-availability lower bounds are given in the fourth column of Tables 5–7.

At first glance, the average size of these gaps between the SA solutions and the lower bounds may appear to be large. However, it should be noted that a solution for the item-availability DLSP is in general not feasible for the case of batch-availability. Hence, the computed lower bounds cannot be expected to be sharp for the latter case. The production of some batch may typically extend over a demand instance in a solution for the item-availability case. This batch then either has to be split into two batches causing an additional setup or has to be shifted to earlier periods which incurs higher inventory holding costs. For example, consider the easy 3/30-problem with the usual parameters and the demand structure given in Table 8. Optimal solutions for the item- and batch-availability cases are given in Figs. 8 and 9 with objective function values of 413 and 506, respectively. Note that the item-availability solution has to be adjusted for the case of batch availability in both ways: the item-1 batch is split up into two parts which yields extra costs for the additional setup (60) while there are positive (30) and negative (15) effects on the inventory holding costs of item 1 due to the shifting of the two resulting batches. In contrast, the batch of item 2 is shifted by six periods to an earlier setup and production with additional costs of 48. These

Table 8
Explicit demand structure for a 3/30 example problem

```
                1         2         3
  | I |123456789012345678901234567890|
--|---|------------------------------|-
1 | 1 |         3         3         2|
2 | 3 |         2         3         3|
3 | 3 |         1         0         2|
__|___|______________________________|_
```

```
                1         2         3
  | I |123456789012345678901234567890|
--|---|------------------------------|-
1 | 1 |       R11111111              |
2 | 3 |                 R11111111    |
3 | 3 |                          R111|
__|___|______________________________|_
```

Fig. 8. Optimal solution of the 3/30 example problem in the case of item availability.

```
                1         2         3
  | I |123456789012345678901234567890|
--|---|------------------------------|-
1 | 1 |    R11111            R111    |
2 | 3 |           R11111111          |
3 | 3 |                          R111|
__|___|______________________________|_
```

Fig. 9. Optimal solution of the 3/30 example problem in the case of batch availability.

structural changes explain the cost difference of $60 - 30 + 15 + 48 = 93$. This example with a 22.5% gap between the optimal solutions for the item- and batch-availability case was chosen deliberately to demonstrate in a small instance all possible changes needed to convert optimal solutions. The average size of the gap between the optimal solutions for the two cases (taken over 75 3/30-instances) is 12.53% with a standard deviation of 4.58% and contributes substantially to the average deviations of the SA solutions from item-availability lower bounds.

The structural differences between the optimal solutions in the cases of item and batch availability as shown in the example will occur more frequently in larger problems. Hence, it can be assumed that the difference of the associated cost values will approximately grow proportionally in problem size. The average deviations of the SA solutions from the item-availability lower bounds remain at a relatively constant level over all problem sizes and the average CV for the objective function values are at most 5% in all cases (fifth column of Tables 5–7). Note that the CVs are computed here for the ten different trials of the stochastic solution procedure and averages are taken over all 15 instances in a DOD class. Hence, it can be concluded that the SA solutions must be of high quality and that the cooling schedule is selected appropriately, i.e., that the results are stable.

The two-phase solution procedure is apparently sensitive to higher demands at the beginning of the planning period. Due to this sensitivity there are problem instances with DODs in the 85–92% interval where the algorithm is unable to come up with a feasible solution in some or even all of the ten different trials. The number of such unsuccessful trials and the number of unsolved instances, i.e., no solution was found in ten trials, are given in the last two columns of Tables 5–7. We investigate this phenomenon for the 6/60-problem in more detail. A further natural subdivision of the 88–92% DOD class is found into instances with an average computation time similar to the easier problems and with a feasible solution found in every trial. On the other hand, there are problems which exceed 500 s of required CPU time. In this latter subset only, it happens that phase 1 of the algorithm is unable to find a feasible solution on several occasions. This may be due to the fact that the set of feasible solutions is in general smaller for a high demand density towards the beginning of the planning period. This hypothesis is supported by small coefficients of variation of the final objective function value. High DOD in combination with high initial demand density is characteristic for the hard subclass. Note for example that there are only five periods left for additional setup for a 6/60 demand schedule with 49 units demanded (91.7% DOD). To satisfy early demand exceeding the initial inventory for several items, small batches have to be produced at the beginning. Hence, the free periods are consumed early and later production is thereby restricted to larger and fewer batches which might not meet the demand. Thus, the DOD does not encompass all dimensions of actual difficulty of the feasibility problem. Since the number of successful trials becomes small in this subset, the corresponding results are not listed in Table 5.

6. Summary and conclusions

The problem presented here differs from the item-availability DLSP given for example by Salomon (1991, Sections 2.5.4 and 5.4) in considering the case of batch availability, not necessarily vanishing initial and target inventories, and in using different time scales for production scheduling and demand. The complexity analysis for different variants of the DLSP by Salomon et al. (1991) is incorrect. In Section 3, we give a positive proof for the NP-hardness in the strong sense of the DLSP. This result also pertains to the case of batch availability. Therefore, a two-phase SA solution heuristic is suggested. The optimization procedure is separated into phase 1 searching for a feasible solution and optimizing cost in phase 2. Production schedules are generated by dividing, combining and shifting batches. The solution procedure is readily applied to larger instances of this difficult problem where plausible solutions are obtained within reasonable computation times by a problem-size dependent choice of the cooling schedule. In contrast to the prohibitive computation times required for the exact solution, our SA approach yields results quickly enough to be employed in the future, for example, to carry out a numerical sensitivity analysis of the production schedules to changes in the problem parameters.

Acknowledgements

The authors would like to thank two anonymous referees for their helpful comments, especially for suggesting a more intuitive formulation of the batch-availability constraints (8)–(11) in Section 4.1. Bob Daniel from Dash Associates, besides supplying Release 10 of XPRESS-MP, made a valuable hint by Laurence Wolsey of CORE with respect to instance-dependent additional constraints available to us. This enabled us to solve a whole set of DLSP test instances and to compute lower bounds for the remaining problems in the case of batch availability where all other general-purpose solvers we had tried before failed. Moreover, we would also like to thank Dirk Cattrysse from the Centre for Industrial Management, Research Group Quantitative Techniques, Heverlee, Belgium, to let us use his program for the item-availability DLSP and

Carsten Jordan, KPMG, Berlin, Germany, for making available to us a working copy of this program.

References

Banker, R.D., Datar, S.M., Kekre, S., 1988. Relevant costs, congestion and stochasticity in production environments. Journal of Accounting and Economics 10, 171–197.

Brüggemann, W., 1995. Ausgewählte Probleme der Produktionsplanung – Modellierung, Komplexität und neuere Lösungsmöglichkeiten, Physica-Verlag, Heidelberg.

Brüggemann, W., Fischer, K., Jahnke, H., 1999. Problems – Models – Complexity, in preparation.

Brüggemann, W., Jahnke, H., 1993. DLSP for multi-item batch production. In: Fandel, G., Gulledge, Th., Jones, A. (Eds.), Operations Research in Production Planning and Control, Proceedings of a Joint German/US Conference, Hagen, 25–26 June, 1992, pp. 459–472.

Brüggemann, W., Jahnke, H., 1997. Remarks on: "Some extensions of the discrete lotsizing and scheduling problem". Management Science 43, 122.

Cattrysse, D., Salomon, M., Kuik, R., Van Wassenhove, L.N., 1993. A dual ascent and column generation heuristic for the discrete lotsizing and scheduling problem with setup times. Management Science 39, 477–486.

Chen, W.-H., Thizy, J.-M., 1990. Analysis of relaxations for the multi-item capacitated lot-sizing problem. In: Queyranne, M., Mazzola, J.B. (Eds.). Production Planning and Scheduling, Annals of Operations Research 26, pp. 29–72.

Černý, V., 1985. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. Journal of Optimization Theory and Application 45, 41–51.

Collins, N.E., Eglese, R.W., Golden, B.L., 1988. Simulated annealing – an annotated bibliography. American Journal of Mathematical and Management Sciences 8, 209–307.

Dinkelbach, W. 1964. Zum Problem der Produktionsplanung in Ein- und Mehrproduktunternehmen, Würzburg, Wien, 1964.

Dobson, G., Karmarkar, U.S., Rummel, J.L., 1987. Batching to minimize flow times on one machine. Management Science 33, 784–799.

Drexl, A., Haase, K., 1992. A new type of model for multi-item capacitated dynamic lotsizing and scheduling, Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel Nr. 286, Kiel.

Fleischmann, B., 1990. The discrete lot-sizing and scheduling problem. European Journal of Operational Research 44, 337–348.

Fleischmann, B., 1994. The discrete lot-sizing and scheduling problem with sequence-dependent setup costs. European Journal of Operational Research 75, 395–404.

Fleischmann, B., Popp, T., 1989. Das dynamische Losgrößenproblem mit reihenfolgeabhängigen Rüstkosten, In: Operations Research Proceedings 1988, Heidelberg, pp. 510–515.

Garey, M.R., Johnson, D.S., 1978. Strong NP-completeness results: Motivation, examples, and implications. Journal of the Association for Computing Machinery 25, 499–508.

Garey, M.R., Johnson, D.S., 1979. Computers and Intractability – A Guide to the Theory of NP-Completeness, New York.

Glassey, C.R., 1968. Minimum change-over scheduling of several products on one machine. Operations Research 16, 342–352.

Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P., 1983. Optimization by simulated annealing. Science 220, 671–680.

Kroon, L.G., Salomon, M., 1988. On the complexity of the discrete lot sizing and scheduling problem, Management Report Series 26, Erasmus Universiteit, Rotterdam.

Lasdon, L.S., Terjung, R.C., 1971. An efficient algorithm for multi-item scheduling. Operations Research 19, 946–969.

Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E., 1953. Equation of state calculations by fast computing machines. The Journal of Chemical Physics 21, 1087–1092.

Potts, C.N., Van Wassenhove, L.N., 1992. Integrating scheduling with batching and lot-sizing: A review of algorithms and complexity. Journal of the Operational Research Society 43, 395–406.

Salomon, M. 1991. Deterministic Lotsizing Models for Production Planning, Berlin.

Salomon, M., Kroon, L.G., Kuik, R., Van Wassenhove, L.N., 1991. Some extensions of the discrete lotsizing and scheduling problem. Management Science 37, 801–812.

Salomon, M., Solomon, M.M., Van Wassenhove, L.N., Dumas, Y., Dauzère-Pérès, S., 1997. Solving the discrete lotsizing and scheduling problem with sequence dependent set-up costs and set-up times using the Travelling Salesman Problem with time windows. European Journal of Operational Research 100, 494–513.

Santos, C., Magazine, M., 1985. Batching in single operation manufacturing systems. Operations Research Letters 4, 99–103.

Schrage, L. 1981. The multiproduct lot scheduling problem. In: Dempster, M.A.H., Lenstra, J.K., Rinnooy Kan, A.H.G. (Eds.), Deterministic and Stochastic Scheduling, Kluwer, Dordrecht, pp. 233–244.

Silver, E.A., Pyke, D.F., Peterson, R., 1988. Inventory Management and Production Planning and Scheduling, 3rd ed., New York.

Van Hoesel, S., Kuik, R., Salomon, M., Van Wassenhove, L.N., 1994. The single-item discrete lotsizing and scheduling problem: optimization by linear and dynamic programming. Discrete Applied Mathematics 48, 289–303.

Van Laarhoven, P.J.M. 1988. Theoretical and Computational Aspects of Simulated Annealing, Ph.D. thesis, Erasmus Universiteit, Rotterdam.

Van Laarhoven, P.J.M., Aarts, E., Lenstra, J., 1992. Job shop scheduling by simulated annealing. Operations Research 40, 113–125.
Webster, S., Baker, K.R., 1995. Scheduling groups of jobs on a single machine. Operations Research 43, 692–703.
Webster, S. 1999. Remarks on: "Some extensions of the discrete lotsizing and scheduling problem", accepted for publication in Management Science 1999.

Update

European Journal of Operational Research
Volume 126, Issue 3, 1 November 2000, Page 688

DOI: https://doi.org/10.1016/S0377-2217(00)00173-9

ELSEVIER

European Journal of Operational Research 126 (2000) 688

EUROPEAN JOURNAL OF OPERATIONAL RESEARCH

www.elsevier.com/locate/dsw

Erratum

Erratum to 'The discrete lot-sizing and scheduling problem: Complexity and modification for batch availability' [EJOR 124 (3) (2000) 511–528] ☆

Wolfgang Brueggemann [a,*], Hermann Jahnke [b]

[a] *Institut für Logistik und Transport, Universität Hamburg, Von-Melle-Park 5, 20146 Hamburg, Germany*

[b] *Fakultät für Wirtschaftswissenschaften, Universität Bielefeld, Universitätstrasse 25, Postfach 10 01 31, 33501 Bielefeld, Germany*

The Publisher regrets that some errors have occurred in the article.

Formula (4) should read:

$$\sum_{i=1}^{N} y_{i,t} + \sum_{\substack{i=1 \\ a_i > 0}}^{N} v_{i,t} \leqslant 1, \quad t = 1, \ldots, T.$$

H. Jahnke's e-mail address should read: h.jahnke@wiwi.uni-bielefeld.de

On page 512, right-hand column, line 11, the reference should read: "(e.g., Silver et al., 1998, pp. 602–610)".

The authors regret that they inadvertently overlooked to discuss and to cite the work by Jordan (1996) and Jordan and Drexl (1998).

References

Jordan, C., 1996. Batching and Scheduling – Models and Methods for Several Problem Classes. Lecture Notes in Economics and Mathematical Systems No. 437. Springer-Verlag, Berlin.

Jordan, C., Drexl, A., 1998. Discrete lotsizing and scheduling by batch sequencing. Management Science 44, 698–713.

Silver, E.A., Pyke, D.F., Peterson, R., 1998. Inventory Management and Production Planning and Scheduling, third ed. Wiley, New York.

☆ PII of the original article: S0377-2217(99)00190-3.

* Corresponding author. Tel.: +49-40-42838-2950; fax: +49-40-42838-6283.

E-mail addresses: brueggemann@na-net.ornl.gov (W. Brueggemann), h.jahnke@wiwi.uni-bielefeld.de (H. Jahnke).

PII: S0377-2217(00)00173-9