Available online at www.sciencedirect.com
SCIENCE @ DIRECT®

PERGAMON

Omega 31 (2003) 365–378

omega
The International Journal of Management Science

www.elsevier.com/locate/dsw

# The capacitated lot sizing problem: a review of models and algorithms

B. Karimi[a], S.M.T. Fatemi Ghomi[a], J.M. Wilson[b,*]

[a] *Industrial Engineering Department, Amirkabir University of Technology, Tehran, Iran*
[b] *Business School, Loughborough University, Loughborough LE11 3NL, UK*

Received 21 July 2000; accepted 15 April 2003

**Abstract**

Lot sizing is one of the most important and also one of the most difficult problems in production planning. This subject has been studied extensively in the literature. In this article, we consider single-level lot sizing problems, their variants and solution approaches. After introducing factors affecting formulation and the complexity of production planning problems, and introducing different variants of lot sizing and scheduling problems, we discuss single-level lot sizing problems, together with exact and heuristic approaches for their solution. We conclude with some suggestions for future research.


*Keywords:* Capacitated lot sizing problem; Heuristics; Integer programming; Production planning

## 1. Introduction

Production planning is an activity that considers the best use of production resources in order to satisfy production goals (satisfying production requirements and anticipating sales opportunities) over a certain period named the *planning horizon*.

Production planning typically encompasses three time ranges for decision making: *long-term*, *medium-term* and *short-term*. In long-term planning usually the focus is on anticipating aggregate needs and involves such strategic decisions as product, equipment and process choices, facility location and design, and resource planning. Medium-term planning often involves making decisions on material requirements planning (MRP) and establishing production quantities or lot sizing over the planning period, so as to optimise some performance criteria such as minimising overall costs, while meeting demand requirements and satisfying existing capacity restrictions. In short-term planning, decisions usually involve day-to-day scheduling of operations such as job sequencing or control in a workshop.

In this review the focus is on medium-term production planning and especially on *single-level lot sizing* decisions. Lot sizing decisions give rise to the problem of identifying when and how much of a product to produce such that setup, production and holding costs are minimised. Making the right decisions in lot sizing will affect directly the system performance and its productivity, which are important for a manufacturing firm's ability to compete in the market. Therefore, developing and improving solution procedures for lot sizing problems is very important. The applicability of these problems arises commonly in operations such as forging and casting and in industries which consist of a single production process, or where all production process can be considered as a single operation, such as some medical or chemical industries.

After an introduction to lot sizing problems, this paper will focus on the capacitated lot sizing problem and will review the main contributions to this long standing but active research field focusing, particularly, on developments that have taken place since research was reviewed and compared in [1,2].

* Corresponding author. Tel.: +44-1509-223-108; fax: +44-1509-223-960.
*E-mail address:* j.m.wilson@lboro.ac.uk (J.M. Wilson).

0305-0483/03/$ - see front matter 
doi:10.1016/S0305-0483(03)00059-8

## 2. Characteristics of lot sizing models

The complexity of lot sizing problems depends on the features taken into account by the model. The following characteristics affect classifying, modelling and the complexity of lot sizing decisions.

### 2.1. Planning horizon

The *planning horizon* is the time interval on which the master production schedule extends into the future. The planning horizon may be *finite* or *infinite*. A finite-planning horizon is usually accompanied by dynamic demand and an infinite planning horizon by stationary demand. In addition, the system can be observed continuously or at discrete time points, which then classifies it as a continuous- or discrete-type system. In terms of time period terminology, lot sizing problems fall into the categories of either *big bucket* or *small bucket* problems. Big bucket problems, are those where the time period is long enough to produce multiple items (in multi-item problem cases), while for small bucket problems the time period is so short that only one item can be produced in each time period. Another variant of the planning horizon is a *rolling horizon* usually considered when there is uncertainty in data. Under this assumption, optimal approaches for each horizon act as heuristics but cannot guarantee the optimal solution.

### 2.2. Number of levels

Production systems may be *single-level* or *multi-level*. In single-level systems, usually the final product is simple. Raw materials, after processing by a single operation such as forging or casting, are changed to final product. In other words, the end item is directly produced from raw materials or purchased materials with no intermediate subassemblies. Product demands are assessed directly from customer orders or market forecasts. This kind of demand, as will be further discussed later, is known as *independent demand*. In multi-level systems, there is a parent–component relationship among the items. Raw materials after processing by several operations change to end products. The output of an operation (level) is input for another operation. Therefore, the demand at one level depends on the demand for its parents' level. This kind of demand is named *dependent demand*. Multi-level problems are more difficult to solve than single-level problems.

Multi-level systems are further distinguished by the type of product structure, which includes *serial*, *assembly*, *disassembly and general* or *MRP* systems.

### 2.3. Number of products

The number of end items or final products in a production system is another important characteristic that affects the modelling and complexity of production planning problems. There are two principal types of production system in terms of number of products. In *single-item* production planning there is only one end item (final product) for which the planning activity has to be organised, while in *multi-item* production planning, there are several end items. The complexity of multi-item problems is much higher than that of single-item problems. van Hoesel and Wagelmans [3] provide theoretical results for the performance of algorithms for the single item capacitated lot sizing problem. (See also Section 4 of this paper.)

### 2.4. Capacity or resource constraints

Resources or capacities in a production system include manpower, equipment, machines, budget, etc. When there is no restriction on resources, the problem is said to be *uncapacitated*, and when capacity constraints are explicitly stated, the problem is named *capacitated*. Capacity restriction is important, and directly affects problem complexity. Problem solving will be more difficult when capacity constraints exist.

### 2.5. Deterioration of items

In the case that deterioration of items is possible, we encounter restrictions in the inventory holding time. This in turn is another characteristic which would affect problem complexity.

### 2.6. Demand

Demand type is considered as an input to the model of the problem. *Static demand* means that its value does not change over time, it is stationary or even constant, while *dynamic demand* means that its value changes over time. If the value of demand is known in advance (static or dynamic), it is termed *deterministic*, but if it is not known exactly and the demand values occurring are based on some probabilities, then it is termed *probabilistic*. In *independent* demand cases, an item's requirements do not depend on decisions regarding another item's lot size. This kind of demand can be seen in single-level production systems. In multi-level lot sizing, where there is a parent–component relationship among the items, because the demand at one level depends on the demand for their parents (pervious level), it is called *dependent*. Problems with dynamic and dependent demands are much more complex than problems with static and/or independent demands. Also, problems with probabilistic demand will be more complex than problems with deterministic demand.

### 2.7. Setup structure

Setup structure is another important characteristic that directly affects problem complexity. Setup costs and/or setup times, are usually modelled by introducing zero–one variables in the mathematical model of the problem and cause problem solving to be more difficult. Usually, production changeover between different products can incur setup time and setup cost. There are two types of setup structure: *simple setup* structure and *complex setup* structure. If the setup time and cost in a period are independent of the sequence and the decisions in previous periods, it is termed a simple setup structure, but when it is dependent on the sequence or previous periods, it is termed a complex setup. Three types of complex setups will now be described. First, if it is possible to continue the production run from the previous period into the current period without the need for an additional setup, thus reducing the setup cost and time, the structure is named *setup carry-over*. We also can define a second type of complex setup, family or major setup, caused by similarities in manufacturing process and design of a group of item(s). An item setup or minor setup also occurs when changing production among items within the same family. If we have sequence-dependent setup, item setup cost and time depend on the production sequence; this is the third type of complex setup structure. It is obvious that the complex structures are more awkward in both modelling and solving the lot sizing problems.

### 2.8. Inventory shortage

Inventory shortage is another characteristic affecting modelling and complexity of problem solving. If shortage is allowed it means that it is possible to satisfy the demand of the current period in future periods (*backlogging* case), or it may be allowable for demand not to be satisfied at all (*lost sales* case). The combination of backlogging and lost sales is also possible (see for instance [4] for a model of this type based on a Weibull distribution). This usually introduces a shortage cost in the objective function. Problems with shortage are more difficult to solve than without shortage.

### 2.9. Previous reviews

Gelders and Van Wassenhove [5] in their review paper, discussed medium- and short-term production planning. In particular, they discussed hierarchical planning, material requirement(s) planning (MRP), lot sizing and scheduling and they classified the concepts and variants of these problems. Bahl et al. [1] in their outstanding review paper, classified lot sizing problems into four categories based on type of demand and presence or absence of resource constraints: single-level lot sizing without resource constraints (SLUR), single-level lot sizing with resource constraints (SLCR), multi-level lot sizing without resource constraints (MLUR), and multi-level lot sizing with resource constraints (MLCR). The characteristics and solution approaches for each category are discussed and some suggestions for further research in each category are presented.

## 3. Variants of lot sizing and scheduling problems

Because the range of lot sizing problems is very large, in this review we will only focus on deterministic, single-level dynamic lot sizing—the capacitated lot sizing problem (CLSP). CLSP, which is an NP-hard problem, will be discussed in detail in the next section. In the remainder of this section, five other problem variants will be identified together with an associated reference. These are: the economic lot scheduling problem (ELSP), the discrete lot sizing and scheduling problem (DLSP), the continuous setup lot sizing problem (CSLP), the proportional lot sizing and scheduling problem (PLSP), and the general lot sizing and scheduling problem (GLSP).

ELSP [6] is a single-level, multi-item problem with stationary demand. The time is continuous and planning horizon is infinite. Solving the ELSP where capacity restrictions are involved is NP-hard.

The NP-hard problem DLSP [7,8] subdivides the (macro) periods of the CLSP into several (micro) periods. The fundamental assumption of the DLSP is the so-called *all-or-nothing* production, which means only one item may be produced per period, and, if so, the production amount would be as much as using full capacity. From this viewpoint, DLSP is called a small bucket problem.

CSLP [9] is a step towards a more realistic situation compared to DLSP. In CSLP the *all-or-nothing* assumption, that seems to be strict and makes efficient implementation of mathematical programming approaches possible, does not exist any more, but still only one item may be produced per period.

The basic idea behind the PLSP [10] is to use the remaining capacity for scheduling a second item in the particular period, if the capacity of a period is not used in full. This is in fact the shortcoming of the CSLP. The underlying assumption of the PLSP is that the setup state of the machine can be changed at most once per period. Production in a period could take place only if the machine is properly set up either at the beginning or at the end of the period. Hence, at most two products may be produced per period.

GLSP [11,12] integrates lot sizing and scheduling of several products on a single capacitated machine. Continuous lot sizes are determined and scheduled, thus generalising models using restricted time structures.

It should also be noted that DLSP introduces a connection between batching and lot sizing. Jordan and Drexl [7] develop an algorithm for the batch sequencing problem (BSP) and use this as a building block to solve DLSP. A paper by Fisher et al. [13] considers end effects of inventory policies for DLSP and produces optimal or near-optimal results on problems using a new algorithm. Further connections between scheduling and batching are reviewed in [14].

## 4. The single-level lot sizing problem

In this section, we describe the literature relating to the single-level lot sizing problem. First, we begin with the uncapacitated case and then we consider the capacitated case.

### 4.1. The single-level uncapacitated lot sizing problem

Many articles have discussed the single-level uncapacitated lot sizing problem. In this section, we present the model and review some major contributions related to this problem.

**Assumption.**

- Planning horizon is finite and consists of $T$ periods.
- Demand, $(d_t,\ t = 1,\dots,T)$ is known in each period and is satisfied at the beginning of the period.
- Variable production unit cost is independent of production amount.
- Each unit item is produced independently from other units.
- Lead time is known and constant (without loss of generality it is set to zero).
- No shortage(s) are allowed.
- Setup cost for each production lot is constant over time.
- Inventory holding cost is linear and is paid to the end of period stock.
- Without loss of generality the initial and terminal inventories are set to zero.

The objective is to minimize the sum of setup, production and inventory holding cost under the above assumptions. As it is well known [15] that every $N$-item uncapacitated problem can easily be divided into $N$ single-item uncapacitated problems, we only present the single-item formulation.

**Notation**

$S_t$ setup cost in period $t$
$Y_t$ a binary variable that assumes value 1 if the product is produced in period $t$ and 0, otherwise
$C_t$ variable unit production cost in period $t$
$X_t$ production amount in period $t$
$h_t$ unit inventory holding cost in period $t$ (usually constant for all $t$)
$I_t$ inventory at the end of period $t$

$$M_t = \sum_{k=t}^{T} d_k.$$

#### 4.1.1. Model

The single-item uncapacitated lot sizing problem can be formulated as follows:

$$\text{Minimise } Z = \sum_{t=1}^{T} (S_t Y_t + C_t X_t + h_t I_t)$$

$$\begin{aligned} \text{Subject to}\quad & X_t + I_{t-1} - I_t = d_t \quad (t = 1,\dots,T),\\ & X_t \leqslant M_t Y_t \quad (t = 1,\dots,T),\\ & Y_t \in \{0,1\} \quad (t = 1,\dots,T)\\ & X_t, I_t \geqslant 0 \quad (t = 1,\dots,T). \end{aligned}$$

#### 4.1.2. Algorithms

Many authors have studied the problem modelled in 4.1.1. Historically, the *economic order quantity* (EOQ) presented by Harris [16] predates this problem. EOQ is also known as the *Wilson lot size formula* since it was used in practice by Wilson [17]. EOQ balances the setup cost and inventory holding cost. In the EOQ model, demand is known with a stationary rate and the planning horizon is infinite.

Wagner and Whitin [18] presented a dynamic programming algorithm for the single-item uncapacitated lot sizing problem, which provides its *optimal* solution. However, algorithms running in linear time are now available, essentially removing the need for heuristic algorithms.

Wagelmans et al. [19] and Aggarwal and Park [20] have proposed algorithms based on dynamic programming for the uncapacitated lot sizing problem case of Wagner–Whitin. Federgruen and Tzur [21] also have proposed a forward algorithm that solves the general single-item lot sizing model. The main contribution of the three above-mentioned papers is a reduction in computational complexity in comparison with the Wagner–Whitin algorithm. The Wagelmans et al. [19] paper provides an $O(n \log n)$ algorithm that runs in linear time in the Wagner–Whitin case, while the paper by Federgruen and Tzur [21] develops an alternative simple forward algorithm to solve general dynamic lot sizing models with equivalent complexity properties. More recent work by van Hoesel and Wagelmans [3] provides a theoretical underpinning for fully polynomial approximation schemes for single-item capacitated economic lot sizing problems. Wolsey [22] has also described some solution approaches for the uncapacitated lot sizing problem. Stadtler [23] develops a modified model which looks at demand forecasts beyond the planning horizon. He states that ‘As there are algorithms available now that solve the standard SLLSP in linear time former arguments favouring simple myopic heuristics because of its solution quality and computational complexity have become obsolete’. He is of the view that ideas developed by Tempelmeier and Helber [24] and improved on by him, Stadtler [25], in the form of the shortest-route model show future promise.

For certain models, demand may be independent of time period and can be regarded as constant. An early reference to models with constant demand is [26] but more recently Ganas and Papachristos [27] looks at heuristics for constant demand models and Papachristos and Ganas [28] have investigated the effect of stationary demand on optimal inventory policy. The authors develop a six-stage algorithm and construct a solution table which provides simple to understand information regarding the stability regions for inventory policy.

### 4.2. Single-level capacitated lot sizing problems

In the context of single-level production planning, with finite planning horizon and known dynamic demand without incurring backlogs, the classical capacitated lot sizing problem (CLSP), consists of determining the amount and the timing of the production of products in the planning horizon. Capacity restrictions constrain the production quantity in each period. A fixed setup cost and a linear production cost are specified and there is also an inventory holding cost proportional to the inventory amount and time carried.

In the classical CLSP, although the setup costs may vary for each product and each period, they are sequence independent. There are also some variants of CLSP, where setups are sequence dependent. This kind of problem as mentioned before is named the *complex setup* structure. The objective of classical CLSP is to determine a production plan with minimum cost. Its mathematical formulation is as follows:

| | |
|---|---|
| $T$ | number of periods in the planning horizon |
| $X_{it}$ | production (lot size) of item $i$ in period $t$ |
| $I_{it}$ | inventory of item $i$ at the end of period $t$ ($I_{i1} = I_{iT} = 0$, without loss of generality) |
| $Y_{it}$ | a binary variable that assumes value 1 if item $i$ is produced in period $t$ and 0, otherwise |
| $R_t$ | available capacity in period $t$ |
| $d_{it}$ | demand for item $i$ in period $t$ |
| $C_{it}$ | unit production cost of item $i$ produced in period $t$ |
| $S_{it}$ | setup cost incurred if item $i$ is produced in period $t$ |
| $M_{it} = \sum_{k=t}^{T} d_{ik}$ | upper bound on the production of item $i$ in period $t$ |
| $a_i$ | unit resource consumption for item $i$ |
| $h_{it}$ | unit holding cost of item $i$ at the end of period $t$. |

$$\text{Minimise} \quad Z = \sum_{i=1}^{n} \sum_{t=1}^{T} S_{it} Y_{it} + C_{it} X_{it} + h_{it} I_{it}$$

$$\text{Subject to} \quad \sum_{i=1}^{n} a_i X_{it} \leqslant R_t \quad (t = 1, \ldots, T),$$

$$X_{it} + I_{i,t-1} - I_{it} = d_{it}$$

$$(i = 1, \ldots, n; \;\; t = 1, \ldots, T),$$

$$X_{it} \leqslant M_{it} Y_{it} \quad (i = 1, \ldots, n; \;\; t = 1, \ldots, T),$$

$$Y_{it} \in \{0, 1\} \quad (i = 1, \ldots, n; \;\; t = 1, \ldots, T),$$

$$X_{it} \leqslant 0 \quad (i = 1, \ldots, n; \;\; t = 1, \ldots, T),$$

$$I_{it} \geqslant 0 \quad (i = 1, \ldots, n; \;\; t = 1, \ldots, T).$$

The single-item CLSP has been shown by Florian et al. [29] and Bitran and Yanasse [30] to be NP-hard. In consequence, Chen and Thizy [31] have shown that the multi-item CLSP problem, is *strongly* NP-hard. Maes et al. [32] have shown that even finding a feasible solution for CLSP with setup times is NP-hard. Based on these results, it is unlikely that we can develop any effective optimal algorithm for this problem. Therefore, research on developing effective heuristics has been a profitable research area for a long time.

Based on the literature, solution methods of the problem can be classified into three main categories. The first is *exact methods*, the second category is *common-sense or specialised heuristics* and the third category belongs to *mathematical programming-based heuristics*. Table 1 shows the heuristics in each category.

#### 4.2.1. Exact methods

Since the CLSP is NP-hard, most of the practical algorithms are heuristic. However, besides the straightforward implementation of a mixed integer programming formulation of the problem, and using branch and bound technique to solve it, there are basically two other exact approaches. One is Barany et al. [33] and Leung et al. [35] who use cut-generation techniques, and the other is the variable redefinition technique of Eppen and Martin [34].

In the cut-generation technique of Barany et al., by the addition of strong valid inequalities, which are facets for the single-item uncapacitated problem (Wagner–Whitin-type schedules), the problem is reformulated to speed up the solution process and obtain a good approximation of the convex hull of feasible solutions to the CLSP. The inequalities (cuts) are generated using a cutting plane procedure. The resulting reformulated problem is then solved using a branch-and-bound algorithm.

In the second approach, Eppen and Martin, used a variable redefinition technique to change the classic CLSP formulation into a graph-based representation. This reformulation has more variables and constraints, but has a much tighter linear relaxation than the traditional model, and therefore reduced solution time. To find the optimal solution, the LP-relaxation problem is first solved and then a branch and bound procedure is used in the final stage.

These older methods mentioned so far in this section need considerable computational effort to obtain an optimal solution and so were only able to solve relatively small problems within a reasonable time, thus intensifying the reason to explore heuristics.

More recent work suggests that practical sized problems may be solvable. The *bc-prod* system of Belvaux and Wolsey [36] provides a general framework for modelling and solving lot sizing problems. Cuts are generated and various ingenious developments, deduced from the data, are introduced. The problems solved are from various families including big-bucket and small-bucket variants of sizes up

Table 1
Summary of CLSP algorithms

| | | | |
|---|---|---|---|
| Barany et al. [33]<br>Eppen and Martin [34]<br>Leung et al. [35]<br>Belvaux and Wolsey [36]<br>Belvaux and Wolsey [37]<br>Fatemi Ghomi and Hashemin [38] | Exact methods | | Methods for solving CLSP |
| Eisenhut [39]<br>Lambrecht and Vanderveken [40]<br>Dixon and Silver [41]<br>Maes and Van Wassenhove [42]<br>Kirca and Kokten [15] | Period-by-period heuristics | Common-sense or specialised heuristics | |
| Dogramaci et al. [43]<br>Karni and Roll [44]<br>Gunther [45]<br>Selen and Heuts [46]<br>Trigeiro [47] | Improvement heuristics | | |
| Newson [48]<br>Billington et al. [49]<br>Thizy and Van Wassenhove [50]<br>Bitran and Matsuo [51]<br>Trigeiro [52]<br>Trigeiro et al. [53]<br>Chen and Thizy [31]<br>Thizy [54]<br>Diaby et al. [55]<br>Millar and Yang [56] | Relaxation heuristics | Mathematical Programming Based Heuristics | |
| Gelders et al. [57]<br>Diaby et al. [58]<br>Chung et al. [59]<br>Lotfi and Yoon [60]<br>Hindi [61]<br>Armentano et al. [62] | Branch-and-bound heuristics | | |
| Manne [63]<br>Cattrysse et al. [64] | Set partitioning and column generation heuristics | | |
| Lozano et al. [65]<br>Hindi [66]<br>Hindi [67]<br>Hung and Hu [68] | Other heuristics | | |

to 10 machines, up to 20 items and up to three periods. The overall idea is that a variety of types of problem variants can be solved by a general approach, namely integer programming with cuts. The work is further extended in [37] where the modelling of start-ups, changeovers and switch-offs is introduced.

Most of the previous works have considered multi-item CLSP. However, recently Fatemi Ghomi and Hashemin [38] considered single item CLSP. They developed an analytical method based on reformulating the problem as a shortest path problem. They have shown that under certain circumstances their algorithm provides optimal solutions.

#### *4.2.2. Common-sense or specialised heuristics*

The structure of common-sense heuristics is often characterised by three steps: the lot sizing step, the feasibility routine and the improvement step.

The lot sizing step essentially consists in converting a given matrix of demand $d_{it}$ into a matrix of production lot sizes $x_{it}$. One of the differences between heuristics is the way in which they combine demands into lots, i.e. in their lot sizing step. The second part of the structure of the heuristic is a feasibility routine. This part of the heuristic ensures that all demand is satisfied without backlogging. Since it is possible that in some periods total demand exceeds total capacity, in these cases some inventory should be built up in earlier periods with slack capacity. There are two basic feasibility mechanisms that can be employed for the feasibility step: the *feedback mechanism* and the *look-ahead mechanism*. In the feedback mechanism, whenever a period in which demand exceeds available capacity is encountered during the execution of the lot sizing step (an infeasible period), excess demand is pushed back to earlier periods with leftover capacity, taking into consideration setup and holding costs. In a look-ahead mechanism, the minimum required inventory build-up in every period in order to avoid capacity violations in later periods is computed a priori. In other words, the cumulative requirements up to any period $t$ ($t \leqslant T-1$) are computed a priori, so that inventory is held in anticipation of infeasibility in future period ($t+1$). The lot sizing step is then adjusted so that the planned production lots in each period satisfy these conditions.

In the improvement step several rules are applied to the solution obtained by the lot sizing step to refine and improve the current solution and find further savings. Since in the improvement step, demand splitting is allowed over different lots (whereas it is not allowed in the lot sizing step), this step often is very effective and can improve the solution considerably.

As Maes and Van Wassenhove [69] describe, heuristics in the common-sense category can be classified into two groups, period-by-period heuristics, and improvement heuristics.

*Period-by-period heuristics*: Period-by-period heuristics work from period 1 to $T$ in a single-pass construction algorithm. After producing the required amount for all products in period $t$ ($\max\{0, d_{it} - I_{i,t-1}\}$), to save the setup costs, any excess capacity can be used to produce for demand in future periods. To choose the product and the amount of its production for future periods, all heuristics use a priority index. These priority indexes can be any well-known criteria taken from uncapacitated dynamic lot sizing heuristics such as the Silver and Meal [70] heuristic, part-period balancing, etc. Among this group of heuristics, Eisenhut's heuristic [39] is the pioneering work. Other more recent heuristics are Maes and Van Wassenhove [42], Gunther [45], Trigeiro [53], Selen and Heuts [46], and Kirca and Kokten [15].

Maes and Wassenhove [42], also use a period-by-period approach. In the current period the products which must be produced are determined, i.e. products that need setup. By identifying the minimum amount of production for each product in the current period and by considering the available capacity, the algorithm seeks products for which their future demand periods can be satisfied in the current period. For this purpose, the algorithm uses well-known single-item uncapacitated heuristics such as Silver–Meal, least-unit cost, least total cost or absolute cost criterions. The sequence in which the future demands are checked for inclusion in the production lots is important. The items are always searched in lexicographic order. There are two basic strategies for doing this. The first is named *east* strategy, which adds demands to the production lot until either the criterion (e.g. S–M or P-P-B, etc.) is no longer satisfied or there does not exist enough capacity. The second strategy named *south*, searches all demands in a given period in lexicographic order before the procedure moves to the next period. A demand will be added to a production lot when the criterion is satisfied and when the available capacity is sufficient. Besides the two pure strategies, a mixed *southeast* strategy can be used, that is the items are divided into classes according to their time between orders. Since searching for items in each period is in lexicographic order, a priori ordering of the items is very important. Maes and Wassenhove implemented six types of orderings: TBO (i.e. ordering according to non-increasing values of time between orders), SH (according to non-increasing values of ordering cost over inventory holding cost, $S_i/h_i$), SHC (according to non-increasing $S_i/h_i a_i \bar{d}_i$ ratios), EC (according to non-increasing values of expected average cost per period), ES (according to expected savings when combining demands over TBO periods into a single production lot), ESC (same as ES but normalised for average capacity utilisation). Based on the three parts of their algorithm, i.e. various prior ordering of the items, the criterion used to decide whether or not to include demand into the current production lot and the search strategy, their heuristic has 72 variants ($6 \times 4 \times 3$). Their algorithm uses a look-ahead routine (but different from [41]) to ensure feasibility. They also used a lot elimination approach in the improvement step.

Kirca and Kokten [15] have developed a heuristic algorithm, named *item-by-item*. This algorithm differs in that at each iteration, a set of items from those not already scheduled is selected and production schedules over the planning horizon for this set of items are determined. The selection rule for scheduling is an algorithm named 1-*item algorithm*, which is based on the well-known economic order quantity (EOQ) concept where uniform demand is assumed (average demand per period). By using this selection rule the $N$-item problem changes to $N$-single-item problems. In the next step, the available capacity in each period is updated. A capacity adjustment factor is used to cope with high usage of the capacity by the items that are scheduled at the initial iterations. The resulting single-item bounded lot sizing problems are then solved by a dynamic programming based procedure. Their computational results demonstrate that an algorithm which uses this approach is more efficient than some other well-known algorithms such as Lambrecht and Vanderveken [40], Dixon and Silver, Maes and Wassenhove, and Cattrysse et al. [64].

*Improvement Heuristics*: Heuristics in this category all start with an initial solution (often infeasible) for the complete planning horizon usually found by uncapacitated lot sizing techniques. These heuristics generally contain three steps. In the first step generate an initial solution, ignoring capacity constraints. In the second step try to enforce feasibility conditions, by shifting lots from period to period at minimal extra cost. Finally, in the third (cost reduction) step, the aim is to maximise cost savings as long as no new infeasibilities are incurred. For choosing shifts, usually a simple rule based on trade-off between setup cost and holding cost is used, and both left and right shifts can be applied. One such simple rule for example could be

$$\Delta C = S_i Y + h_i(t-k)\Delta x_{it}$$

where $\Delta C$ is the potential cost saving, $S_i$ the setup cost for product $i$ (assumed the same at any period), $Y = \begin{cases} 1 & \text{if a new setup is needed in period } t \\ -1 & \text{if a setup is eliminated from period } t \end{cases}$, $k$ the period to which the production lot is shifted, and $\Delta x_{it}$ the amount of item $i$ shifted from period $t$ to production in period $k$.

The *four-step algorithm* of Dogramaci et al. [43] begins with a lot-for-lot initial solution (step I), which ignores the capacity constraints. The result of this step is a solution with minimum inventory cost and maximum setup cost. In the next step (II) it tries to reduce the total cost and achieve feasibility. This is done by a left-shift procedure, that searches for shifts with the largest reduction in costs over all items and the whole planning horizon. Step three (III), is basically another left shift, which tries to get more cost reduction as long as the feasibility condition is not violated. Since the algorithm performs in the order I–II–I–III, it is called the *four-step algorithm*.

The algorithm of Karni and Roll [44] is an improvement heuristic that begins with the single item uncapacitated Wagner–Whitin dynamic programming solution. Their algorithm consists of five sub-algorithms and is executed in three phases. In phase one, an initial lower bound solution is derived based on the Wagner–Whitin algorithm. In phase two, by combining adjacent lots in varying proportions it tries to remove infeasibility and improve the solution, for this the algorithm uses both left and right shifts. By forcing changes in the structure of the current best solution, it tries to get further improvements in phase three. To limit the number of possible shifts, the algorithm defines some conditions for effective shifts. Based on these conditions 10 types of shifts are introduced. The effect of a shift is expressed as a cost-saving coefficient based on trade off between setup cost and holding cost, which is similar to the above-mentioned simple rule.

The heuristic of Gunther [45] starts with an initial lot-for-lot solution and comprises three elements: a marginal cost coefficient as a lot sizing rule, capacity constraints which ensure feasibility, and a cost coefficient as a priority index for capacity balancing. The algorithm uses Groff's [71] single-level uncapacitated criterion for the lot sizing rule and defines the lot sizing criterion as the marginal cost saving per unit of additional capacity absorbed. The cost coefficient of product $i$ in current period $t$ is

$$U_i = \frac{[2S_i/h_i - d_{ip}T_i(P(i) - t + 1)]}{d_{ip}a_i},$$

where $t$ is current period, $T_i$ the number of period requirements which a batch of product $i$ will satisfy (time supply), $P(i)$ the next period to $t$ with a positive requirement for product $i$ (supply period), and $d_{ip}$ requirement for product $i$ at period $p(i)$. Other parameters are the same as before. The lot size of the item with the maximum positive $U_i$ is increased, as long as a cost saving can be achieved and capacity is available. To ensure feasibility, Gunther's algorithm defines some constraints on capacity, which can be viewed as a look-ahead procedure for a feasibility step. These constraints ensure that cumulative capacity available exceeds the cumulative capacity requirements with respect to any future period. If there is a positive maximum capacity overload then the lot sizes in the current period have to be increased, subject to capacity constraints mentioned before. The cost increase per unit of additional capacity absorbed serves as a priority index in capacity balancing. The product with the maximum cost increase is selected in order to reduce the capacity overload in the remainder of the horizon. This priority index is given by

$$V_i = ((p(i) - t)q_{ip}h_i + S_i[1 - d(x_{it})])/a_i q_{ip},$$

where $q_{ip}$ is the maximum quantity of product $i$ which can be shifted from period $p(i)$ to period $t$ for pre-production, and $d(x_{it})$ the binary decision variable indicating whether product $i$ is set up in period $t$ or not where

$$d(x_{it}) = \begin{cases} 1 & \text{if } x_{it} > 0, \\ 0 & \text{otherwise.} \end{cases}$$

Selen and Heuts [46] have suggested a modification of Gunther's heuristic. The modification is for the case when an entire future period requirement is added to the existing lot sizes of an already-scheduled product. They have introduced the following priority index:

$$U_i = \frac{(P(i) - t)q_{ip}h_i + S_i(d(x_{ip}, q_{ip}) - d(x_{it}))}{a_i q_{ip}},$$

where

$$d(x_{ip}, q_{ip}) = \begin{cases} 1 & \text{if } x_{ip} > q_{ip}, \\ 0 & \text{if } x_{ip} = q_{ip}. \end{cases}$$

They have mentioned that this modification may outperform Gunther's heuristic.

Trigeiro [47] developed a heuristic algorithm for CLSP with setup times. This heuristic (named *Simple Heuristic*) is also based on the Silver–Meal lot sizing heuristic and

Table 2
Characteristics of specialised heuristics

| Lot shifting | Feasibility check | Priority indices | Initial solution | Algorithm |
|---|---|---|---|---|
| — | — | PPB | — | Eisenhut [39] |
| Left shifting | Feedback | Silver–Meal | — | Lambrecht and Vanderveken [40] |
| Left shifting | Look-ahead | Silver–Meal | — | Dixon and Silver [41] |
| Left shifting | Look-ahead | Several indices | — | Maes and Van Wassenhove [42] |
| — | Cumulative feasibility | EOQ | — | Kirca and Kokten [15] |
| Left and right shifting | Cumulative feasibility | A simple rule | Lot-for-lot | Dogramaci et al. [43] |
| Left and right shifting | Cumulative feasibility | A simple rule | Wagner–Whitin | Karni and Roll [44] |
| Left shifting | Look-ahead | Groff | Lot-for-lot | Gunther [45] |
| Left shifting | Look-ahead | Groff | Lot-for-lot | Selen and Heuts [46] |
| Left shifting | Feedback | Silver–Meal | Lot-for-lot | Trigeiro [47] |

uses the same cost reduction coefficient. The heuristic starts with an initial lot-for-lot solution. Trigeiro used a feedback mechanism for ensuring feasibility. In periods with excess capacity usage of available capacity, among products with positive production, some production is shifted into earlier periods. The criterion is minimising the sum of production, setup and inventory costs, for each unit of overtime eliminated in a shift. Then the algorithm tries to improve the solution with some rearrangement moves.

The improvement heuristics involving checking a large number of shifts and feasibility, and need more computation time compared to period-by-period heuristics, which is a drawback for these kind of heuristics. For some comparisons among the heuristics of Lambrecht and Vanderveken, Dixon and Silver, and Dogramaci et al. refer to [72]. Table 2 summarises the characteristics of the specialised heuristics.

#### 4.2.3. Mathematical programming-based heuristics

Heuristics, which belong to the mathematical programming class, are based on an optimum seeking mathematical programming methodology. Comparing to common sense approaches, to which modifications are very difficult to make (because in most cases we have to alter the heuristic completely), the heuristics of this class usually use a mathematical programming procedure to generate a solution. Mathematical programming heuristics usually produce better quality solutions, are more general and allow extensions to different problems. They also have the advantage that there exist many commercial solvers (see Section 5), which can be used as black boxes with some customisation. Another advantage is that many of these heuristics provide a lower bound on the optimal solution. Therefore, they provide guidance for the assessment of the quality of the solution. On the other hand, these heuristics have much more computational complexity for real-world problems, and due to their technical concepts cannot be implemented easily by practitioners.

*Relaxation Heuristics*: Heuristics of the relaxation category rely upon a relaxation of capacity constraints. These heuristics are very popular and have been implemented by many researchers. By relaxing capacity constraints, the problem reduces to $N$ single item uncapacitated problems. These single item problems then can be solved using Wagner–Whitin (or any other uncapacitated single item) algorithm.

Thizy and Van Wassenhove [50], using Lagrangian relaxation for capacity constraints, decomposed the problem into $N$ single item uncapacitated lot sizing sub-problems, solvable by the Wagner–Whitin algorithm. The solution of the relaxed problem (dual solution) is a lower bound on the original problem. By fixing the setup variables given by the dual solution, a transportation problem will result, which provides an upper bound. Then the Lagrangian multipliers are updated using the well-known subgradient optimisation procedure of Held et al. [73]. This process repeats until the lower bound equals the upper bound or a pre-specified number of iterations are completed. Contrary to the Newson [48] algorithm and many other mathematical programming based heuristics, their solutions are not restricted to solutions with Wagner–Whitin extreme point conditions.

The algorithm of Trigeiro [52] basically is the same as Thizy and Van Wassenhove's. The difference and main contribution in Trigeiro's algorithm is the use of a smoothing procedure to construct a feasible schedule from the dual solution to the original problem. In the smoothing routine both forward and backward passes are used. At each iteration, a forward pass yields cumulative feasibility, and then its backward pass finds a feasible schedule, if one exists, since the cumulative feasibility is always satisfied. A Lagrangian cost criterion is implemented to determine which item to move and how much to shift.

The works of other authors such as Billington et al. [49], Trigeiro et al. [53], and Diaby et al. [55], all follow the same basic approach as Thizy and Van Wassenhove [50].

Bitran and Matsuo [51] gave error bounds for the relaxation method. In their research, they concluded that as the number of products increase, the relative difference between the capacity-relaxed problem and the original problem becomes negligible.

Millar and Yang [56] presented two algorithms for solving a network-based formulation of the capacitated multi-item lot sizing problem with backordering. They implemented Lagrangian decomposition and Lagrangian relaxation techniques. The algorithms guarantee finding a feasible schedule and also provide an obvious measure of the quality of the solutions generated.

Chen and Thizy [31] gave a comprehensive analysis of relaxation methods for classical CLSP. They compared Lagrangian relaxation with other alternate relaxations of CLSP, and showed that Lagrangian relaxation is most precise in a rigorous sense. They also showed that the Lagrangian relaxation of the capacity constraints provides the tightest lower bound to the optimal solution, compared to the relaxation of other types. Thizy [54] also applied Lagrangian decomposition for the CLSP.

Relaxation techniques are more flexible, but worthy of mention is the fact that their useful properties may not be applicable for more complex production planning models, (e.g. multi-level problems or complex setup structures).

*Branch and bound based heuristics*: The branch and bound approach can be used as a general method for solving integer-programming problems, but solving large-scale CLSP problems by this approach is very time consuming needing a huge amount of computational effort. Therefore, this approach is not usually used as an optimal solution method for large problems. The previously mentioned relaxation heuristics can be used as a lower bound for the branch and bound procedure.

Gelders et al. [57] presented a branch and bound algorithm which finds the lower bound at each node based upon a Lagrangian relaxation of capacity constraints and subgradient optimisation. Again as mentioned in [50], at each iteration of the subgradient procedure, a primal transportation problem may yield a feasible solution, as an upper bound.

Diaby et al. [58] in their paper try to develop several new procedures for solving classic CLSP and CLSP with setup times, limited regular time and limited overtime. Their most successful procedure is a branch and bound method, which is based on Lagrangian relaxation and subgradient optimisation of capacity constraints.

Hindi [61] in his heuristic has implemented the branch and bound method as a solution strategy for CLSP. The heuristic first reformulates the problem as a shortest path problem via variable redefinition. The multi-item lower bound problems are solved by using a column generation strategy with capacity constraints as the linking constraints. The resulting single-item uncapacitated subproblems are solved as shortest path problems. Effective upper bounds, i.e. good feasible solutions for the original problem, are found by solving an appropriate minimum cost flow problem at each node of the branch and bound search tree.

Armentano et al. [62], represented multi item single-level CLSP with setup times as a minimum cost network flow problem. They used a branch and bound method (implicit enumeration) for solving this model, in which an initial solution is obtained by a heuristic approach. Other related research is Chung et al. [59], which combines dynamic programming and branch and bound to solve a single item CLSP. Another related work is Lotfi and Yoon [60].

*Set partitioning and column generation heuristics*: Manne [63] approximated a production scheduling problem by a linear programming model, which provides a good approximation when the number of distinct parts is large in comparison with the number of time periods. In this approximation, instead of individual combinations of item-period, complete production plans are considered. The original problem is handled by being partitioned into sets of *dominant production sequences*. Many heuristics are based on Manne's approximate representation of the CLSP, which is similar to a set-partitioning approach. For example, Cattrysse et al. [64], discussed set partitioning and column generation heuristics for the CLSP problem. They suggested three-set partitioning heuristics and a column generation heuristic. All their heuristics start from a feasible set of tentative schedules (generated by several well-known heuristics) for each individual item. Then by solving the LP relaxation of the set partitioning problem formulation, a subset of these schedules is chosen. At the final step they used a heuristic to convert the possibly fractional LP solution to an integer one, for the original problem. Chen and Thizy [31], have shown that the column generation technique often can obtain the Lagrangian optimal solution in less time than the relaxation heuristics.

The above-mentioned methods have some drawbacks, particularly when capacity constraints are particularly tight and also when the number of items is not substantially larger than the number of periods. These drawbacks limit their implementation in industry and since these methods are based on Manne's formulation, which only considers Wagner–Whitin schedules, they may not obtain any feasible solution even though the original problem may be feasible. Besides, it is possible that the true optimal solution may not always be of the Wagner–Whitin type. Trigeiro et al. [53] remark that the set-partitioning approaches tend to under account for the setup costs and times, because these are charged only once when a lot is split.

*Other approaches*: Among other research in the literature, there are some other works that used hybrid solution methods or approaches different in some aspects from previously discussed ones. Hindi [67], formulated the CLSP problem as a shortest path problem. The LP relaxation of this problem

Table 3
Summary of mathematical based heuristics

| Algorithm | Methods used in algorithm |
|---|---|
| Newson [48] | Lagrangian relaxation, shortest path, Wagner–Whitin |
| Thizy and Van Wassenhove [50] | Lagrangian relaxation, transportation, Wagner–Whitin, subgradient optimisation |
| Billington et al. [49] | Lagrangian relaxation, transportation, Wagner–Whitin, subgradient optimisation |
| Trigeiro [52] | Lagrangian relaxation, Wagner–Whitin, subgradient optimisation, smoothing |
| Trigeiro et al. [53] | Lagrangian relaxation, transportation, Wagner–Whitin, subgradient optimisation |
| Diaby et al. [58] | Lagrangian relaxation, transportation, Wagner–Whitin, subgradient optimisation |
| Bitran and Matsuo [51] | Various relaxations |
| Millar and Yang [56] | Network formulation, Lagrangian relaxation, Lagrangian decomposition |
| Chen and Thizy [31] | Relaxation methods |
| Thizy [54] | Lagrangian decomposition |
| Gelders et al. [57] | Branch and bound, Lagrangian relaxation |
| Diaby et al. [58] | Branch and bound, Lagrangian relaxation, subgradient optimisation |
| Hindi [61] | Branch and bound, shortest path, column generation, minimum cost network flow |
| Armentano et al. [62] | Branch and bound, minimum cost network flow |
| Chung et al. [59] | Branch and bound, dynamic programming |
| Lotfi and Yoon [60] | Branch and bound |
| Manne [63] | Set partitioning, linear programming |
| Cattrysse et al. [64] | Set partitioning, column generation |
| Lozano et al. [65] | Primal–dual, relaxation |
| Hindi [66] | Tabu search, transshipment |
| Hindi [67] | Shortest path, column generation, minimum cost network flow, tabu search |
| Hung and Hu [68] | Shadow prices |

is then solved by column generation. This relaxation yields a feasible solution, which is further improved by adopting the corresponding setup schedules and re-optimising variable costs by solving a minimum-cost network flow problem. Finally, this improved solution is used by the tabu search method as a starting solution to obtain better solutions. The initial feasible solution used by tabu search is also compared with Lagrangian relaxation and set-partitioning feasible solutions. More recent work by Hindi et al. [74] to solve multi-item CLSPs with setup times uses Lagrangian relaxation with subgradient optimization combined with a smoothing heuristic and local search. The method first forms good feasible solutions (upper bounds) and then improves on these.

Lozano et al. [65] used a primal–dual approach method to solve a Lagrangian relaxation of CLSP. The approach works as the steepest ascent method. To obtain a feasible solution in each iteration, a heuristic routine is used. Their computational experiences show that the method usually yields better solutions than the subgradient method, although it requires greater CPU time. Some other works are Hindi [66], and Hung and Hu [68]. Table 3 summarises the mathematical based heuristics.

## 5. Software

For approaches to solving CLSP using mathematical programming, especially integer programming, the popular solvers are CPLEX (www.ilog.com) and XPRESS-MP (www.dash.co.uk). *bc-prod* developed by Belvaux and Wolsey [36] and updated in [37] provides a complete framework for solving problems using the integer programming solver XPRESS-MP. The framework allows model development and the addition of cuts.

Codes are available for approximate methods. Shaw and Wagelmanns [75] have developed a code called ELSP for single-item capacitated economic lot sizing with piecewise linear production costs and general holding costs. The code is available from the authors for testing.

## 6. Conclusions

In this work, after a brief introduction in Section 1, in Section 2 we described factors which cause different variants of production planning problem and affect its complexity. The variants are discussed in Section 3, and in Section 4, among different variants, the single-level lot sizing problem, in both uncapacitated and capacitated cases is discussed. Basic formulations are presented and, based on different solution approaches, the literature and its classification is reviewed in detail. The focus has been more on the capacitated variant, termed CLSP in the literature.

Although the traditional CLSP problem has been studied by many authors, looking for more efficient solution approaches is a challenging subject. Variants of the CLSP with complex setup and other variants which are more realistic and practical have received less attention in the literature. There has been little literature regarding problems such as CLSP with backlogging or with setup times and setup carry-over. Since these problems are NP-hard, fast and efficient heuristics are required. Also there is little literature for problems such as CLSP with single-family or multi-family joint setup, in both capacitated and uncapacitated cases. Developing heuristics with reasonable speed and solution quality for these kinds of problems is another interesting research area.

Using some relatively new solution approaches such as tabu search, simulated annealing, and other meta-heuristics for solving CLSP is also another fruitful area of research. Such techniques have been shown to be effective for similar NP-hard problems.

## References


[1] Bahl HC, Ritzman LP, Gupta JND. Determining lot sizes and resource requirements: a review. Operations Research 1987;35(3):329–45.

[2] Baker KR. Lot-sizing procedures and a standard data set: a reconciliation of the literature. Journal of Manufacturing and Operations Management 1990;2:199–221.

[3] van Hoesel CPM, Wagelmans APM. Fully polynomial approximation schemes for single-item capacitated economic lot-sizing problems. Mathematics of Operations Research 2001;26(2):339–57.

[4] Wee HM. Deteriorating inventory model with quantity discount, pricing and partial backordering. International Journal of Production Economics 1999;59:511–8.

[5] Gelders LF, Van Wassenhove LN. Production planning: a review. European Journal of Operational Research 1981;7(2):101–10.

[6] Gallego G, Shaw DX. Complexity of the ELSP with general cyclic schedules. IEEE Transactions 1997;29:109–13.

[7] Jordan C, Drexl A. Discrete lot-sizing and scheduling by batch sequencing. Management Science 1998;44(5):698–713.

[8] Bruggemann W, Jahnke H. The discrete lot-sizing and scheduling problem: complexity and modification for batch availability. European Journal of Operational Research 2000;124(3):511–28.

[9] Drexl A, Kimms A. Lot sizing and scheduling-survey and extensions. European Journal of Operational Research 1997;99(2):228–49.

[10] Drexl A, Haase K. Proportional lot sizing and scheduling. International Journal of Production Economics 1995;40: 73–87.

[11] Fleischmann B, Meyr H. The general lot sizing and scheduling problem. OR Spektrum 1997;19(1):11–21.

[12] Meyr H. Simultaneous lot-sizing and scheduling combining local search with dual reoptimisation. European Journal of Operational Research 2000;120(2):311–26.

[13] Fisher ML, Ramdas K, Zheng YS. Ending inventory valuation in multi-period production scheduling. Management Science 2001;47(5):679–92.

[14] Potts C, Kovalyov MY. Scheduling with batching: a review. European Journal of Operational Research 2000;120(2): 228–49.

[15] Kirca O, Kokten M. A new heuristic approach for the multi-item dynamic lot sizing problem. European Journal of Operational Research 1994;75(2):332–41.

[16] Harris FW. How many parts to make at once. Factory, The Magazine of Management 1913;10(2):135–6.

[17] Wilson RH. A scientific routine for stock control. Harvard Business Review 1934;13(1):116–28.

[18] Wagner HM, Whitin TM. A dynamic version of the economic lot size model. Management Science 1958;5(1): 89–96.

[19] Wagelmans A, van Hoesel S, Kolen A. Economic lotsizing: an $O(n \log n)$ algorithm that runs in linear time in the Wagner–Whitin case. Operations Research 1992;40:S145–55.

[20] Aggarwal A, Park JK. Improved algorithms for economic lot-size problem. Operations Research 1993;41(3): 549–71.

[21] Federgruen A, Tzur MA. A simple forward algorithm to solve general dynamic lot sizing models with $n$ periods in $O(n \log n)$ or $O(n)$ time. Management Science 1991;37(8): 909–25.

[22] Wolsey LA. Progress with single-item lot-sizing. European Journal of Operational Research 1995;86(3):395–401.

[23] Stadtler H. Improved rolling schedules for the dynamic single-level lot-sizing problem. Management Science 2000;46(2):318–26.

[24] Tempelmeier H, Helber S. A heuristic for dynamic multi-item multi-level capacitated lotsizing for general product structures. European Journal of Operational Research 1994;75(2): 296–311.

[25] Stadtler H. Reformulations of the shortest route model for dynamic multi-item multi-level capacitated lotsizing. OR Spektrum 1997;19:87–96.

[26] Chand S. Lot sizing for products with finite demand horizon and periodic review inventory policy. European Journal of Operational Research 1982;11:145–8.

[27] Ganas IS, Papachristos S. Analytical evaluation of heuristic performance for the single-level lot-sizing problem for products with constant demand. International Journal of Production Economics 1997;48:129–39.

[28] Papachristos S, Ganas I. Optimal policy and stability regions for the single product periodic review inventory problem. Journal of the Operational Research Society 1998;49(2): 165–75.

[29] Florian M, Lenstra JK, Rinnooy Kan AHG. Deterministic production planning algorithms and complexity. Management Science 1980;26(7):669–79.

[30] Bitran GR, Yanasse HH. Computational complexity of the capacitated lot size problem. Management Science 1982;28(10):1174–86.

[31] Chen WH, Thizy JM. Analysis of relaxations for the multi-item capacitated lot-sizing problem. Annals of Operations Research 1990;26:29–72.

[32] Maes J, McClain JO, Van Wassenhove LN. Multilevel capacitated lot sizing complexity and LP-based heuristics. European Journal of Operational Research 1991;53(2): 131–48.

[33] Barany I, Van Roy TJ, Wolsey LA. Strong formulations for multi-item capacitated lot sizing. Management Science 1984;30(10):1255–61.

[34] Eppen GD, Martin RK. Solving multi-item capacitated lot sizing problems using variable redefinition. Operations Research 1987;35(6):832–48.

[35] Leung JMY, Magnanti TL, Vachani R. Facets and algorithms for capacitated lot sizing. Mathematical Programming 1989;45:331–59.

[36] Belvaux G, Wolsey LA. Bc-prod: a specialised branch-and-cut system for lot-sizing problems. Management Science 2000;46(5):993–1007.

[37] Belvaux G, Wolsey LA. Modelling practical lot-sizing problems as mixed integer programs. Management Science 2001;47(7):724–38.

[38] Fatemi Ghomi SMT, Hashemin SS. An analytical method for single level-constrained resources production problem with constant set-up cost. Iranian Journal of Science & Technology, Transaction B 2002;26(B1):69–82.

[39] Eisenhut PS. A dynamic lot sizing algorithm with capacity constraints. AIIE Transactions 1975;7(2):170–6.

[40] Lambrecht MR, Vanderveken H. Heuristic procedures for the single operation, multi-item loading problem. AIIE Transactions 1979;15(4):319–25.

[41] Dixon PS, Silver EA. A heuristic solution procedure for the multi-item, single level, limited capacity, lot sizing problem. Journal of Operations Management 1981;21(1):23–40.

[42] Maes J, Van Wassenhove LN. A simple heuristic for the multi-item single level capacitated lot sizing problem. Operations Research Letters 1986a;4(6):265–73.

[43] Dogramaci A, Panayiotopoulos JC, Adam NR. The dynamic lot-sizing problem for the multiple items under limited capacity. AIIE Transactions 1981;13(4):294–303.

[44] Karni R, Roll Y. A heuristic algorithm for the multi-item lot sizing problem with capacity constraints. AIIE Transactions 1982;14(4):249–59.

[45] Gunther HO. Planning lot sizes and capacity requirements in a single stage production systems. European Journal of Operational Research 1987;31(2):223–31.

[46] Selen WJ, Heuts RM. A modified priority index for Gunther's lot-sizing heuristic under capacitated single stage production. European Journal of Operational Research 1989;41: 181–5.

[47] Trigeiro WW. A simple heuristic for lot-sizing with setup times. Decision Science 1989;20:294–303.

[48] Newson EF. Multi-item lot size scheduling by heuristic, part I: with fixed resources. Management Science 1975;21(10): 1186–93.

[49] Billington PS, McClain JO, Thomas LJ. Mathematical Programming approaches to capacity-constrained MRP systems: review formulation and problem reduction. Management Science 1983;29(10):1126–41.

[50] Thizy JM, Van Wassenhove LN. Lagrangean relaxation for the multi-item capacitated lot-sizing problem: a heuristic implementation. IIE Transactions 1985;17(4):308–13.

[51] Bitran GR, Matsuo H. The multi-item capacitated lot-sizing problem: error bounds of Manne's formulations. Management Science 1986;32(3):350–9.

[52] Trigeiro WW. A dual-cost heuristic for the capacitated lot sizing problem. IIE Transactions 1987;19(3):67–72.

[53] Trigeiro WW, Thomas LJ, McClain JO. Capacitated lot sizing with setup times. Management Science 1989;35(3): 353–66.

[54] Thizy JM. Analysis of the Lagrangian decomposition for the multi-item capacitated lot-sizing problem. INFOR 1991;29(4):271–83.

[55] Diaby M, Bahl HC, Karwan MH, Zionts S. A Lagrangean relaxation approach for very-large-scale capacitated lot-sizing. Management Science 1992a;38(9):1329–40.

[56] Millar HH, Yang M. Lagrangian heuristics for the capacitated multi-item lot-sizing problem with backordering. International Journal of Production Economics 1994;34:1–15.

[57] Gelders LF, Maes J, Van Wassenhove LN. A branch and bound algorithm for the multi item single level capacitated dynamic lotsizing problem. In: Axaster S, Schneeweiss CH, Silver E, editors. Multistage production planning and inventory control. Lecture notes in economics and mathematical systems, vol. 266. Berlin: Springer, 1986. p. 92–108.

[58] Diaby M, Bahl HC, Karwan MH, Zionts S. Capacitated lot-sizing and scheduling by Lagrangean relaxation. European Journal of Operational Research 1992b;59(3):444–58.

[59] Chung C, Flynn J, Lin CM. An effective algorithm for the capacitated single item lot size problem. European Journal of Operational Research 1994;75(2):427–40.

[60] Lotfi V, Yoon Y. An algorithm for the single-item capacitated lot sizing problem with concave production and holding costs. Journal of the Operational Research Society 1994;45(8): 934–41.

[61] Hindi KS. Computationally efficient solution of multi-item capacitated lot sizing problems. Computers and Industrial Engineering 1995a;28(4):709–19.

[62] Armentano VA, Franca PM, de Toledo FMB. A network flow model for the capacitated lot-sizing problem. OMEGA 1999;27(2):275–84.

[63] Manne AS. Programming of economic lot sizes. Management Science 1958;4(2):115–35.

[64] Cattrysse D, Maes J, Van Wassenhove LN. Set partitioning and column generation heuristics for capacitated dynamic lot sizing. European Journal of Operational Research 1990;46(1):38–47.

[65] Lozano S, Larraneta J, Onieva L. Primal–dual approach to the single level capacitated lot-sizing problem. European Journal of Operational Research 1991;51(3):354–66.

[66] Hindi KS. Solving the single-item, capacitated dynamic lot-sizing problem with startup and reservation costs by tabu search. Computers and Industrial Engineering 1995b;28(4): 701–7.

[67] Hindi KS. Solving the CLSP by a tabu search heuristic. Journal of the Operational Research Society 1996;47(1):151–61.

[68] Hung Y, Hu Y. Solving mixed integer programming production planning problems with setups by shadow price information. Computers and Operations Research 1998;25(12):1027–42.

[69] Maes J, Van Wassenhove LN. Multi-item single-level capacitated dynamic lot-sizing heuristics: a general review. Journal of the Operational Research Society 1988;39(11): 991–1004.

[70] Silver EA, Meal HC. A heuristic for selecting lot size quantities for the case of a deterministic time varying demand rate and discrete opportunities for replenishment. Production and Inventory Management 1973;14(2):64–74.

[71] Groff GK. A lot sizing rule for time-phased component demand. Production and Inventory Management 1979;20: 47–53.

[72] Maes J, Van Wassenhove LN. Multi item single level capacitated dynamic lotsizing heuristics: a computational

comparison—Part I: static case, and Part II: rolling horizon. IEE Transactions 1986;18(2):114–29.

[73] Held M, Wolfe P, Crowder HP. Validation of subgradient optimization. Mathematical Programming 1974;6(1):62–88.

[74] Hindi KS, Fleszar K, Charalambous C. An effective heuristic for the CLSP with setup times. Research Paper 2001, Department of Systems Engineering, Brunel University, UB8 3PH, England, 2001.

[75] Shaw DX, Wagelmans APM. An algorithm for single-item capacitated economic lot sizing with piecewise linear production costs and general holding costs. Management Science 1998;44(6):831–8.