

SOME EXTENSIONS OF THE DISCRETE LOTSIZING AND SCHEDULING PROBLEM*

MARC SALOMON, LEO G. KROON, ROELOF KUIK AND
LUK N. VAN WASSENHOVE

Erasmus University Rotterdam, P.O. Box 1738, 3000 DR Rotterdam, The Netherlands
Erasmus University Rotterdam, P.O. Box 1738, 3000 DR Rotterdam, The Netherlands
Erasmus University Rotterdam, P.O. Box 1738, 3000 DR Rotterdam, The Netherlands
INSEAD, Boulevard de Constance, F-77305 Fontainebleau Cedex, France

In this paper the Discrete Lotsizing and Scheduling Problem (DLSP) is considered. DLSP relates to capacitated lotsizing as well as to job scheduling problems and is concerned with determining a feasible production schedule with minimal total costs in a single-stage manufacturing process. This involves the sequencing and sizing of production lots for a number of different items over a discrete and finite planning horizon. Feasibility of production schedules is subject to production quantities being within bounds set by capacity.

A problem classification for DLSP is introduced and results on computational complexity are derived for a number of single and parallel machine problems. Furthermore, efficient algorithms are discussed for solving special single and parallel machine variants of DLSP.
(PRODUCTION PLANNING; LOTSIZING; SEQUENCING; COMPUTATIONAL COMPLEXITY)

1. Introduction

The Discrete Lotsizing and Scheduling Problem (DLSP) is the problem of stating the sequence and size of production lots or batches for a number of different items at one or more machines in a single-stage manufacturing process. The time horizon is segmented into a finite number of equal length time periods. There is a capacity constraint for every machine in every period and it is assumed that at most one item can be produced per machine and per period. Demand is stated per time period and the problem is to find a production schedule with minimal costs. The costs considered in DLSP consist of holding costs for inventory, production costs and machine setup costs.

Among the first who contributed to the research on DLSP are Lasdon and Terjung (1971), who used the model in an automated production-scheduling system for a tire company. Schrage (1982) investigated several model extensions for DLSP. An application of DLSP in a hierarchical production planning approach for a chemical plant is described by Van Wassenhove and Vanderhenst (1983). Recently, solution procedures for DLSP have been discussed by Magnanti and Vachani (1987), Fleischmann (1988), Fleischmann and Popp (1988), Kuik et al. (1989), Gavish and Johnson (1990), and Cattrysse et al. (1990).

The DLSP is closely related to the Continuous Setup Lotsizing Problem (CSLP), the well-known Capacitated Lotsizing Problem (CLSP), the Economic Lotsizing and Scheduling Problem (ELSP) and to various Job Scheduling Problems (JSP). Some references which discuss these models in more detail are listed below.

- DLSP versus CSLP: The CSLP differs only slightly from the DLSP as far as the problem statement is concerned. As in DLSP, the finite time axis is subdivided into a number of equal length periods. Both models have the same cost structure and in each period at most one item can be produced. However, the subtle difference between DLSP and CSLP consists herein that in DLSP the quantity produced in each period is either zero or equal to the full production capacity ("all or nothing production") while CSLP

* Accepted by Stephen C. Graves; received April 3, 1989. This paper has been with the authors 5 months for 3 revisions.

allows for any production quantity, which does not exceed available production capacity. Note that there are many practical situations in which "all or nothing production" models are suitable (see e.g. Van Wassenhove and Vanderhenst 1983), especially in production processes in which the production quantities are restricted to be integer multiples of some minimum batch size.

The CSLP was introduced by Bitran and Matsuo (1986) and was later studied by Karmarkar et al. (1987), among others. Recently, de Matta and Guignard (1989) considered the parallel machine CSLP. An advantage of the DLSP formulation over the CSLP formulation lies herein that DLSP algorithms can solve rather large problems in reasonable time (see Fleischmann 1988), while Karmarkar and Schrage (1985) report that computational experiences with large multi-item CSLP problems were rather disappointing.

- **DLSP versus CLSP:** DLSP and CLSP are similar in that both problems allow for time-varying dynamic demand and assume production capacity to be limited per period. However, in DLSP at most one item can be produced in any time period at a machine, whereas in CLSP multiple items may be produced. Furthermore, the cost structure in both problems is different. In CLSP setup costs for an item are incurred in each period that the item is produced, while in DLSP a setup cost is charged only in a period in which a production batch starts, even when the batch runs over several periods. CLSP is mostly used for medium-range production planning problems with relatively large planning periods (months or weeks), while DLSP is used both for medium-range planning and for short-range planning in which periods stand for days, shifts or even hours. An extensive overview of research concerned with CLSP is given by Maes and Van Wassenhove (1988). Complexity results for CLSP are presented by Bitran and Yanasse (1982).

- **DLSP versus ELSP:** Similarities between ELSP and DLSP can be found in the setup cost structure and the capacity restriction on the production output per unit of time. The main difference between ELSP and DLSP is that ELSP assumes time to be continuous, demand for items to be constant and the planning horizon to be infinite, while DLSP assumes discrete time periods, time-varying demand and a finite planning horizon. From these differences it is clear that ELSP applies to a production environment with stable and high volume demand (mass production), while DLSP applies to an environment in which demand may fluctuate substantially. A detailed review on ELSP research is given by Elmaghraby (1978).

- **DLSP versus JSP:** DLSP and JSP are similar in that both problems allow for time-varying demand. However, for DLSP demand per period is aggregated per item, while in the job scheduling problem demand is expressed in terms of jobs with release dates, due dates and processing times. Hence, in the job scheduling problem the batching/unbatching decision is usually not considered as part of the problem. Furthermore, in most job scheduling problems criteria other than setup and holding costs are considered. Exceptions can be found in the work of Monma and Potts (1989), who studied batching/unbatching decisions for jobs, and in the work of Bruno and Downey (1978), who considered job scheduling problems with setup costs and setup times. A final difference between DLSP and JSP consists herein that for DLSP a production run can only start at the beginning of a planning period, while for the JSP a production run (job) can start at any point in time. However, by taking the length of a planning period small enough, DLSP can be used for detailed planning too. An overview of recent JSP research is given by Lawler et al. (1989).

Table 1 summarizes similarities and dissimilarities between DLSP, CSLP, CLSP, ELSP and JSP.

The remainder of this paper is organized as follows: In §2 we present a mathematical model for the generic DLSP, and in §3 we suggest a unified notation to identify various problem types. Results on computational complexity for DLSP are derived in §4. The paper concludes with §5, which discusses algorithms, and summarizes the results.

TABLE 1
A Comparison Between Various Lotsizing Models

	Time Axis	Maximum # of Items Produced Per Period	Setup Cost	Demand	Production Quantity Per Period
DLSP	finite, discrete	1	per batch	dynamic	zero or equal to available capacity
CSLP					less than or equal to available capacity
CLSP	finite, discrete	unrestricted	per period	dynamic	less than or equal to available capacity
ELSP	infinite, continuous	1	per batch	constant	2
JSP			many variants	dynamic	

¹ Maximum production of one item per unit of time.

² Production quantity equal to zero or equal to available capacity per unit of time.

2. Mathematical Formulation of DLSP

The "generic" version of the DLSP was recently suggested by Fleischmann (1988), and can be formulated as the following mixed integer program:

Generic DLSP

$$\min \sum_{i=1}^N \sum_{t=1}^T (S_i \max(0, y_{i,t} - y_{i,t-1}) + h_i I_{i,t} + p_{i,t} y_{i,t}) \quad (1)$$

subject to

$$\sum_{i=1}^N y_{i,t} \leq 1, \quad t = 1, \dots, T. \quad (2)$$

$$I_{i,t-1} + r_i y_{i,t} - d_{i,t} = I_{i,t}, \quad i = 1, \dots, N; \quad t = 1, \dots, T. \quad (3)$$

$$I_{i,t} \geq 0, \quad i = 1, \dots, N; \quad t = 1, \dots, T. \quad (4)$$

$$y_{i,t} \in \{0, 1\}, \quad i = 1, \dots, N; \quad t = 1, \dots, T, \quad (5)$$

where N is the number of items, and T is the number of planning periods. The binary decision variables $y_{i,t}$ indicate production for item i in period t . More precisely, $y_{i,t}$ is equal to one if production for item i takes place in period t and $y_{i,t}$ is equal to zero otherwise. If production for item i takes place in period t , then r_i units are produced. Furthermore, the nonnegative decision variable $I_{i,t}$ represents the on-hand inventory level for item i at the end of planning period t . The initial stock position for item i is given by the predetermined variable $I_{i,0}$.

The objective (minimizing the sum of setup costs, holding costs and production costs) is expressed by (1). To explain the setup cost structure we first define a *batch* of item i as an uninterrupted sequence of periods in which production for item i takes place. Setup costs for item i are incurred in each period t in which a batch starts, which is reflected by the term $S_i \max(0, y_{i,t} - y_{i,t-1})$ in (1). The holding costs and the production costs for item i in period t are expressed by the terms $h_i I_{i,t}$ and $p_{i,t} y_{i,t}$ respectively. The set of equations (2) guarantees that at most one item is produced in a period, while the sets of

equations (3) and (4) assure that the demand $d_{i,t}$ for item i in period t is fulfilled without backlogging. Finally, the binary character of the production variables is expressed by (5), while the initial machine state is specified by the predetermined variables $y_{i,0}$.

The generic DLSP formulation is rather restrictive. Several extensions and generalizations may prove useful in practice. To identify these, we introduce a standard problem notation in the following section.

3. Standard Problem Notation

The standard problem notation we introduce to denote various DLSP problems corresponds to the notation suggested by Gavish and Johnson (1990), but considers in addition some elements, specific for DLSP. Our six-field notation is stated as follows: $L/M/N/SC/PC/ST$.

The specific problem characteristics represented in this notation are discussed below:

- Layout of the production line (*field identifier: L*). In the problems that will be considered in this paper, production takes place at one or more *Parallel* machines. The production rate of each parallel machine is denoted by constants $r_{i,m}$, where i ranges over all item numbers, and m ranges over all machine numbers. The parallel machines are *Identical (PI)*, when the production rate is machine independent, that is, when $r_{i,m}$ is independent of m , for all items i . The parallel machines are *Uniform (PU)*, when the production rate is machine dependent, but differs only by a constant C_m , that is $r_{i,m} = C_m r_i$. In all other cases the machines are called *Unrelated (P)*. When only one machine is considered, this field is omitted.

- The number of machines (*field identifier: M*). If the number of machines is not part of the problem description (as will be assumed at certain points when addressing questions concerning computational complexity), this is denoted by an asterisk (*).

- The number of items (*field identifier: N*). When the number of items is not part of the problem description, this is denoted by an asterisk (*).

- The setup cost structure (*field identifier: SC*). In the generic version of DLSP, the setup costs are assumed to be *Sequence Independent (SI)* implying that the setup costs for a batch of item i , S_i , do not depend on the preceding batch. If, on the other hand, the setup costs for a batch of item i depend on the preceding batch, consisting, say, of item j , then setup costs are *Sequence Dependent (SD)*. Finally, if all setup costs are zero, this is denoted by A (*Absent*) in the six-field notation.

- The production cost structure (*field identifier: PC*). The production costs $p_{i,t}$ may be *Constant* over time (C), that is, for each item i , $p_{i,t}$ does not depend on the period t . In all other cases the production costs are *General (G)*.

- The setup time structure (*field identifier: ST*). The symbol A (absent) denotes zero setup times. In case of sequence independency (SI), the setup time (a_i) for a batch of item i only depends on i and not on preceding batches. If setup times are sequence dependent (SD), a switch from production of item j to production of item i requires a setup time which depends on both j and i .

In Table 2 all possible values for each field in the six-field notation are summarized.

Subsequently, when a particular field takes any fixed value from the range, considered as part of the problem description and not as (variable) input to the problem, the corresponding field identifier will be used explicitly.

To conclude this section, we demonstrate the use of our six-field notation by the following examples:

EXAMPLES.

1. The notation $1/N/SI/G/A$ describes the generic DLSP, formulated in §2. (Note that the first field is left out, as the one-machine case is considered.)

2. The generic DLSP in which the number of items is not fixed beforehand, i.e., is not part of the problem description, is denoted by $1/*/SI/G/A$.

3. The notation $P/*/N/SC/PC/ST$ identifies the unrelated parallel machine problem

TABLE 2
Six-Field Notation for DLSP

Contents	Field	Range
layout of production line ¹	<i>L</i>	Parallel Identical (<i>PI</i>) Parallel Uniform (<i>PU</i>) Parallel Unrelated (<i>P</i>)
number of machines	<i>M</i>	Positive integer or *
number of items	<i>N</i>	Positive integer or *
setup cost structure	<i>SC</i>	Absent (<i>A</i>) Sequence Independent (<i>SI</i>) Sequence Dependent (<i>SD</i>)
production cost structure	<i>PC</i>	Constant (<i>C</i>) General (<i>G</i>)
setup time structure	<i>ST</i>	Absent (<i>A</i>) Sequence Independent (<i>SI</i>) Sequence Dependent (<i>SD</i>)

¹ This field is omitted for one machine problems.

in which the number of machines is not regarded as prespecified, that is, not as part of the problem description.

4. Complexity results for the Single- and Multiple-Machine DLSP

In this section the computational complexity of several DLSP variants is analyzed. We differentiate between feasibility problems, which are considered in §4.1, and optimization problems, to be considered in §4.2. In what follows the *feasibility* problem for a given DLSP variant *L/M/N/SC/PC/ST* is denoted by $(L/M/N/SC/PC/ST)_F$, while the optimization problem is simply denoted by *L/M/N/SC/PC/ST*.

4.1. Feasibility Problems

The feasibility problem is to determine whether a production schedule exists wherein demand for all items is fulfilled without backlogging and capacity constraints are not violated. Subsequently, we analyse the feasibility problem for the one machine case and for parallel machines.

4.1.1. *One-Machine Problems*. Mathematically, the feasibility problem for the one-machine DLSP can be stated as: Does there exist a feasible solution to the following set of equations:

$$\sum_{\tau=1}^t y_{i,\tau} \geq D_{i,t}, \quad i = 1, \dots, N; \quad t = 1, \dots, T-1. \quad (6a)$$

$$\sum_{\tau=1}^T y_{i,\tau} = D_{i,T}, \quad i = 1, \dots, N. \quad (6b)$$

$$\sum_{i=1}^N y_{i,t} + \sum_{i \in \mathcal{A}} v_{i,t} \leq 1, \quad t = 1, \dots, T. \quad (7)$$

$$v_{i,t-a_i+\tau} \geq y_{i,t} - y_{i,t-1}, \quad i \in \mathcal{A}; \quad t = a_i + 1, \dots, T; \quad \tau = 0, \dots, a_i - 1. \quad (8)$$

$$y_{i,t}, v_{i,t} \in \{0, 1\}, \quad i = 1, \dots, N; \quad t = 1, \dots, T, \quad (9)$$

where a_i is the (integer) setup time for item i , expressed in periods. The set \mathcal{A} is defined as the set of item numbers with positive setup time, thus $\mathcal{A} = \{i | a_i > 0\}$. Furthermore, constants $D_{i,t}$ are defined as the *normalized* cumulative demand for item i up to period t , i.e. demand expressed as the (integer) number of required production periods:

$$D_{i,t} = \max \left(0, \left\lceil \left(\sum_{\tau=1}^t d_{i,\tau} - I_{i,0} \right) / r_i \right\rceil \right) \quad (10)$$

with $\lceil x \rceil$ the smallest integer greater than or equal to x .

Decision variables in the model are $v_{i,t}(y_{i,t})$, which are equal to one when the machine is in setup (production) for item i in period t , and zero otherwise. The set of equations (6a) and (6b) assure that cumulative production up to period t is large enough to fulfill cumulative demand, expressed in terms of required production periods. The equations (7) guarantee that in each planning period the machine is either in production, or in setup, or idle. The coupling between setup ($v_{i,t}$) and production variables ($y_{i,t}$) is expressed by equations (8). Here, it is assumed that these variables are prespecified for $t \leq 0$. Finally, the set of equations (9) represent integrality conditions on setup and production variables.

For DLSP wherein all setup times are zero, the feasibility problem can be solved efficiently by checking whether available capacity up to period t is sufficient to accommodate total demand up to period t , i.e.

$$\sum_{i=1}^N D_{i,t} \leq t \quad \text{for} \quad t = 1, \dots, T. \quad (11)$$

When setup times are nonzero, feasibility cannot be verified using (11), since the number of setups for each item (and therefore the total setup time) is unknown beforehand. In this section it will be shown that when setup times are nonzero, the feasibility problem is NP-Complete.

In the proofs below, the following definitions and notations will be used:

DEFINITION. A *deadline* is a period in which a positive demand occurs. The number of deadlines for item i is denoted by DL_i and the period wherein the n th deadline occurs is denoted by $t_{i,n}^d$, for $n = 0, \dots, DL_i$ ($t_{i,0}^d = 0$ by convention). Moreover, the set of deadline periods is denoted by \mathcal{D} . \square

Furthermore, to introduce the difference between *nonpreemptive* and *preemptive* DLSP schedules, let $y = \{y_{i,t}\}_{\text{all } i,t}$ be a feasible schedule for DLSP and let $L_i(y)$ be the set of periods in which a production batch ends and cumulative production is not equal to any occurring cumulative demand, that is:

$$L_i(y) = \left\{ t \mid y_{i,t} - y_{i,t+1} = 1 \wedge \left(\sum_{\tau=1}^t y_{i,\tau} \neq D_{i,t_{i,n}^d} \text{ for } n = 1, \dots, DL_i \right) \right\}.$$

DEFINITION. A DLSP schedule y is called *nonpreemptive* if $L_i(y) = \emptyset$ for $i = 1, \dots, N$. Otherwise, the schedule is called *preemptive*. \square

LEMMA 1. Every preemptive feasible schedule $y = \{y_{i,t}\}_{\text{all } i,t}$ for problem 1/N/SC/G/SI can be transformed into a feasible nonpreemptive schedule $y^* = \{y_{i,t}^*\}_{\text{all } i,t}$ in polynomial time. Furthermore, the number of setups made for schedule y^* is not greater than for schedule y .

PROOF. For an arbitrary item i and an arbitrary deadline $n (= 0, \dots, DL_i - 1)$, let t_1 be the first period t for which $\sum_{\tau=1}^t y_{i,\tau} = D_{i,t_{i,n+1}^d}$. Determine t_0 as the first period t for which $\sum_{\tau=1}^t y_{i,\tau} > D_{i,t_{i,n}^d}$. Note that in the interval $[t_0, t_1]$ production takes place during $q = D_{i,t_{i,n+1}^d} - D_{i,t_{i,n}^d}$ periods. All $t \in L_i(y)$ which may exist in the interval $[t_0, t_1 - 1]$ can be eliminated without creating additional setups by replanning the production of q to occur in the interval $[t_1 - q + 1, t_1]$ and feasibility is maintained by, if necessary, shifting

production of items j ($\neq i$) to earlier periods. The latter is also done without creating additional setups. Call the resulting schedule y' . Now, it is clear that $L_i(y') = L_i(y) \setminus \{t \mid t \in [t_0, t_1]\}$. Furthermore $L_j(y') \subseteq L_j(y)$ for all items $j = 1, \dots, N$. Proceeding in this way, ultimately a schedule y^* results for which $L_i(y^*) = \emptyset$ for $i = 1, \dots, N$. Since the number of deadlines is bounded by T , and since finding a pair (t_0, t_1) can be done in $\mathcal{O}(T)$, the algorithm runs in $\mathcal{O}(T^2)$. \square

In Theorem 1 we prove that problem $(1/*A/G/SI)_F$ is NP-Complete. The proof of this theorem uses a reduction from the Job Class Scheduling Problem (*JCSP*) considered by Bruno and Downey (1978). This problem can be formulated as follows:

JCSP: Suppose J jobs are given which have to be carried out on a single machine. The attributes of job j are: an integer processing time β_j , a deadline δ_j , a job class γ_j . The number of different job classes is equal to Γ , and within each job class *three* distinct deadlines exist. If a job does *not immediately* follow another job of the same job class, then a machine setup is required before processing of the job can start. Such a setup takes 1 time unit.

The question to be answered in *JCSP* is whether there exists a *nonpreemptive* schedule in which no job finishes after its deadline. Bruno and Downey (1978) prove that *JCSP* is NP-Complete.

THEOREM 1. *Problem $(1/*A/C/SI)_F$ is NP-Complete, even if all the setup times are equal to 1.*

PROOF. We show that $JCSP \propto (1/*A/C/SI)_F$. Let I_1 be an instance of *JCSP* with attributes as described above. Then an instance I_2 of $(1/*A/G/SI)_F$ can be constructed as follows:

—Each item in I_2 corresponds to a job class in I_1 . Hence the number of items is equal to Γ .

—The set of deadlines for an item in I_2 is equal to the set of deadlines of the jobs in the corresponding job class in I_1 .

— $D_{i,n}^d = \sum_{j \in J_{i,n}} \beta_j$ for $i = 1, \dots, \Gamma$ and $n = 1, \dots, 3$. Here the set of jobs $J_{i,n}$ is defined as: $J_{i,n} = \{j \mid \gamma_j = i \text{ and } \delta_j \leq t_{i,n}^d\}$.

— $a_i = 1$ for $i = 1, \dots, \Gamma$.

It remains to be shown that I_1 is a yes-instance if and only if I_2 is a yes-instance. It is clear that any feasible schedule for I_1 can be interpreted as a feasible schedule for I_2 . This proves the 'only-if' part of the statement.

Unfortunately, a feasible schedule for I_2 may be preemptive and consequently this schedule is not feasible for I_1 , since *JCSP* only allows nonpreemptive schedules. However, Lemma 1 shows that any feasible schedule for I_2 can be transformed into a feasible nonpreemptive schedule for I_2 in polynomial time. This feasible nonpreemptive schedule for I_2 can be interpreted as a feasible schedule for I_1 . This proves the 'if' part of the statement. Since *JCSP* is NP-Complete and since it is obvious that $(1/*A/G/SI)_F$ is in \mathcal{NP} , it follows that $(1/*A/G/SI)_F$ is NP-Complete. \square

4.1.2. Parallel Machine Problems. The feasibility problem for parallel machine systems with zero setup times can be stated mathematically as: Does there exist a feasible solution to the following set of equations:

$$\sum_{\tau=1}^t \sum_{m=1}^M r_{i,m} y_{i,\tau,m} \geq \sum_{\tau=1}^t d_{i,\tau}, \quad i = 1, \dots, N; \quad t = 1, \dots, T. \quad (12)$$

$$\sum_{i=1}^N y_{i,t,m} \leq 1, \quad t = 1, \dots, T; \quad m = 1, \dots, M. \quad (13)$$

$$y_{i,t,m} \in \{0, 1\}, \quad i = 1, \dots, N; \quad t = 1, \dots, T; \quad m = 1, \dots, M, \quad (14)$$

where N is the number of items, M is the number of machines, and T is the number of periods. Furthermore, $r_{i,m}$ is the production rate for item i at machine m , and $d_{i,t}$ is the demand for item i in period t . Decision variables $y_{i,t,m}$ are equal to one if item i is produced in period t at machine m , and zero otherwise. Constraints (12) state that cumulative production for item i is at least equal to cumulative demand in each period t , while (13) assures that for each period-machine combination at most one item is produced. Finally, (14) state that production variables are binary.

In case of identical machines, feasibility can be checked efficiently by an adapted version of the set of equations (11), i.e.,

$$\sum_{i=1}^N D_{i,t} \leq Mt \quad \text{for } t = 1, \dots, T. \quad (11')$$

However, if the machines are uniform, constants $D_{i,t}$ cannot be computed anymore, since the production rate is machine dependent. We will show that in this case the feasibility problem is NP-Complete, and consequently, it is unlikely that an efficient check like (11') exists. The proof is based on the Set Partitioning Problem (SPP), which can be stated as follows:

SPP: Given J positive integers $\tau_1, \tau_2, \dots, \tau_J$ and an integer K , such that $\sum_i \tau_i = 2K$. Does there exist a set $I \subset \{1, \dots, J\}$ for which:

$$\sum_{i \in I} \tau_i = \sum_{i \in I} \tau_i?$$

It is well known that *SPP* is NP-Complete (see Garey and Johnson 1979).

THEOREM 2. *Problem $(PU/*/2/A/C/A)_F$ is NP-Complete.*

PROOF. We show that *SPP* $\propto (PU/*/2/A/C/A)_F$. Let I_1 be an instance of *SPP* with attributes as described above. Then an instance I_2 of $(PU/*/2/A/C/A)_F$ can be constructed as follows:

The number of periods (T) is equal to 1.

The number of machines (M) is equal to J .

$C_m = \tau_m$ for $m = 1, \dots, M$.

$r_i = 1$ for $i = 1, 2$.

$D_{1,1} = D_{2,1} = K$.

Suppose a feasible solution exists for I_2 . Then, by taking the set I as:

$$I = \{ \text{all machine numbers which produce for item 1} \},$$

it is clear that I is a solution to I_1 . On the other hand, it is also clear that any solution to I_1 can be transformed into a feasible solution to I_2 . From the above it follows, that any solution technique for $(PU/*/2/A/C/A)_F$ can be used as a solution technique for *SPP*.

Since *SPP* is NP-Complete, and since $(PU/*/2/A/C/A)_F$ clearly belongs to \mathcal{NP} , it follows that $(PU/*/2/A/C/A)_F$ is NP-Complete. \square

Two remarks can be made with respect to this complexity result. First, it is obvious that the feasibility problem for unrelated machines is also NP-Complete, since uniform machines are a special case of unrelated machines. Secondly, it is clear that a feasible schedule for the single-item problem can be obtained in polynomial time, by scheduling demand period by period at the fastest available machine not yet used.

4.2. Optimization Problems

The optimization problem is the problem of finding a *feasible production* schedule with *minimal total costs*. We restrict our analysis to those problems for which the feasibility problem is solvable in polynomial time, since it is obvious that NP-Completeness of the

feasibility problem implies NP-Hardness of the corresponding optimization problem. Consequently, we consider only the single-machine problem with nonzero setup costs (§4.2.1), and the parallel identical machine problem with zero setup costs (§4.2.2).

4.2.1. *The Single-Machine Problem with Nonzero Setup Costs.* Mathematically, the single-machine optimization problem with nonzero setup costs is the problem of determining a production schedule which satisfies the conditions (1) to (5). The complexity of this problem, which is denoted by $1/*/SI/C/A$ when production costs are constant over time, is stated by the following theorem.

THEOREM 3. *Problem $1/*/SI/C/A$ is NP-Hard.*

PROOF. We show that $SPP \propto 1/*/SI/C/A$. Let I_1 be an instance of SPP . Consider the following instance I_2 of $1/*/SI/C/A$:

The number of periods (T) is equal to $2K + 1$.

The number of items (N) is equal to $J + 1$.

$S_i = 1$ for $i = 1, \dots, N$.

$h_i = 0$ for $i = 1, \dots, N - 1$.

$h_N = 1$.

$r_i = 1$ for $i = 1, \dots, N$

$$d_{i,t} = \begin{cases} \tau_i & \text{for } i = 1, \dots, N - 1 \quad \text{and} \quad t = 2K + 1, \\ 1 & \text{for } i = N \quad \text{and} \quad t = K + 1, \\ 0 & \text{otherwise.} \end{cases}$$

Now we prove that I_1 is a yes-instance if and only if the optimal solution to I_2 has total costs equal to N . Therefore, suppose a solution with total costs equal to N exists for I_2 . Since there is a deadline for each item, at least N setups must be made in any feasible solution. A solution with total costs equal to N must therefore have no holding costs. Hence, demand for item N is produced in period $K + 1$ and there is exactly one setup for all other items. Some of these items are produced in periods 1 to K and others in periods $K + 2$ to $2K + 1$.

Now it is clear that if the set I is defined as:

$$I = \{ \text{all item numbers produced in the periods 1 to } K \},$$

then I is a solution to I_1 . On the other hand, any solution to I_1 can be transformed into a solution to I_2 , with total cost equal to N . From the above it follows that any solution technique for $1/*/SI/C/A$ can be used as a solution technique for SPP . Since SPP is NP-Complete, it follows that the optimization problem $1/*/SI/C/A$ is NP-Hard. \square

4.2.2. *The Parallel Identical Machine Problem with Zero Setup Costs.* In Theorem 4 below we state the computational complexity of the parallel identical machine problem with zero setup costs, which is denoted by $PI/M/N/A/G/A$.

THEOREM 4. *Problem $PI/M/N/A/G/A$ is solvable in polynomial time.*

PROOF. The reader may verify that $PI/M/N/A/G/A$ can be formulated as a transportation problem. Since it is well known that the transportation problem is solvable in polynomial time, $PI/M/N/A/G/A$ is solvable in polynomial time too. \square

With respect to this complexity result two remarks can be made. First, it can easily be seen that the special case of problem $P/M/N/A/G/A$ for which $r_{i,m} \in \{0, r_i\}$ for all $i = 1, \dots, N$ and $m = 1, \dots, M$ can also be formulated as a transportation problem. Consequently, this problem is also solved in polynomial time. Secondly, it can be shown (see Salomon et al. 1989) that $PI/M/N/A/C/A$ is solved in a *greedy* fashion, using the following *column minima* procedure:

COLUMN MINIMA PROCEDURE.

Step 0. Sort and renumber all items such that $r_1h_1 \geq r_2h_2 \geq \dots \geq r_Nh_N$.

Step 1. Schedule for the lowest numbered item i , not yet considered, all demand as late as possible.

Step 2. Update the available capacity and repeat Step 1 until all demand is scheduled for all items.

The reader may verify that the running time of this procedure is $\mathcal{O}(N \log N + NT)$.

5. Summary and Discussion

Below we briefly review the literature on algorithms for DLSP, and summarize the complexity results obtained in this paper.

5.1. The Single-Item Problem

Problems $1/1/SI/G/A$ and $1/1/SI/G/SI$ were investigated by Salomon et al. (1989) and Gavish and Johnson (1990). The authors formulate straightforward Dynamic Programming (DP) algorithms, with running times of $\mathcal{O}(T|\mathcal{D}|)$, and $\mathcal{O}(a_1T|\mathcal{D}|)$ respectively. Additional results with respect to the single-item problem were obtained by Kuik et al. (1989). They present an $\mathcal{O}(T \log T)$ algorithm for solving $1/1/SI/C/A$, based on DP. Moreover, they propose a linear programming formulation for this problem, which naturally leads to all-integer solutions, when solved by the simplex method. Lasdon and Terjung (1971) and Schrage (1982) considered the parallel identical machine problem $PI/M/1/SI/C/A$, and propose a DP algorithm with a running time of $\mathcal{O}(T^2M^3)$.

Summarizing the results derived in the above-mentioned papers, it is concluded that DLSP is one of the few capacitated lotsizing problem formulations for which the single-item, single-machine version can be solved efficiently, even in case of nonzero setup times.

5.2. Multi-Item Problems

In Table 3 below we summarize the complexity results obtained in this paper. Algorithms for solving (NP-Hard) multi-item optimization problems can be subdivided into (i) *DP algorithms*, (ii) *mathematical programming algorithms*, and (iii) *decomposition algorithms*.

An illustration of the DP-approach for solving DLSP is found in Salomon et al. (1989). However, although these DP algorithms run in polynomial time when both the number of items (N) and the number of machines (M) are part of the problem description, computation times and memory requirements will become prohibitively large for all but

TABLE 3
Complexity Results for DLSP

Problem	Feasibility Problem	Optimization Problem
$PU/*/2/A/G/A$ $P/*/2/A/G/A$	NP-Complete ¹	NP-Hard
$PI/M/N/A/G/A$	Polynomially solvable	Polynomially solvable
$1/*/SI/G/A$	Polynomially solvable	NP-Hard ²
$1/*/A/G/SI$	NP-Complete	NP-Hard

¹ If $r_{i,m} \in \{0, r_i\}$ the problem is polynomially solvable.

² Even problem $1/*/SI/C/A$ is NP-Hard.

very small problem instances. As a result, these DP algorithms are mainly of theoretical interest. For this reason we must resort to alternative algorithms for solving larger problem instances. For instance, Magnanti and Vachani (1987) suggest a mathematical programming based algorithm for a problem closely related to $1/N/SI/G/A$. They derive a set of valid inequalities which can be added to the LP-relaxation of DLSP to tighten lower bounds. Optimal solutions are obtained using heuristics combined with a branch-and-bound procedure. Fleischmann (1988) proposes a decomposition algorithm for problem $1/N/SI/G/A$, which relies upon Lagrangean relaxation, subgradient optimization, DP, and an enumeration procedure. Computational experience shows that this algorithm is quite effective, both in terms of computational speed as well as in terms of required memory. A modification of this algorithm, to solve problem $1/N/SD/G/A$, is presented in Fleischmann and Popp (1988). Cattrysse et al. (1990) suggest a heuristic for solving $1/N/SI/G/SI$. This heuristic, which is based on column generation techniques and DP, is one of the few procedures that deals with a capacitated lotsizing problem that includes setup times. Finally, Lasdon and Terjung (1971) propose a heuristic based on column generation and dynamic programming for the parallel identical machine problem $PI/M/N/SI/C/A$.

5.3. Conclusions

DLSP is an interesting model for two reasons mainly. First, DLSP is of interest because it combines lotsizing (medium-range planning) and job scheduling (short-range planning). This may be important in many practical situations, where inconsistency between medium- and short-range planning may cause serious problems. Second, DLSP is interesting because empirical studies by Lasdon and Terjung (1971), Fleischmann (1988), and Cattrysse et al. (1990) show that DLSP has computational advantages over other capacitated lotsizing formulations, such as CLSP.

References

- BITRAN, G. R. AND H. H. YANASSE, "Computational Complexity of the Capacitated Lotsizing Problem," *Management Sci.*, 28, 10 (1982), 1174-1186.
- AND H. MATSUO, "Approximation Formulations for the Single-Product Capacitated Lot Size Problem," *Oper. Res.*, 34 (1986), 63-74.
- BRUNO, J. AND P. DOWNEY, "Complexity of Task Sequencing with Deadlines, Set-up Times and Changeover Costs," *SIAM J. Comp.*, 7, 4 (1978), 393-404.
- CATTRYSSE, D., M. SALOMON, R. KUIK AND L. N. VAN WASSENHOVE, "Heuristics for the Discrete Lotsizing and Scheduling Problem with Setup Times," Working Paper 62, Rotterdam School of Management, Erasmus University Rotterdam, The Netherlands (NL), 1990.
- ELMAGHRABY, S., "The Economic Lot Scheduling Problem (ELSP): Review and Extensions," *Management Sci.*, 24, 6 (1978), 587-598.
- FLEISCHMANN, B., "The Discrete Lotsizing and Scheduling Problem," *European J. Oper. Res.*, 44, 3 (1990), 337-348. (Appeared previously as Working Paper, University of Hamburg, Germany (FRG), 1988.)
- AND T. POPP, "Das dynamische Losgrößenproblem mit reihenfolgeabhängigen Rüstkosten," Working Paper, University of Hamburg, Germany (FRG), 1988 (in German).
- GAREY, M. R. AND D. S. JOHNSON, *Computers and Intractability, A Guide to the Theory of NP-Completeness*, W.H. Freeman and Company, New York, 1979.
- GAVISH, B. AND R. E. JOHNSON, "A Fully Polynomial Absolute Approximation Scheme for Single Product Scheduling in a Finite Capacity Facility," *Oper. Res.*, 38 (1990), 70-83.
- KARMAKAR, U. S. AND L. SCHRAGE, "The Deterministic Dynamic Product Cycling Problem," *Oper. Res.*, 33 (1985), 326-345.
- , S. KEKRE AND S. KEKRE, "The Deterministic Lotsizing Problem with Startup and Reservation Costs," *Oper. Res.*, 35 (1987), 389-398.
- KUIK, R., M. SALOMON, S. VAN HOESSEL AND L. N. VAN WASSENHOVE, "The Single Item Discrete Lotsizing and Scheduling Problem: Optimization by Linear and Dynamic Programming," Paper presented at the Special Interest Conf. Combinatorial Optimization (CO89), Leeds (UK), 1989.
- LASDON, L. S. AND R. C. TERJUNG, "An Efficient Algorithm for Multi-Item Scheduling," *Oper. Res.*, 19 (1971), 946-969.
- LAWLER, E. L., J. K. LENSTRA, A. H. G. RINNOOY KAN AND D. B. SHMOYS, "Sequencing and Scheduling:

- Algorithms and Complexity," Report BS-R89, Centre for Mathematics and Computer Science, Amsterdam, The Netherlands, 1989.
- MAES, J. AND L. N. VAN WASSENHOVE, "Multi-Item Single-Level Capacitated Dynamic Lot-Sizing Heuristics: A General Review," *J. Oper. Res. Soc.*, 39, 11 (1988), 991-1004.
- MAGNANTI, T. L. AND R. VACHANI, "A Strong Cutting-Plane Algorithm for Production Scheduling with Changeover Costs," Working Paper OR 173-87, Massachusetts Institute of Technology, 1987.
- DE MATTA, R. AND M. GUIGNARD, "Production Scheduling with Sequence-Independent Changeover Cost," Working Paper, The Wharton School, University of Pennsylvania, 1989.
- MONMA, C. L. AND C. N. POTTS, "On the Complexity of Scheduling with Batch Set-up Times," *Oper. Res.*, 37 (1989), 798-804.
- SALOMON, M., L. G. KROON, R. KUIK AND L. N. VAN WASSENHOVE, "The Discrete Lotsizing and Scheduling Problem," Working Paper 30, Rotterdam School of Management, Erasmus University Rotterdam, The Netherlands (NL), 1989.
- SCHRAGE, L., "The Multiproduct Lot Scheduling Problem," In M. A. H. Dempster, J. K. Lenstra and A. H. G. Rinnooy Kan (Eds.), *Deterministic and Stochastic Scheduling*, D. Reidel Publishing Company, Dordrecht, Holland, 1982, 233-244.
- VAN WASSENHOVE, L. N. AND P. VANDERHENST, "Planning Production in a Bottleneck Department," *European J. Oper. Res.*, 12 (1983), 127-137.

Copyright 1991, by INFORMS, all rights reserved. Copyright of Management Science is the property of INFORMS: Institute for Operations Research and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.