INT. J. PROD. RES., 1994, VOL. 32, NO. 4, 755–768

DLSP for two-stage multi-item batch production

W. BRÜGGEMANN and H. JAHNKE†

The standard mixed-integer linear model formulation for the multi-item discrete lot-sizing and scheduling problem (DLSP) is extended by additional partially non-linear constraints for the case of two-stage batch production. The corresponding feasibility problem is NP-complete in the case of non-zero setup times. A simulated annealing approach is suggested for computing production schedules on both stages. Numerical results are presented.

1. Introduction

In general, manufacturing is a multi-level process, where coordinating the different stages requires additional effort compared to single-stage problems. The multi-level lot-sizing problem deals with choosing cost-optimal lot-sizes in the case of uncapacitated (MLLP) or capacitated (MLCLP) production facilities. For the MLLP, some research on serial and assembly product structure is reviewed by Salomon (1991). The MLCLP is known to be NP-hard even for a single product (Chen and Thizy 1990). Salomon (1991) reviews briefly the literature on the MLCLP and suggests heuristics for the single bottleneck case. Maes *et al.* (1991) present heuristics for the MLCLP with capacity constraints on more than one production stage. However, these approaches do not consider the sequencing of the batches within a production period. In contrast, the discrete lot-sizing and scheduling problem (DLSP) is a deterministic production planning model for the simultaneous cost-optimal choice of the lot-sizes and the sequence of jobs in the multi-item case. The principal idea of the DLSP is to divide the finite time horizon into (small) time intervals in which the machines can be used either for production of at most a single product, or can be set up for such a production. By considering only finite production speeds, the manufacturing capacity is limited. A comprehensive reference to the DLSP is given by Salomon (1991). Drexl and Haase (1992) explore the similarities of different model types suggested for the lot-sizing and scheduling problem.

The standard DLSP pertains to the case of instantaneous availability of the manufactured units prior to completion of the lot. But in practice units may arrive in inventory in one batch no earlier than completion of the whole lot. The DLSP is modified for this case of batch production by Brüggemann and Jahnke (1993). Salomon (1991) shows that generating feasible solutions for the standard DLSP with non-vanishing setup times is NP-complete, which is also true for the version modified for batch production. Especially in this latter case, exact algorithms will be time-consuming, if they succeed in finding a solution at all. However, in practical applications it is desirable to study the sensitivity of the optimal schedules to changes in the cost parameters which are often not known precisely. Moreover, quick responses to changing data are needed in applied production planning. These tasks can only be accomplished by fast heuristics even though the solutions proposed might be suboptimal.

† Institut für Logistik und Transport, Universität Hamburg, Von-Melle-Park 5, 20146 Hamburg, Germany.

The coordination of production in the multi-level case can be carried out by storing a sufficient amount of inter-stage inventory or by synchronizing the stage-individual production schedules. Increasing inter-stage inventory yields an enlarged degree of independent action on each stage at the price of higher holding costs. In the case of batch production, the restrictions of upstream requirement for the following manufacturing stage are even more prominent, since final-stage production may start only after components become available by completion of the whole batch on the preprocessing stage. In order to model the case of two-stage batch production for a single machine on each stage, we introduce additional constraints in § 2. Note again that finite production speeds imply limited capacity on each stage. The resulting extended DLSP is formulated as a mixed-integer, partially non-linear program. It can easily be concluded from the single-stage case that this problem is also NP-complete. For solving this problem, we suggest a heuristic based on simulated-annealing (SA) because of its general applicability (§ 3). Numerical results are presented in § 4.

2. DLSP in the case of two-stage batch production

In the following, we assume that $N^{(2)}$ products are manufactured on the final production stage 2, each using a certain number out of the $N^{(1)}$ components from the preprocessing stage 1 for which there is no (external) demand. In order to model this two-stage manufacturing, constraints are introduced in the subsequent section describing demand schedule and batch production on each stage separately. We assume that capacity utilization can be planned independently for production on each stage. This part of the model is developed from the single-stage case in Brüggemann and Jahnke (1993) whereas the interdependencies between the two production stages are modelled in § 2.2. Here, the requirements (internal demand) for components are generated from the current production schedule of the final stage.

2.1. *Stage-individual batch production*

External demand is given for the products of the final stage only while the requirements for the preceding stage are variable and depend on stage-2 decisions. Therefore, the following first subset of the constraints is explained in terms of the independent stage 2 in direct analogue to the single stage DLSP.

Ordinarily, DLSP is formulated for $N^{(2)}$ 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 for example as demand per day, while the DLSP requires a less coarse discretization of the time axis due to the underlying idea that during one time unit (e.g. hours or 30-min intervals), the machine can be used at most for setup or production of only one item (see constraint (4) below). Hence, the time units for DLSP will be chosen to be the greatest common divisor of setup times and minimal production times for all items and stages. The different time scales of demand and scheduling yield a division of the planning period T into M (demand) subintervals of lengths $T_m - T_{m-1}$ ($m=1,\dots,M$; $T_0=0$; $T_M=T$) where external demand $d_{it}^{(2)}$ for product i is positive only at times T_m($d_{iT_m}^{(2)} \geqslant 0$ and $d_{it}^{(2)}=0$ for $t=1,\dots,T$; $t \neq T_m$; $m=1,\dots,M$; $i=1,\dots,N^{(2)}$).

A production schedule for either stage ($j=1,2$) is a matrix $(v^{(j)}, y^{(j)}) = ((v_{it}^{(j)}), (y_{it}^{(j)}))$ where

- $v_{it}^{(j)}$ are zero-one variables indicating setup in period t for production of product i on stage j and

$y_{it}^{(j)}$ are zero-one variables indicating production of product i in period t on stage j with $y_{iT+1}^{(j)}=0$ for formal reasons.

Then the objective is to minimize the sum of the stage-individual cost contributions, which consist of the total of setup and inventory holding costs,

$$\operatorname{Min} C((v^{(1)},y^{(1)}),(v^{(2)},y^{(2)}))=C^{(1)}((v^{(1)},y^{(1)}))+C^{(2)}((v^{(2)},y^{(2)}))$$

$$=\sum_{j=1}^{2}\sum_{i=1}^{N^{(j)}}\sum_{t=1}^{T} r_i^{(j)}v_{it}^{(j)}+h_i^{(j)}I_{it}^{(j)} \qquad (1)$$

subject to

$$I_{it-1}^{(j)}+o_i^{(j)}y_{it}^{(j)}-d_{it}^{(j)}=I_{it}^{(j)} \qquad j=1,2;i=1,\ldots,N^{(j)};t=1,\ldots,T \qquad (2)$$

$$v_{it-a_i^{(j)}+\tau}^{(j)}\geqslant y_{it}^{(j)}-y_{it-1}^{(j)} \qquad j=1,2;i=1,\ldots,N^{(j)};t=a_i^{(j)}+1,\ldots,T;\tau=0,\ldots,a_i^{(j)}-1 \qquad (3)$$

$$y_{it}^{(j)}=0 \qquad j=1,2;i=1,\ldots,N^{(j)};t=1,\ldots,a_i^{(j)} \qquad (3\,a)$$

$$\sum_{i=1}^{N^{(j)}}(y_{it}^{(j)}+v_{it}^{(j)})\leqslant 1 \qquad j=1,2;t=1,\ldots,T. \qquad (4)$$

The objective (1) and constraints (2), (3) and (4) are similar to the standard DLSP (Salomon 1991, pp. 37 and 43). For each stage, $r_i^{(j)}$ in (1) are sequence independent setup costs per setup period and the inventory holding costs are given by the product of parameter $h_i^{(j)}$ (costs per unit and period) and non-negative inventories $I_{it}^{(j)}$. The demand-satisfaction constraints (2) of the standard DLSP describe the dependence of current inventories on the inventories of the preceding period, the quantity of the corresponding item produced (where $o_i^{(j)}$ is the production speed), and the demand or requirement, respectively. The correct sequence of setup and production periods is modelled in equations (3) where $a_i^{(j)}>0$ is the corresponding number of setup periods. Constraints (3 *a*) prohibit production of item i in periods $1,\ldots,a_i^{(j)}$ with no preceding setup. Constraints (4) are used to prevent simultaneous action (setup or production) on the machines.

These constraints alone do not pertain to the case of batch production. While the inventory including work in process $I_{it}^{(j)}$ (as described above) is required for computing holding costs in (1), 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. Therefore, we introduce inventory $\tilde{I}_{im}^{(2)}$ for stage 2 which is already available in demand instance T_m.

$$e_{it}^{(j)}\geqslant y_{it}^{(j)}-y_{it+1}^{(j)} \qquad j=1,2;i=1,\ldots,N^{(j)};t=1,\ldots,T \qquad (5)$$

$$e_{it}^{(j)}\leqslant y_{it}^{(j)} \qquad j=1,2;i=1,\ldots,N^{(j)};t=1,\ldots,T \qquad (6)$$

$$e_{it}^{(j)}\leqslant 1-y_{it+1}^{(j)} \qquad j=1,2;i=1,\ldots,N^{(j)};t=1,\ldots,T \qquad (7)$$

$$\tilde{y}_{itm}^{(2)}\leqslant y_{it}^{(2)} \qquad i=1,\ldots,N^{(2)};m=1,\ldots,M;t=1,\ldots,T_m \qquad (8)$$

$$\tilde{y}_{itm}^{(2)}\leqslant\left(\sum_{\tau=t+1}^{T_m} v_{i\tau}^{(2)}\right)$$
$$+\left(a_i^{(2)}-\sum_{\tau=1}^{T_m}(v_{i\tau}^{(2)}-a_i^{(2)}e_{i\tau}^{(2)})\right) \qquad i=1,\ldots,N^{(2)};m=1,\ldots,M;t=1,\ldots,T_m \qquad (9)$$

$$\tilde{I}_{im}^{(2)}=\sum_{\tau=1}^{T_m}(o_i^{(2)}\tilde{y}_{i\tau m}^{(2)}-d_{i\tau}^{(2)})+I_{i0}^{(2)} \qquad i=1,\ldots,N^{(2)};m=1,\ldots,M \qquad (10)$$

$$e_{it}^{(j)}, I_{it}^{(j)}, I_{i0}^{(j)}, \tilde{I}_{im}^{(2)} \geqslant 0 \qquad j=1,2; i=1,\ldots,N^{(j)}; t=1,\ldots,T;\ m=1,\ldots,M \qquad (11)$$

$$y_{iT+1}^{(j)}=0, v_{it}^{(j)}, y_{it}^{(j)}, \tilde{y}_{itm}^{(2)} \in \{0,1\} \qquad j=1,2; i=1,\ldots,N^{(j)}; t=1,\ldots,T; m=1,\ldots,M. \qquad (12)$$

In order to model this available inventory, constraints (5)–(10) and some auxiliary variables are needed: here, zero-one end-of-batch variables $e_{it}^{(j)}$ indicate the production of the last unit of the current batch in t by $e_{it}^{(j)}=1$. End-of batch constraints (5)–(7) are similar to setup constraints (3). It should be pointed out that $e_{it}^{(j)}$ assumes only a value of zero or one by construction without explicit binary constraints. Additionally, zero-one production variables of available units of final products $\tilde{y}_{itm}^{(2)}$ are introduced in (8) and (9) which resemble the decision variables $y_{it}^{(2)}$. It should be noted that $\tilde{y}_{itm}^{(2)}$ vanishes even if the decision variable $y_{it}^{(2)}$ is equal to one, if period t is used for production of item i in a batch which was started before but has not been completed by T_m. Hence, $\tilde{y}_{itm}^{(2)}$ may only assume a value of 1 if item i is produced in period t and the corresponding batch has been finished by T_m. This behaviour of the auxiliary variables $\tilde{y}_{itm}^{(2)}$ is guaranteed by constraints (8) and (9), where the latter consists of two parts, both of which are zero in the case of incomplete production at time T_m: in this case, the first bracket (the difference between cumulated number of setup periods before T_m and before $t \leqslant T_m$) is greater than zero for all batches that are completed before T_m, while the second bracket is positive only in the case of no production extending in time over T_m (for more details, see Brüggemann and Jahnke 1993). Moreover, $\tilde{y}_{itm}^{(2)}$ may also vanish despite completion of the batch by T_m, if not all of the units produced are needed to satisfy demand in T_m. Note, that the auxiliary inventory of available units $\tilde{I}_{im}^{(2)}$, which is defined using the production variables $\tilde{y}_{itm}^{(2)}$ in (10), will be smaller than the actually available inventory only in this case. Constraints (11) and (12) are the standard non-negativity and binary constraints, respectively, for the decision and auxiliary variables. Initial inventories of stage 2 can be thought of as given constants while final inventories are implicitly included in the demand of period T.

2.2. *Coupling preprocessing and final stage*

The two stages of final-units and component production are joint by the internal demand on the preprocessing stage 1 which is generated depending on the actual stage 2 production schedule. In contrast to the given demand instances T_m for the final products, components are required whenever the final stage is set up for manufacturing. These requirement instances are indicated by variables z_{kt} defined in constraints (13)–(16).

$$z_{kt} \geqslant v_{kt}^{(2)} - v_{kt-1}^{(2)} \qquad k=1,\ldots,N^{(2)}; t=1,\ldots,T \qquad (13)$$

$$z_{kt} \leqslant 1 - v_{kt-1}^{(2)} \qquad k=1,\ldots,N^{(2)}; t=1,\ldots,T \qquad (14)$$

$$z_{kt} \leqslant v_{kt}^{(2)} \qquad k=1,\ldots,N^{(2)}; t=1,\ldots,T \qquad (15)$$

$$z_{kt} \geqslant 0 \qquad k=1,\ldots,N^{(2)}; t=1,\ldots,T \qquad (16)$$

The z_{kt} are zero-one variables similar to setup and end-of-batch variables, where $z_{kt}=1$ means the beginning of setup for production of item k in period t on stage 2. Internal demand for components $d_{it}^{(1)}$ is positive only in the requirement instances and depends on the size of the current batch on stage 2. More specifically, the requirement of component i is calculated in such instances as the production coefficients p_{ki} multiplied by the batch size of the final product which is set up in period t

$$d_{it}^{(1)} = \sum_{k=1}^{N^{(2)}} z_{kt} p_{ki} \sum_{\tau=t}^{T} y_{k\tau}^{(2)} \left[1 + \sum_{\tau'=1}^{t} e_{k\tau'}^{(2)} - \sum_{\tau'=1}^{\tau} e_{k\tau'}^{(2)} \right]^{+} \qquad i=1,\ldots,N^{(1)}; t=1,\ldots,T \qquad (17)$$

$$d_{it}^{(1)} \geqslant 0 \qquad i=1,\ldots,N^{(1)}; t=1,\ldots,T, \tag{18}$$

where $[\cdot]^+ := \max\{0, \cdot\}$ is the maximum operator. The batch size is computed as the second sum in (17). Here, the maximum operator term equals one until setup and manufacturing of the current batch have been completed, and vanishes afterwards. Setup periods are excluded by multiplying the zero-one result of the maximum operator by the production variables $y_{k\tau}^{(2)}$. The correct choice of the actually manufactured final product is ensured by the multiplication of the batch size by the z_{kt} variables. Note the non linearity of constraints (17) because of the product of decision variables and the use of the maximum operator.

It should be emphasized that the variable requirement instances make up the main difference between preprocessing stage and final stage. Besides these variable requirement instances, constraints (19)–(21) for actually available component inventory are very much alike the corresponding stage 2 constraints (8)–(10):

$$\tilde{y}_{its}^{(1)} \leqslant y_{it}^{(1)} \qquad i=1,\ldots,N^{(1)}; s=1,\ldots,T; t=1,\ldots,s \tag{19}$$

$$\tilde{y}_{its}^{(1)} \leqslant \left(\sum_{\tau=t+1}^{s} v_{i\tau}^{(1)} \right) + \left(a_i^{(1)} - \sum_{\tau=1}^{s} (v_{i\tau}^{(1)} - a_i^{(1)} e_{i\tau}^{(1)}) \right) \qquad i=1,\ldots,N^{(1)}; s=1,\ldots,T; t=1,\ldots,s \tag{20}$$

$$\tilde{I}_{is}^{(1)} = \sum_{\tau=1}^{s} \left(o_i^{(1)} \tilde{y}_{i\tau s}^{(1)} - d_{i\tau}^{(1)} \right) + I_{i0}^{(1)} \qquad i=1,\ldots,N^{(1)}; s=1,\ldots,T \tag{21}$$

$$\tilde{I}_{is}^{(1)} \geqslant 0, \tilde{y}_{its}^{(1)} \in \{0,1\} \qquad i=1,\ldots,N^{(1)}, s=1,\ldots,T, t=1,\ldots,s. \tag{22}$$

In the case of non-zero setup times, the single stage DLSP for multi-item batch production is of the same complexity as the NP-complete standard DLSP (Brüggemann and Jahnke 1993). The two-stage DLSP is also in NP, because feasibility of any given structure (production schedule) for any problem instance can be checked in polynomial time. Since any instance of the single-stage modified DLSP can be transformed to an instance of the two-stage DLSP in polynomial time, and a feasible production schedule for the two-stage DLSP contains also a feasible solution for the single-stage version, the single-stage DLSP reduces to the two-stage DLSP which is then also NP-complete (Florian *et al.* 1980).

3. A simulated annealing solution procedure for the two stage DLSP

Simulated annealing is a wide-spread heuristic for the solution of combinatorial optimization problems. Van Laarhoven and Aarts (1987) is a general reference on SA. This heuristic has been applied to scheduling and many related problems: Van Laarhoven *et al.* (1992), for example, study an SA application to the job-shop scheduling problem; Matsuo *et al.* (1988) propose an SA based heuristic for the job-shop scheduling problem and the single machine weighted tardiness problem (1989), while Vakharia and Chang (1990) apply simulated annealing to the scheduling problem of a manufacturing cell. An SA approach for the flow-shop problem is discussed for example by Ogbu and Smith (1990). SA is also employed by Kuik and Salomon (1990) for the uncapacitated multi-level lot-sizing problem. Brüggemann and Jahnke (1993) present a two-phase SA solution procedure for the modified single-stage DLSP.

Batch production on each stage and coupling of preprocessing and final production schedule are modelled separately in the previous section. This twofold character is reflected in the proposed algorithm as well: production schedules are generated and optimized for the final stage using the method by Brüggemann and Jahnke (1993). Afterwards, the resulting component requirements are computed. These are used to concretize the requirement satisfaction constraints (17)–(21) for the preprocessing stage. Subject to the requirement data, component manufacturing is optimized using the same solution procedure as for the final stage. Of course, this procedure is sub-optimal, since both stages are treated almost separately, i.e. the algorithm does not consider the common optimization problem.

3.1. *Two-phase simulated annealing*

In order to achieve stage-individual solutions by a simulated annealing based algorithm, a neighbourhood structure is superimposed on the finite but usually large space of feasible configurations or solutions of either stage (in this case production schedules). Given such a current feasible configuration on stage j, $(v^{(j)}_{\mathrm{cur}}, y^{(j)}_{\mathrm{cur}})$, a candidate solution $(v^{(j)}_{\mathrm{can}}, y^{(j)}_{\mathrm{can}})$ 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 $\exp(-\Delta C^{(j)}/\gamma)$ where $\Delta C^{(j)} = C^{(j)}(v^{(j)}_{\mathrm{can}}, y^{(j)}_{can}) - C^{(j)}(v^{(j)}_{\mathrm{cur}}, y^{(j)}_{\mathrm{cur}})$ is the difference of the contributions $C^{(j)}$ to the cost function value of the candidate and the current configurations. γ is a control parameter corresponding to temperature in the original physical analogue. 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 and it can be shown (Mitra *et al.* 1986) that, roughly spoken, the sequence of configurations converges asymptotically to a globally optimal solution, almost surely, if the Markov chains are irreducible.

Solving a specific problem with SA requires determination of a neighbourhood structure and an appropriate 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). Different cooling schedules are discussed for example by Van Laarhoven (1988). On stage j, the neighbourhood structure for a given production schedule $(v^{(j)}, y^{(j)})$ is defined by reducing $(v^{(j)}, y^{(j)})$ to a 'pure' production schedule $y^{(j)\mathrm{pure}}$ by physically eliminating setup periods. An element $(v^{(j)}_{\mathrm{can}}, y^{(j)}_{\mathrm{can}})$ out of the neighbourhood of $(v^{(j)}, y^{(j)})$ is then obtained by exchanging the activities of two arbitrary periods in $y^{(j)\mathrm{pure}}$ and expanding (i.e. inserting setup periods in front of each production batch and shifting later production by the corresponding number of time units) $y^{(j)\mathrm{pure}}_{\mathrm{can}}$ to $(v^{(j)}_{\mathrm{can}}, y^{(j)}_{\mathrm{can}})$. During expansion of $y^{(j)\mathrm{pure}}_{\mathrm{can}}$ to $(v^{(j)}_{\mathrm{can}}, y^{(j)}_{\mathrm{can}})$ 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 $(v^{(j)}, y^{(j)})$.

However, it is important to point out that the neighbourhood choice is usually performed on the set of feasible configurations only. Feasibility in the context of final and preprocessing stages is mainly given by the demand- or requirement-satisfaction under batch production (10) or (21) and setup constraints (3) and (4). On either stage, it is therefore necessary in general 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 is explained in more detail in the following two subsections, while the coupling of the two stages is described in §3.2.

3.1.1. *Phase 1: search for a feasible production plan*

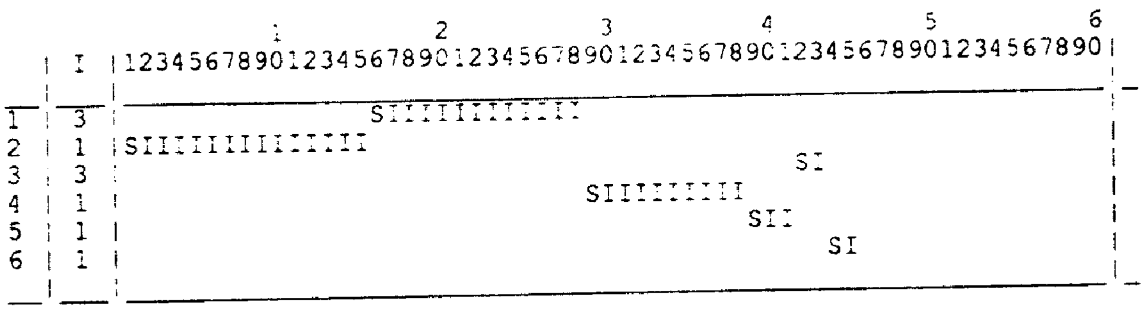
The main ideas of the two-phase algorithm developed from the single-stage case are illustrated more easily for the final stage because of its externally given demand structure. A first possibly infeasible production schedule is chosen to consist of a single batch for each item where the batch size is determined to satisfy the cumulated demand in *T*. Production is carried out as soon as possible in the chronological order of cumulated demand exceeding initial inventory. As an example, the demand schedule for the final stage of a particular problem instance with six products, 60 periods, and six equidistant demand instances is listed in Table 1. Figure 1 shows the corresponding initial production schedule, where S stands for a setup period while production is indicated by I. Each row corresponds to one of the six products while the 60 time periods are represented by the columns. Inventories for the example can be found in the second column I of Fig. 1. For the first perhaps infeasible production schedule, hypothetical inventories are computed which would be needed to fulfil the demand satisfaction constraints (10) 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. 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, neglecting the final idle periods. A first feasible production schedule for the example given in Table 1 can be found in Fig. 2.

3.1.2. *Phase 2: search for a cost-optimal production plan*

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 carried out

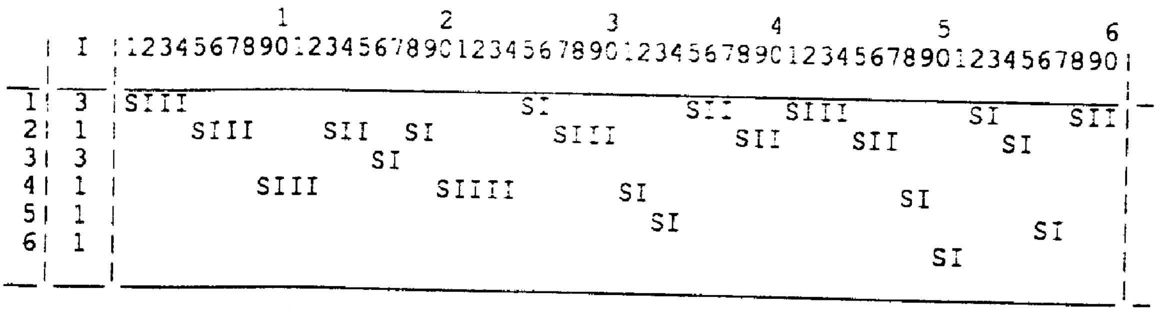
	Demand instances					
Demand product	10	20	30	40	50	60
1	2	4	1	1	2	2
2	3	1	2	3	2	3
3	1	0	0	0	0	0
4	1	3	1	2	2	0
5	0	0	0	1	1	0
6	0	0	0	1	0	0

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



```
                1         2         3         4         5         6
  | I |123456789012345678901234567890123456789012345678901234567890|
1 | 3 |               SIIIIIIIIIIII
2 | 1 |SIIIIIIIIIIIIII
3 | 3 |                                          SI
4 | 1 |                            SIIIIIIIIII
5 | 1 |                                       SII
6 | 1 |                                            SI
```

Figure 1. Initial (infeasible) stage 2 production schedule for the demand of Table 1.



```
               1         2         3         4         5         6
 | I |123456789012345678901234567890123456789012345678901234567890|
-|---|------------------------------------------------------------|-
1| 3 |SIII                    SI        SII   SIII       SI    SII|
2| 1 |    SIII    SII  SI       SIII       SII    SII             |
3| 3 |               SI                                    SI     |
4| 1 |        SIII       SIIII      SI               SI           |
5| 1 |                                SI                          |
6| 1 |                                                 SI    SI   |
-|---|------------------------------------------------------------|-
```

Used Capacity on Stage 2: 100·0%
Stage-2 Cost Contribution: 4050·0

Figure 2. First feasible stage 2 production schedule for the demand of Table 1.

between arbitrary (active and idle) periods of $y^{(j)\,pure}$. Production schedules that are not feasible in the sense of (10) are not considered as candidates. Thus, feasibility is preserved in phase 2 before each acceptance subject to the contribution of the final-stage production schedule 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. 3 for the example and note the improvement of the objective function value where the cost parameters are as defined in §4.

3.2. *Coupling preprocessing and final stage*

Components are required for every batch of final products in its first setup period. Throughout the following, we assume the matrix of production coefficients to be the identity matrix, i.e. there is a 1–1 correspondence between final products and components. Thus, required units of components and requirement instances are easily determined, and can be found in Fig. 4 for the example.

As already mentioned above, the preprocessing stage is subsequently optimized using the same two-phase solution procedure discussed in the preceding section. The resulting production schedule on the preprocessing level is combined with the production plan for stage 2 yielding the common production schedule, which is shown in Fig. 5 for the example, where s and i are used for the preprocessing stage in direct analogue to S and I in the final-stage figures.

The two-phase solution procedure for the single-stage DLSP tends to be sensitive to higher demand at the beginning of the planning period (Brüggemann and Jahnke 1993). In this situation, the Markov chains might even become reducible. In the two-stage case, the search for a feasible solution on the first stage can become still more difficult if large batches occur towards the beginning on the final product level which yields high and early requirements on the preprocessing stage. It may even happen that there is no feasible solution due to instantaneous requirement of more units than are stocked. In contrast to the first feasible solutions, optimized production schedules reveal the – in this context – disadvantageous structure of fewer and larger batches more often. Since the difference between two adjacent configurations consists in the exchange of two activities only, the solution structure will change almost continuously from high entropy configurations with many distributed small batches towards the organized arrangement with a high degree of compactness. During the optimization process, it is therefore reasonable to store production schedules with different degrees

```
                  1         2         3         4         5         6
   |  I |123456789012345678901234567890123456789012345678901234567890|
 1 |  3 |         SIIIII                        SIIIIIII             |
 2 |  1 |     SIII             SIIIIIII                         SIIII|
 3 |  3 |                                                     SI     |
 4 |  1 |               SIIII             SIIIII                     |
 5 |  1 |                                               SII          |
 6 |  1 |                                                   SI       |
```

Used Capacity on Stage 2: 81·7%
Stage-2 Cost Contribution: 2468·0

Figure 3. Final phase 2 stage 2 production schedule for the demand of Table 1.

```
             1         2         3         4         5         6
   |123456789012345678901234567890123456789012345678901234567890|
 1 |         5                             7                    |
 2 |     3                7                                4    |
 3 |                                                     1      |
 4 |               4                 5                          |
 5 |                                               2            |
 6 |                                                   1        |
```

Figure 4. Structure of required components for the production schedule in Figure 3.

```
                 1         2         3         4         5         6
   | I |123456789012345678901234567890123456789012345678901234567890
 1 | 2 |      siii                   siiiiiiiiii                     |
 1 | 3 |         SIIIII                        SIIIIIII             |
 2 | 4 |                siiiiii                       siiiiiiii     |
 2 | 1 |     SIII             SIIIIIII                         SIIII|
 3 | 2 |                                         si                 |
 3 | 3 |                                                     SI     |
 4 | 4 |                       siiiii                          siiii|
 4 | 1 |               SIIII             SIIIII                     |
 5 | 2 |                                            sii             |
 5 | 1 |                                               SII          |
 6 | 2 |                                        si                  |
 6 | 1 |                                                   SI       |
```

Used Capacity on Stage 1: 80·0%
Common Objective Function: 4295·0

Figure 5. Common production schedule for preprocessing and final manufacturing stage.

of compactness in order to successively replace the stage 2 production schedule if the algorithm fails to find a feasible solution for the preprocessing stage. In this case, the corresponding requirements are generated from the replacement production schedule (RPS), and phase 1 is triggered again. Thus, a list of selected RPS is compiled for this purpose during phase 2 of stage 2. The acceptance of an RPS depends on two criteria: for a feasible solution of the preprocessing stage to exist, a necessary condition is that at any time the ratio of the capacity, which has been used previously to uphold stage 2 production (i.e. number of production periods including setups on the preprocessing stage for the required amount of components minus inventory), to actually available capacity does not exceed a value of one. As an approximation, this ratio is computed for the first requirement instance of each component. Prior to listing of a production schedule, the maximum of these ratios is checked to be less than $(1-\varepsilon)$, $\varepsilon > 0$. The chance to solve the stage-1 feasibility problem is improved depending on ε, which is paid for by higher stage-2 costs. Among the production schedules that fulfil the condition for the first batches, it is desirable to enlist configurations that exhibit suitable different degrees of compactness identified by their objective function values. From this list, the actually used RPS is chosen subject to minimal increase of costs. However, the list might be empty when the optimization phase for the final stage is finished, and the algorithm might moreover be unable to find a feasible solution on the preprocessing stage, even if the final production schedule reveals an approximative maximum ratio between $(1-\varepsilon)$ and 1. This problem is tackled by an iterative restart of the final-stage phase 2 to explore different solution structures: in the first place, stage 2 optimization is carried out again unchanged. For the second iteration, setup costs are reduced in order to avoid the preference for large batches. If even this approach does not yield a production schedule of sufficiently low compactness, phase 2 search is activated again with the more exhaustive phase-1 cooling schedule (Table 2) and reduced setup costs.

3.3. *Choice of target inventories and cooling schedule*

In many cases, 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 inventories at the end of the planning horizon on both stages. Thus, sensible inventories have to be determined which can be used as target stock for production planning, while actual inventories result from the deviation of realized from estimated demand or requirement. Sensible inventories can be found for example by a parametric application of the two-stage DLSP-model to the expected external demand. These target inventories for the stochastic application will usually contain an additional safety stock component in order to compensate for the deviations of the actual from the estimated external or internal demand. Positive initial inventories yield more feasible production schedules with respect to the demand and requirement restrictions with additional cost saving potential. Because the requirement instances are variable, it might happen that component requirements cannot be met without units stored. Since a parametric application of the model is more time consuming in the two-stage situation, the target inventories of the final stage used for the following numerical experiments are based on the more detailed analysis by Brüggemann and Jahnke (1993) for the single-stage case, while the inter-stage stock (number of components) is chosen to be larger than, and asymmetrical to, the final-

Parameter	Phase 1	Phase 2
Initial temperature	10	100
Reduction	0·94	0·9
Max number of acceptances	$0{\cdot}24 \cdot N^{(j)} \cdot T + 30$	$15 \cdot N^{(j)} - 20$
Max number of repetitions	$19 \cdot N^{(j)} \cdot T + 3000$	$375 \cdot N^{(j)} - 1125$
Number of reductions	80	50

Table 2. Simulated annealing cooling schedules.

the first requirement instance of each component. Prior to listing of a production schedule, the maximum of these ratios is checked to be less than $(1-\varepsilon)$, $\varepsilon > 0$. The chance to solve the stage-1 feasibility problem is improved depending on ε, which is paid for by higher stage-2 costs. Among the production schedules that fulfil the condition for the first batches, it is desirable to enlist configurations that exhibit suitable different degrees of compactness identified by their objective function values. From this list, the actually used RPS is chosen subject to minimal increase of costs. However, the list might be empty when the optimization phase for the final stage is finished, and the algorithm might moreover be unable to find a feasible solution on the preprocessing stage, even if the final production schedule reveals an approximative maximum ratio between (1-ε) and 1. This problem is tackled by an iterative restart of the final-stage phase 2 to explore different solution structures: in the first place, stage 2 optimization is carried out again unchanged. For the second iteration, setup costs are reduced in order to avoid the preference for large batches. If even this approach does not yield a production schedule of sufficiently low compactness, phase 2 search is activated again with the more exhaustive phase-1 cooling schedule (Table 2) and reduced setup costs.

3.3. *Choice of target inventories and cooling schedule*

In many cases, 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 inventories at the end of the planning horizon on both stages. Thus, sensible inventories have to be determined which can be used as target stock for production planning, while actual inventories result from the deviation of realized from estimated demand or requirement. Sensible inventories can be found for example by a parametric application of the two-stage DLSP-model to the expected external demand. These target inventories for the stochastic application will usually contain an additional safety stock component in order to compensate for the deviations of the actual from the estimated external or internal demand. Positive initial inventories yield more feasible production schedules with respect to the demand and requirement restrictions with additional cost saving potential. Because the requirement instances are variable, it might happen that component requirements cannot be met without units stored. Since a parametric application of the model is more time consuming in the two-stage situation, the target inventories of the final stage used for the following numerical experiments are based on the more detailed analysis by Brüggemann and Jahnke (1993) for the single-stage case, while the inter-stage stock (number of components) is chosen to be larger than, and asymmetrical to, the final-

Parameter	Phase 1	Phase 2
Initial temperature	10	100
Reduction	0·94	0·9
Max number of acceptances	$0{\cdot}24 \cdot N^{(j)} \cdot T + 30$	$15 \cdot N^{(j)} - 20$
Max number of repetitions	$19 \cdot N^{(j)} \cdot T + 3000$	$375 \cdot N^{(j)} - 1125$
Number of reductions	80	50

Table 2. Simulated annealing cooling schedules.

product inventories in order to cope with the infeasibility problem in the case of high and early requirements. Inventory holding costs per unit are assumed to be lower on stage 1 compared to the final-product stage. Note that the two stages can be decoupled by increasing the inter-stage inventory.

Since the solution method presented here consists essentially in repeated application of the single-stage algorithm, the cooling schedules are adjusted according to the corresponding problem sizes for each stage individually. The problem size is defined as the number of bits occupied by the data (Florian *et al.* 1980, p. 676), and is given in the case of the single-stage DLSP by inventory, setup times, and demand or requirements. Thus, it is asymptotically proportional to the product of the length of the planning period and the number of products or components. Since the modified single-stage DLSP is NP-complete, the number of trials for each phase and stage determined by the cooling schedule should depend on $N^{(j)} \cdot T$ exponentially in order to maintain the solution quality. However, we do not follow these general considerations here. Since phase 1 of each stage is concerned with finding a feasible solution rather than minimizing cost, 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 stage individual 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 the conducted numerical experiments, a geometric cooling schedule is applied, the number of repetitions is given by 'acceptances max constant' and no acceptance at one temperature stage 'max constant' is used as the stopping criterion (for notation see Collins *et al.* 1988). 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 expected magnitude of change in the objective function and therefore on the data given in §4. The maximal numbers of repetitions and acceptances for phase 1 are chosen proportional to $N^{(j)} \cdot T$ rather than exponentially, while these parameters of the cooling schedule for phase 2 depend linearly on the number of products $N^{(j)}$ only. The cooling schedules used for both stages in the numerical experiments are shown in Table 2. The actual maximal number of repetitions and acceptances are rounded to smooth values.

4. Numerical results

Demand schedules are generated randomly for the numerical evaluation of the two-stage DLSP solution procedure suggested in the preceding paragraphs. We consider a planning horizon of 60 periods and six products which are distinguishable into different categories depending on 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. Again, the matrix of production coefficients is assumed to be the identity matrix. Every tenth period is a demand instance for all final products. Demand for all products and components is measured in how many units can be produced during one production period. Thus, all production speeds are equal to one. Similarly, inventory costs for all components are assumed to be equal and unity, while final products can be stored at a doubled cost rate due to their higher value. A setup of one period is required for each batch and the corresponding costs are set to 120. The inventory used for both stages are given in the second column of Fig. 5.

The solution procedure is programmed in FORTRAN 77 and implemented on an IBM compatible NEC PC with an 80 486/20 MHz processor. 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 for final products plus one setup period per product and the length of the planning period. Note that two problems with the same elementary degree of difficulty (DOD) can be quite dissimilar with respect to actual difficulty due to different demand distributions over time. The demand schedules are drawn randomly from DOD intervals which are specified in Brüggemann and Jahnke (1993) for the single-stage DLSP according to similar computation times of the corresponding problems (Table 3). All DOD intervals contain their left boundary value with the exception of the 0·85–0·88 interval which is open. Problems with a $\geqslant 88\%$ degree of difficulty are considered to be toughest. The highest DOD considered is 0·92 since the set of feasible solution becomes very small beyond this limit (it might even be empty) and hence there is no decision problem of any interest left.

For each DOD interval or subclass, 15 demand schedules are generated by chance according to the demand categories described above. We apply the solution procedure to each of these problem realizations ten times with new seeds for the random number generator. The results are shown in Table 3. It should be emphasized that the CPU times for our solution procedure are obtained by a code which is not optimized for speed.

In the single-stage case it was possible to check the solution quality for a very small example by comparison to the exact solution which was also found by the proposed SA algorithm in five out of 10 repetitions while the other five solutions were near optimal with a low coefficient of variation (CV). Due to the non-linearity in the coupling constraints (17) and the increased intricacy, it is not amenable here to validate the proposed solution procedure by comparison to the exact solution. We still use the coefficient of variation of the final objective function value for ten different realizations as a criterion of stability and quality of the solutions found instead of the true optimum.

Degree of difficulty	Average CV[1] of cost	Average CPU time (s)	CV[1] of average CPU time	Unsuccessful trials	Use of additional devices
0·60–0·70	2·89	356	6·35	—	0/0
0·70–0·80	3·09	351	6·79	—	1/0
0·80–0·85	3·21	400	15·23	—	11/2
0·85–0·88	3·39	492	17·83	2	19/18
0·88–0·92[2]	3·82	670	21·68	—	20/22
0·88–0·92[3]	2·52	1119	48·00	29	36/149

Table 3. Results for the six-product/60-period two-stage DLSP. [1] (Coefficient of variation)·100. [2] Easy subclass. [3] Hard subclass.

In all cases, the coefficients of variation for the objective function values are at most 5%, indicating stable results and an appropriate selection of the cooling schedule. CPU times tend to grow with DOD and are roughly doubled compared to the single-stage case for the easy problem classes (Brüggemann and Jahnke 1993). In particular, phase 1 can require a significant amount of computation time for the more difficult problems on both stages. The two-phase solution procedure for the final stage is sensitive to higher demands at the beginning of the planning period as discussed in Brüggemann and Jahnke (1993) for the single-stage DLSP. The same phenomenon can be observed for phase 1 of the preprocessing stage. Due to this sensitivity there are problem instances with DODs in the 0·88–0·92 interval where the algorithm fails to come up with a feasible solution in some or even all of the ten different trials. The division of the problems in this interval is refined into problems with an average computation time similar to the easier problems, and with a feasible solution found in every trial (easy subclass). On the other hand, there are problems where it happens that the algorithm is unable to locate a feasible solution on several occasions. Two different cases arise: first, no feasible solution is found for the final stage. These demand realizations are not considered any further in the numerical evaluation, since this is exclusively a single-stage problem. Second, the algorithm is not able to match a feasible final-stage solution with a preprocessing production schedule. High DOD in combination with high initial demand density is typically characteristic for the hard subclass. It is important to point out that in this subclass the algorithm relies heavily on the replacement and retriggering devices as can be seen in the last column of Table 3, where the first number indicates how often RPS are used while the second gives the number of final-stage phase-2 restarts. This explains the high average computation times in this subclass and the large coefficient of variation for CPU times of 48%, because the additional devices for finding a feasible solution on the first stages are used with different frequencies. It should be noted, however, that a feasible solution is achieved by these devices in many hard problem instances.

5. Summary and conclusions

The standard DLSP (e.g. Salomon 1991) is extended here for the case of two-stage batch production with not necessarily vanishing inventories and using different time scales for production scheduling and demand. The feasibility problem for this two-stage DLSP with non-vanishing setup times is shown to be again NP-complete. The suggested SA solution method is based on intuitive ideas. The optimization procedure for each stage 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. This approach is carried out for the final stage subject to the externally given demand. The resulting production schedule induces component requirements for the preprocessing stage. Afterwards, the two-phase algorithm computes a stage 1 production schedule fulfilling these requirements. In the case that no feasible solution is found for the preprocessing stage, additional algorithmic devices are suggested. The numerical experiments presented indicate that our heuristic solution method yields stable results in short computation times. In contrast to the probably large computation times required for the exact solution, our SA approach allows the numerical sensitivity analysis of cost parameters which are often not easily estimated in practice. Moreover, the study of the trade off between increased inter-stage inventory holding costs and the reduced degree of independent action on each stage leading usually to higher setup costs is made amenable by the relatively short computation times.

References

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

Chen, W. H., and Thizy, J.-M., 1991, Analysis of relaxations for the multi-item capacitated lot-sizing problem. In *Annals of Operations Research*, **26**, 29–72.

Collins, N. E., Eglese, R. W., and Golden, B. L., 1988, Simulated annealing—an annotated bibliography. *American Journal of Mathematical and Management Sciences*, **8** (3/4), 209–307.

Drexl, A., and 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.

Florian, M., Lenstra, J. K., and Rinnooy Kan, A. H. G., 1980, Deterministic production planning: algorithms and complexity. *Management Science*, **26** (7), 669–679.

Kuick, R., and Salomon, M., 1990, Multi-level lot-sizing problem: evaluation of a simulated-annealing heuristic. *European Journal of Operations Research*, **45**, 25–37.

Maes, J., McClain, J. O., and Van Wassenhove, L. N., 1991, Multilevel capacitated lotsizing complexity and LP-based heuristics. *European Journal of Operations Research*, **53**, 131–148.

Matsuo, H., Suh, C. J., and Sullivan, R. S., 1988, A controlled search simulated annealing method for the general jobshop scheduling problem. *Working Paper No. 03–04-88*, Graduate School of Business, The University of Texas at Austin.

Matsuo, H., Suh, C. J., and Sullivan, R. S., 1989, A controlled search simulated annealing method for the single machine weighted tardiness problem. *Annals of Operations Research*, **21**, 85–108.

Mitra, D., Romeo, F., and Sangiovanni-Vincentelli, A., 1986, Convergence and finite-time behavior of simulated annealing. *Advances in Applied Probability*, **18**, 747–771.

Ogbu, F. A., and Smith, D. K., 1990, The application of the simulated annealing algorithm to the solution of the $n/m/C_{\max}$ flowshop problem. *Computers in Operations Research*, **17** (3), 243-253.

Salomon, M., 1991, *Deterministic Lotsizing Models for Production Planning* (Berlin: Springer Verlag).

Vakharia, A. J., and Chang, Y.-L., 1990, A simulated annealing approach to scheduling a manufacturing cell. *Naval Research Logistics*, **37**, 559–577.

Van Laarhoven, P. J. M., 1988, Theoretical and computational aspects of simulated annealing, PhD thesis, Erasmus Universiteit, Rotterdam.

Van Laarhoven, P. J. M., and Aarts, E. H. L., 1987, *Simulated Annealing: Theory and Applications* (Dordrecht: Kluwer).

Van Laarhoven, P. J. M., Aarts, E. H. L., and Lenstra, J. K., 1992, Job shop scheduling by simulated annealing. *Operations Research*, **40** (1), 113–125.