OR Spectrum (2010) 32:231–261
DOI 10.1007/s00291-008-0150-7

REGULAR ARTICLE

Dynamic capacitated lot-sizing problems: a classification and review of solution approaches

Lisbeth Buschkühl · Florian Sahling · Stefan Helber · Horst Tempelmeier

Published online: 22 October 2008

Abstract This paper presents a review of four decades of research on dynamic lot-sizing with capacity constraints. We discuss both different modeling approaches to the optimization problems and different algorithmic solution approaches. The focus is on research that separates the lot-sizing problem from the detailed sequencing and scheduling problem. Our conceptional point of reference is the multi-level capacitated lot-sizing problem (MLCLSP). We show how different streams of research emerged over time. One result is that many practically important problems are still far from being solved in the sense that they could routinely be solved close to optimality in industrial practice. Our review also shows that currently mathematical programing and the use of metaheuristics are particularly popular among researchers in a vivid and flourishing field of research.

Keywords Dynamic capacitated lot-sizing · MLCLSP · CLSP · CLSPL · Mathematical programing · Lagrangian relaxation · Decomposition · Metaheuristics · Greedy heuristics

L. Buschkühl (✉) · H. Tempelmeier
Seminar für Supply Chain Management und Produktion, Universität zu Köln,
Albertus-Magnus-Platz, 50923 Cologne, Germany
e-mail: buschkuehl@wiso.uni-koeln.de

H. Tempelmeier
e-mail: tempelmeier@wiso.uni-koeln.de

F. Sahling · S. Helber
Institut für Produktionswirtschaft, Leibniz Universität Hannover,
Königsworther Platz 1, 30167 Hannover, Germany
e-mail: sahling@prod.uni-hannover.de

S. Helber
e-mail: helber@prod.uni-hannover.de

1 Introduction

Many production processes can only start after the required resources have been set up. This setup usually requires a setup time and/or causes setup cost. As a consequence, a lot-sizing problem arises, because a decision has to be made whether future demands should be produced to stock to save setups. Research on lot-sizing dates back to the early twentieth century, and a large number of different lot-sizing problems have been identified, for which an even larger number of modeling approaches and algorithms have been developed.

Production planning and particularly lot-sizing is strongly related to the layout type and organizational structure of a production system. In industry, many different kinds of production systems are found which have a significant impact on the type of lot-sizing model applicable in a certain planning environment. Lot-sizing problems mainly arise in job-shop production as well as flow production systems. In each type of production system, specific lot-sizing problems emerge for which the literature provides appropriate modeling and solution approaches. For example, in job-shop production systems where setup times are relevant, the dynamic multi-level capacitated lot-sizing problem (MLCLSP) arises. This problem is also the theoretical basis of the standard material requirements (MRP) calculations. Unfortunately, standard MRP (and MRP II) software systems overemphasize the demand explosion part of the problem and almost completely ignore the limited capacity of resources when dealing with the lot-sizing part, if lot-sizing is considered at all. The resulting inability to produce capacity-feasible plans is one of the reasons why the standard MRP approach often fails in industrial practice.

In a different layout type, e.g., when several production stages are arranged serially to produce multiple products with significant setup times, the static economic lot scheduling problem (ELSP), see Rogers (1958), or one of its dynamic counterparts can be applied, for example the continuous setup lot-sizing problem (CSLP), see Karmarkar and Schrage (1985), the dynamic lot-sizing and scheduling problem (DLSP), see Fleischmann (1990), the proportional lot-sizing and scheduling problem (PLSP), see Drexl and Haase (1995), or the general lot-sizing and scheduling problem (GLSP), see Fleischmann and Meyr (1997), just to name a few. All these different lot-sizing problems or rather lot-sizing models require specific solution techniques.

Numerous reviews of solution approaches to lot-sizing, which cover different aspects of the planning problem, exist. Bahl et al. (1987) discussed the subsets of unconstrained and constrained single- and multi-level lot-sizing. Maes and van Wassenhove (1988) gave a structured overview of lot-sizing heuristics to the CLSP and carried out a computational study. Gupta and Keung (1990) reviewed uncapacitated single- and multi-stage lot-sizing models. Salomon et al. (1991) gave an overview of several dynamic capacitated lot-sizing models and some solution algorithms. Kuik et al. (1994) studied the impact of lot-sizing and batching and responded to some general criticism of batching analysis. Wolsey (1995) and Brahimi et al. (2006b) reviewed solution methods for single-item lot-sizing problems. Drexl and Kimms (1997) discussed simultaneous lot-sizing and scheduling models. Staggemeier and Clark (2001) reviewed metaheuristics applied to the solution of lot-sizing and scheduling problems. Karimi et al. (2003) considered solution approaches to single-stage capacitated lot-sizing problems. Jans and Degraeve (2007) gave a review on metaheuristics for dynamic lot-sizing. Quadt and Kuhn (2008) reviewed capacitated lot-sizing problems with extensions.

The remainder of this paper is structured as follows. In Sect. 2, we present the standard mathematical formulation of the MLCLSP and discuss variations of this model w.r.t. the scope of problem aspects covered. Section 3, which is the main part of the paper, provides a structured discussion of the different solution approaches. We start in Sect. 3.1 with mathematical programing-based heuristics that use reformulations and restructuring techniques to make an MIP formulation manageable by a standard MIP solver. Then, in Sect. 3.2, Lagrangian heuristics are discussed, followed by decomposition and aggregation approaches in Sect. 3.3. Section 3.4 covers metaheuristics and finally in Sect. 3.5 common sense (greedy) approaches are presented. Although arguable in individual cases, our classification focusses on the solution approaches to solve these problems. The paper ends with some concluding remarks.

2 The standard model formulation

2.1 The multi-level capacitated lot-sizing problem

The dynamic multi-level capacitated lot-sizing problem (MLCLSP) was introduced by Billington et al. (1983). It describes the following scenario. The planning horizon is finite and divided into T discrete time periods (e.g. weeks). There are K items with period-specific external demands which must be met without delay. The items are produced on M non-identical resources with limited period-specific capacities. Each resource comprises one or more resource units, such as similar machines or workers, which are treated as a single entity.

The capacity of a resource per period is thus the product of the number of units and the net available time. For each item, a unique assignment to a single resource exists. Therefore, an item can also be interpreted as the result of an operation. This operation is part of a process plan describing how to produce the item. The items or rather operations are interrelated through input–output relationships. The production of each unit of an item takes a constant amount of processing time. Whenever the production quantity in a period is greater than zero, the model assumes that a given setup time is incurred—independent of the number of resource units actually required to process the production lot. In addition, setup cost may result. Setup times and cost are assumed to be independent of the sequence of products during a period. A specific characteristic of the standard MLCLSP is that each positive production quantity during a period induces a setup, even in cases where the production of an item takes place in two consecutive periods.

Numerous model formulations have been proposed for the MLCLSP, to allow for an efficient numerical solution. These formulations differ mainly in the type of variables used. The earliest formulation uses production quantities and inventory levels as variables. This so-called inventory and lot-size (I&L) formulation will be described first. Throughout the paper, we use the notation given in Table 1.

The $\mathrm{MLCLSP}_{\mathrm{I\&L}}$ reads as follows:

Table 1 Notation

Index sets	
$\mathcal{K}$	Set of items, $\mathcal{K} = \{1, \ldots, K\}$
$\mathcal{M}$	Set of resource groups, $\mathcal{M} = \{1, \ldots, M\}$
$\mathcal{T}$	Set of periods, $\mathcal{T} = \{1, \ldots, T\}$
$\mathcal{K}_m$	Set of items k produced on resource m
$\mathcal{S}_k$	Set of direct successors of item k
Parameters	
a_{kj}	Quantity of item k directly required to produce one unit of item j (Gozinto factor)
c_{mt}	Available capacity of resource m in period t
d_{kt}	External demand of item k in period t
h_k	Holding cost of item k per unit and period
s_k	Setup cost of item k
tp_k	Production time per unit of item k
ts_k	Setup time for the production of item k
z_k	Planned lead time of item k
b_{kt}	Sufficiently big number
Variables	
γ_{kt}	Binary setup variable of item k in period t
Q_{kt}	Production quantity of item k in period t
Y_{kt}	Inventory of item k at the end of period t

Model MLCLSP$_{\mathrm{I\&L}}$

$$\min \ Z = \sum_{k \in \mathcal{K}} \sum_{t \in \mathcal{T}} (s_k \cdot \gamma_{kt} + h_k \cdot Y_{kt}) \tag{1}$$

subject to

$$Y_{k,t-1} + Q_{k,t-z_k} - \sum_{j \in \mathcal{S}_k} a_{kj} \cdot Q_{jt} - Y_{kt} = d_{kt} \quad \forall k, t \tag{2}$$

$$\sum_{k \in \mathcal{K}_m} (tp_k \cdot Q_{kt} + ts_k \cdot \gamma_{kt}) \leq c_{mt} \quad \forall m, t \tag{3}$$

$$Q_{kt} \leq b_{kt} \cdot \gamma_{kt} \quad \forall k, t \tag{4}$$

$$Y_{k0} = Y_{kT} = 0 \quad \forall k \tag{5}$$

$$Q_{kt}, Y_{kt} \geq 0 \quad \forall k, t \tag{6}$$

$$\gamma_{kt} \in \{0, 1\} \quad \forall k, t \tag{7}$$

The objective function (1) minimizes the total sum of setup cost and inventory holding cost. Constraints (2) are the inventory balance constraints which guarantee that the external demand d_{kt} and the secondary demands $(\sum_{j \in \mathcal{S}_k} a_{kj} \cdot Q_{jt})$ of item k in every period t are met. For item k a planned lead time z_k is used. Historically, this planned lead time was used in non-capacitated lot-sizing models to account for waiting processes etc., during production. In the considered capacitated lot-sizing model, congestion in front of a resource is prevented by shifting production quantities backward. Nevertheless, a minimum planned lead time $z_k = 1$ is required for all component products to ensure that a feasible production schedule can always be generated based on the solution of the lot-sizing model. Consider a serial product structure with one end product that requires a single component, both produced on different resources. Assume that the resource for the end product is fully loaded to produce a single unit of the end product during a period. Then it is not feasible to produce a unit of the component in the same period such that it enters the unit of the end product. Thus, this unit of the component must be produced at least one period ahead, i.e. with a minimum lead time $z_k = 1$ of one period.

Constraints (3) are the capacity constraints concerning production and setup time for each resource m. Constraints (4) ensure that production of item k takes place in period t, only if the resource is setup for this item ($\gamma_{kt} = 1$). To guarantee this, the constant b_{kt} must be large enough not to restrict the lot-size of item k in period t. The initial inventory Y_{k0} and the final inventory Y_{kT} are assumed to be 0, according to constraint (5). Note that this setting makes production in the last periods of the planning horizon unattractive (see Stadtler 2000). Finally, the variables Q_{kt} and Y_{kt} must be non-negative [see constraints (6)] and the setup variables γ_{kt} are defined as binary variables in constraints (7). Variable production cost are usually not included as they are assumed to be constant over the whole planning horizon T.

2.2 Simplified lot-sizing models derived from the MLCLSP

If the sum describing the derived demand on the left-hand side of (2) is omitted, the MLCLSP decomposes into M (single-level) lot-sizing models known as the capacitated lot-sizing problem (CLSP).

If in addition the capacity constraints are relaxed, each CLSP further decomposes into K single-level uncapacitated lot-sizing models known as the Wagner–Whitin problem (see Wagner and Whitin 1958), also referred to as the single-level uncapacitated lot-sizing problem (SLULSP).

Relaxing the capacity constraints in model MLCLSP leads to the so-called multi-level uncapacitated lot-sizing problem (MLULSP) (see Jacobs and Khumawala 1982). If all binary setup variables γ_{kt} are fixed with respect to a given setup pattern, then a linear program (LP) is obtained, which can be solved easily (see Kuik and Salomon 1990; Millar and Yang 1994). Further extensions to the above lot-sizing problems have been presented. Some authors account for overtime and backorder decisions ensuring a feasible solution in a mathematical sense, as the corresponding decision variables have the function of slack variables. Further extensions introduce, e.g., parallel machines each considered as a distinct resource.

2.3 Setup carryovers

As the above model formulation does not provide any information about the sequence of production within a period, each production quantity of an item assigned to a period is assumed to induce a setup. However, in reality it might be possible to continue the production of the same item at the beginning of the next period without a setup. Without changing to a complete lot-sizing and scheduling perspective of the PLSP or GLSP type, the above models can be extended through the introduction of additional setup state variables to determine the products processed at the period borders, which introduces partial sequencing decisions into the model.

Dillenberger et al. (1993) extended the CLSP by this aspect. The term capacitated lot-sizing problem with linked lot-sizes (or setup carryovers) (CLSPL) has been coined by Haase (1994). Later, Sürie and Stadtler (2003) extended the CLSPL to the multi-stage case (MLCLSPL). Note that the notion of a setup state requires the consideration of individual resource units.

To model setup carryovers, additional variables and constraints are used. Binary setup state variables ω_{kt} $(\forall k, t)$ are introduced reflecting the setup state of the resource at the beginning of period t.

Note that $\omega_{kt} = 1$ implies that product k is the first one to be produced in period t using the setup state carried over from period $(t-1)$.

Two cases can be distinguished. First, a setup carryover can only be allowed, if a corresponding setup operation has taken place in the directly preceding period. Second, if consecutive setup carryovers are allowed, the setup state can be transferred across several consecutive periods without additional setup operations. In the latter case, no other setup activity is allowed which would suspend the current setup state. Hence, dummy variables ν_{mt} are introduced. $\nu_{mt} = 1$ indicates that a setup state for one item is carried over from a period $t-1$ to the consecutive periods t and $t+1$ on resource m. In the case of consecutive carryovers, the additional constraints are (Sürie and Stadtler 2003):

$$Q_{kt} \leq b_{kt} \cdot (\gamma_{kt} + \omega_{kt}) \quad \forall k, t \tag{8}$$

$$\sum_{k \in \mathcal{K}_m} \omega_{kt} \leq 1 \quad \forall m, t = 2, \ldots, T \tag{9}$$

$$\omega_{kt} \leq \gamma_{k,t-1} + \omega_{k,t-1} \quad \forall k, t = 2, \ldots, T \tag{10}$$

$$\omega_{k,t-1} + \omega_{kt} \leq 1 + \nu_{mt} \quad \forall m, k \in \mathcal{K}_m, t = 2, \ldots, T \tag{11}$$

$$\gamma_{kt} + \nu_{mt} \leq 1 \quad \forall m, k \in \mathcal{K}_m, t \tag{12}$$

$$\nu_{mt} \geq 0 \quad \forall m, t \tag{13}$$

$$\omega_{kt} \in \{0, 1\} \quad \forall k, t \tag{14}$$

Except for the additional constraints (9)–(14), the only change to the original formulation of the I&L formulation of the MLCLSP is in restrictions (8), which replace restrictions (4). They enforce a setup for every positive production quantity, unless the particular setup state has already been preserved from the preceding period.

For each resource m, at most one setup state can be carried over per period [constraints (9)]. The constraints (10) ensure that a setup can only be carried over into period t if either a setup occurred in period $t-1$ or if the setup state had already been carried over from period $t-2$ to period $t-1$. In the latter case no other item may be produced in the period $t-1$ on this resource.

This is guaranteed by constraints (11) and (12) and by the auxiliary variables ν_{mt}. If two consecutive setup carryovers for the same item occur “into” and “out of” period t, the variable ν_{mt} is forced to 1 and hence all $\gamma_{kt} = 0$. However, if there is at least one γ_{kt} equal 1, ν_{mt} is forced to 0 [due to (12)]. Hence, at most one setup carryover can take place for another item. If consecutive setup carryovers are not allowed, constraints (10) are changed in that $\omega_{k,t-1}$ is removed from the right-hand side and constraints (11) and (12) are omitted.

Florian et al. (1980) have proved that the single-item CLSP is $\mathcal{NP}$-hard. Later, Bitran and Yanasse (1982) showed that even special cases which are solvable in polynomial time become $\mathcal{NP}$-hard through the introduction of a second item. Trigeiro et al. (1989) pointed out that for the CLSP with setup times the question whether a feasible schedule *exists* is already $\mathcal{NP}$-complete. Trigeiro et al. (1989) referred to bin packing as a special case of the CLSP with setup times. For the proof of $\mathcal{NP}$-completeness see Garey and Johnson (1979). As the MLCLSP and the MLCLSPL can be reduced to the CLSP with setup times by setting some parameters to 0, they are at least as hard to solve and hence also $\mathcal{NP}$-hard.

In addition to the standard I&L model formulation, several reformulations have been proposed which are closely related to specific solution approaches. These reformulations will be discussed in the following within the context of the associated solution approach.

3 Solution approaches

The approaches to solve the different types of capacitated lot-sizing models can be classified into five groups, as shown in Fig. 1.

3.1 Mathematical programing-based approaches

The MP-based approaches are rather general and tend to be more flexible than other procedures with respect to model extensions. One can distinguish between exact methods and MP-based heuristics. If an optimal solution exists, the former stop after an optimal solution has been found, regardless of the effort in terms of required computation time and memory. The latter only explore parts of the solution space and try to find a good feasible solution in reasonable time. Imposing a time-limit on an exact method is a simple way to construct an MP-based heuristic. Even if the solution obtained within the time limit is infeasible, it can still serve as a possibly promising starting point to construct a feasible solution. Table 2 gives a systematic overview of the MP-based solution approaches.

Branch and bound The branch and bound method (B&B) is an exact solution procedure that enumerates feasible solutions implicitly. This method consists of two parts, “branching” and “bounding”. While during the “branching” part new disjoint subsets of the solution space are generated, unpromising ones are removed during the “bounding” part. For mixed integer programs (MIPs) with binary variables, as considered here, branching is based on subsequently fixing the binary variables to 0 and 1.

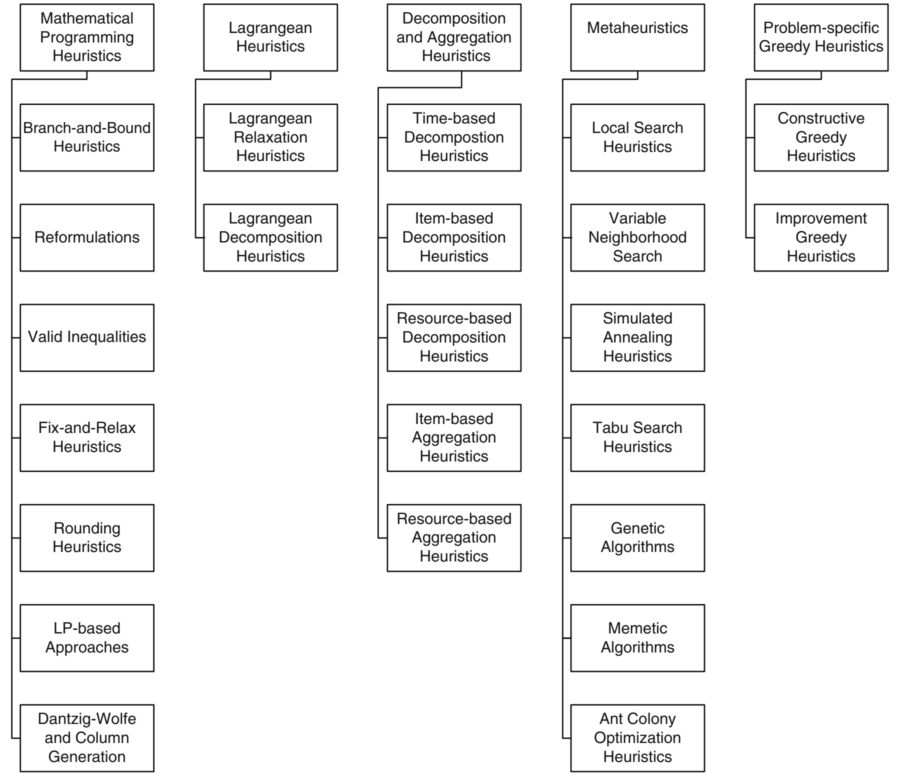

Fig. 1 Classification of solution approaches

A relaxed version of each subproblem is solved to determine a bound. Different options are proposed to relax the model formulation. First, in the LP relaxation the integrality constraint of the original binary variables of the MIP is removed. Another relaxation method is Lagrangian relaxation (see below). B&B methods imbedded into a Lagrangian relaxation scheme are described in Billington et al. (1986), Gelders et al. (1986), Chen and Thizy (1990) and Diaby et al. (1992a). Armentano et al. (1999) used a B&B procedure to solve the CLSP. Here, the corresponding LP relaxation is based on a network flow model.

One major research trend in the field of lot-sizing is to reformulate the mathematical model, to redefine the corresponding decision variables and to introduce valid inequalities. As the bounds obtained by relaxing the aforementioned I&L formulation are quite poor, the goal is to tighten the lower bounds and thus to increase the efficiency of the B&B method.

Reformulations Two reformulations have been introduced which assign each production quantity to a corresponding demand quantity. First, the shortest route (SR) formulation was introduced by Eppen and Martin (1987) for the CLSP. Tempelmeier and Helber (1994) extended this formulation to the capacitated multi-level case. Stadtler (1996, 1997) suggested an improved SR formulation which

Table 2 MP-based heuristics

Reference	FORM	REL	SolA	MOD
Armentano et al. (1999)	I&L	LP	B&B, NFA	SL
Billington et al. (1986)	I&L	LR	B&B	ML, ST
Diaby et al. (1992a)	I&L	LR	B&B	SL, ST, OV
Gelders et al. (1986)	I&L	LR	B&B	SL
Chen and Thizy (1990)	I&L, SR	LR	B&B	SL
Stadtler (1996)	I&L, SPL, SR	LP	B&B	ML, ST, OV
Stadtler (1997)	SR	LP	B&B	ML, ST, OV
Tempelmeier and Helber (1994)	SR	LP	B&B	ML
Barany et al. (1984)	I&L	LP	B&B, VI	SL
Belvaux and Wolsey (2000)	I&L	LP	B&C, VI	ML, ST, BO
Belvaux and Wolsey (2001)	I&L	LP	B&C, VI	ML, ST, BO
Clark and Armentano (1995)	I&L	LP	B&C, VI	ML, ST
Miller et al. (2000)	I&L	LP	B&C, VI	SL, ST
Pochet and Wolsey (1991)	I&L	LP	B&B, VI	ML, ST
Sürie and Stadtler (2003)	SPL	LP	B&C, C&B, VI	SL, ML, ST, SC
Dillenberger et al. (1993)	I&L	LP	F&R	SL, PM, BO, SC
Dillenberger et al. (1994)	I&L	LP	F&R	SL, ST, PM, BO, SC
Federgruen et al. (2007)	I&L	LP	F&R	SL, JSC
Mercé and Fontan (2003)	I&L	LP	F&R	SL, ST, BO
Stadtler (2003)	SPL	LP	F&R	ML, ST, OV
Akartunalı and Miller (2008)	I&L	LP	RH, F&R, VI	ML, ST
Alfieri et al. (2002)	SPL, SR	LP	RH	SL
Eppen and Martin (1987)	SR	LP	RH	SL
Kuik et al. (1993)	SPL	LP	RH	ML
Maes et al. (1991)	SPL	LP	RH	ML
Salomon (1991)	SPL	LP	RH	ML
Harrison and Lewis (1996)	I&L		LPB	ML, ST, BO
Hung and Hu (1998)	I&L		LPB	SL, ST, BO
Katok et al. (1998)	I&L		LPB	ML, ST
Bahl (1983)	SPP		CG	SL, ST, OV
Bitran and Matsuo (1986)	SPP		DW	SL, ST
Cattrysse et al. (1990)	SPP		CG	SL
Degraeve and Jans (2003)	SPP		CG	SL, ST
Dzielinski and Gomory (1965)	SPP		DW	SL
Haase (2005)	SPP, SPL	LP	CG, F&R	SL
Hindi (1995)	SPP		CG	SL
Hindi (1996)	SPP		CG	SL
Huisman et al. (2003)	SPP		CG	SL, OV

Table 2 continued

Reference	FORM	REL	SolA	MOD
Lasdon and Terjung (1971)	SPP		CG	SL
Manne (1958)	SPP		DW	SL, OV
Salomon et al. (1993)	SPP		CG	SL, ST

Abbreviations
FORM Model formulation (I&L = inventory and lot-size, SPL = simple plant location, SPP = set partitioning problem, SR = shortest route)
REL Relaxation (LP = LP relaxation, LR = Lagrange relaxation)
SolA Solution approach (B&B = branch and bound, B&C = branch and cut, C&B = cut and branch, CG = column generation, DW = Dantzig–Wolfe decomposition, F&R = fix and relax, LPB = LP-based approach, NFA = network flow algorithm, RH = rounding heuristic, VI = valid inequalities)
MOD Problem solved (ML = multi-level, SL = single-level, BO = backorders, JSC = joint setup cost, OV = overtime, PM = parallel machines, SC = setup carryover, ST = setup time)

decreases the number of non-negative coefficients in the constraints' matrix, with the effect that the computational effort for solving the corresponding LP relaxation decreases as well. Another formulation is based on the analogy to the Simple Plant Location (SPL) problem. Rosling (1986) introduced the SPL formulation for the MLULSP as an extension of the work of Krarup and Bilde (1977). Later, Maes et al. (1991) used the model of Rosling (1986) with the inclusion of capacity constraints, whereby both lot-sizing models were limited to assembly-type bill-of-material structures. Stadtler (1996) extended the SPL formulation for the case of general bill-of-material structures. The LP relaxation of the SR formulation and of the SPL formulation have identical objective function values, see Denizel et al. (2008). The number of decision variables is also identical, but the number of constraints is not. For the SPL formulation $K \cdot T(\frac{1}{2}(T+1)-1)$ additional constraints are needed.

Valid inequalities Another possibility to tighten the bounds of the LP relaxation is to generate valid inequalities. They reduce the size of the solution space by cutting off irrelevant parts. Three methods can be distinguished. First, if the valid inequalities are generated dynamically to cut off current non-integer solutions, this is referred to as the cutting plane method. Second, valid inequalities can be introduced in the course of a B&B algorithm. This method is called branch and cut (B&C). Third, the cut and branch procedure (C&B) incorporates all generated inequalities into the model formulation prior to starting the B&B algorithm.

Barany et al. (1984) defined a set of valid inequalities. They included lot-sizing variables and inventory variables for the SLULSP. These additional inequalities describe the convex hull for the single-item uncapacitated lot-sizing polytope. They can also be applied to the CLSP. Furthermore, other valid inequalities are derived by Miller et al. (2000) for the capacitated problem. Pochet and Wolsey (1991) and Clark and Armentano (1995) extended the work of Barany et al. (1984) to the multi-level case.

Belvaux and Wolsey (2000) presented a general framework for modeling and solving lot-sizing problems. This system, called bc–prod, includes preprocessing especially for lot-sizing problems and generates lot-sizing specific and general cutting planes. Their work is extended in Belvaux and Wolsey (2001) to further model extensions, e.g., start-ups, changeovers and switch-offs.

Sürie and Stadtler (2003) derived valid inequalities for the CLSPL and the MLCLSPL. To do so, they introduced extended model formulations by redefining the setup carryover constraints. In addition, they performed a pre-processing step which leads to inequalities that link inventories to setups and single-item production to capacity. They adopted C&B and B&C with a given time limit to find the optimal or the first feasible solution. To reduce complexity, a time-oriented decomposition approach with overlapping planning windows (see Stadtler 2003) is applied.

Fix and relax heuristics Fix and relax (F&R) heuristics reduce the number of binary variables to be treated simultaneously by dividing the problem into several subproblems. Three sets of binary variables can be distinguished. The first is solved to optimality, the second is relaxed and the setup states of the third set are fixed to the values of a previous iteration.

Dillenberger et al. (1993) developed an F&R algorithm for the CLSPL. It essentially consists of a B&B algorithm, where the order of branching is determined by the sequence of periods, i.e. a period-by-period heuristic with binary setup variables in the first period of the current lot-sizing window, fixed setup variables in all prior periods and relaxed setup variables in all ongoing periods until the end of the planning horizon. Dillenberger et al. (1994) again applied the F&R heuristic to an extended model formulation.

Stadtler (2003) proposed to use overlapping planning windows. The binary variables preceding a given planning window have been determined in prior iterations and those after the end of the planning window are excluded from the currently solved model, thus capacity requirements are only approximated. This approach yields high-quality solutions for the MLCLSP, but is limited to problems without lead times. It is also applied to the CLSPL in Sürie and Stadtler (2003).

Mercé and Fontan (2003) proposed an MIP-based heuristic, which relies on the division of the planning horizon into several sub-horizons and different freezing methods for past decisions.

Federgruen et al. (2007) presented a so-called progressive interval heuristic for the CLSP with joint setup cost. Starting with a small subset of periods, all binary setup variables are solved to optimality within this time window. In each iteration, this time window is extended while only the binary variables of the last τ periods are determined to optimality. In contrast, the setup variables related to earlier periods are fixed. The heuristic stops when the end of the planning horizon is reached.

Rounding heuristics In rounding heuristics (RH), the LP relaxation of the MIP is solved and the fractional binary variables are subsequently rounded. In the case of capacitated lot-sizing problems, these solutions are often infeasible, as capacity may not be sufficient. Hence, during the RH the fractional binary variables are usually rounded up and rounded down only with respect to a given threshold.

Maes et al. (1991) introduced several RHs using the SPL formulation for the MLCLSP without setup-times. The relaxed setup-variable with the highest value is fixed to 1 and the reduced LP relaxation is solved. Furthermore, all relaxed binary variables within a given range can also be fixed to 1.

Eppen and Martin (1987) and Alfieri et al. (2002) presented RHs, which solve the LP relaxation of the SR formulation for the CLSP. All fractional binary variables are fixed within a given threshold, and a limited B&B is started with the remaining binary variables. Kuik et al. (1993) presented a further RH for the MLCLSP without setup times and one resource. Based on the LP relaxation of the SPL formulation, all binary variables within a given range are fixed either to 0 or 1. The remaining binary variables are initially set to 1 and a simulated annealing or Tabu search approach follows, where changes only affect the unfixed binary variables. A similar approach can be found in Salomon (1991).

Akartunalı and Miller (2008) combined a RH and F&R heuristic for the MLCLSP. In the first step, the LP relaxation of the I&L formulation with additional valid inequalities is solved. Thereby, the setup variables within a given interval are fixed. Subsequently, an F&R heuristic is applied using a moving time window.

LP-based approaches These iterative solution methods exploit the fact that omitting the binary variables leads to a significantly easier problem. Here, the complete setup pattern is either given or only implicitly accounted for, and the remaining LP is solved to optimality.

Hung and Hu (1998) presented an iterative solution approach for the CLSP, in which at each iteration the setup-pattern is fixed and the remaining LP is solved. They started with an initial setup-pattern assuming a production for every product in every period (lot-for-lot). Hence, all setup variables are fixed to 1. After solving the resulting linear program, they used the information of the shadow prices relating to the capacity constraints to identify products and periods for which it is beneficial to fix the corresponding setup variables to 0. Subsequently, the resulting LP is solved.

Harrison and Lewis (1996) presented the iterative coefficient modification heuristic (CMH) for the MLCLSP. They exploited the fact that in the MLCLSP each setup variable is linked to a corresponding continuous production variable, in that both variables must be equal to zero or positive at the same time. Therefore, the binary variables are omitted and setup times are accounted for implicitly via modification of the production time coefficients of the related production variables. This leads to a reduced linear program, as not only the binary variables but also the linking constraints to the continuous variables can be eliminated. In each iteration, the respective linear program is solved and the production time coefficients are modified according to the capacity consumption in the previous iteration. In a later paper, Katok et al. (1998) introduced the coefficient modification subroutine with cost balancing (CMSB), which is an extension to the CMH. Here, also the production cost coefficients are modified to account for setup cost implicitly.

Dantzig–Wolfe decomposition and column generation Dantzig–Wolfe decomposition (see Dantzig and Wolfe 1960) has been applied to the CLSP which is modeled as a set partitioning problem. The objective is to find a convex combination of given single-item schedules, which keeps the capacity constraints of the original CLSP and leads to minimal cost. The decision variables are continuous variables to combine schedules for each item.

To find the optimal solution to the set partitioning problem (SPP), column generation (CG) is applied. Column generation is a general iterative solution procedure for large-scale linear programs. In the case of mixed integer programs, CG can be applied to the relaxed problem to find promising lower bounds. In the current problem environment it is used to dynamically generate new schedules with the Wagner–Whitin property. The SPP serves as a master problem. The corresponding subproblem is to find the single-item schedules to feed into the master problem. For the CLSP, it consists of a set of SLULSPs. The CG procedure starts with a limited number of schedules for each product. At each iteration, first the SPP is solved. Then shadow prices related to the capacity and convexity constraints are updated and used to modify the objective function of the subproblems, which are subsequently resolved. The new schedules thus generated are introduced into the master problem, only if their objective function value is negative and therefore beneficial. The procedure is repeated until no more schedules that improve the objective function of the SPP can be generated. The approach stops with a promising lower bound for the CLSP. Afterward, an additional solution heuristic has to be applied to generate a feasible solution.

An early approach was introduced by Manne (1958). In an SPP, Manne selects from a large set of production plans satisfying the condition $q_t \cdot y_{t-1} = 0 \,\forall\, k, t$ (see Wagner and Whitin 1958) only those dominant production plans which ensure the capacity constraints. Bitran and Matsuo (1986) derived error bounds for Manne's formulation. The work by Manne was extended by Dzielinski and Gomory (1965), who developed a column selection procedure to handle larger problems. Later, Lasdon and Terjung (1971) developed a CG approach. Algorithms of this type are also presented by Bahl (1983), Cattrysse et al. (1990), Salomon et al. (1993), Degraeve and Jans (2003) and Huisman et al. (2003).

Hindi (1995) presented a heuristic including variable redefinition and CG. He solved a minimum cost flow problem to compute the upper bound. Hindi (1996) combined the ideas of LP relaxation, CG, minimum cost network flow and Tabu search into a hybrid algorithm. Haase (2005) also solved the CLSP via column generation. The resulting lower bound is very tight (like in the SR and SPL model formulations) as most of the variables are already integer. Only a small number of variables are left which are considered in a following rolling time window procedure similar to Dillenberger et al. (1993).

If CG is used to solve the LP-relaxation of the MLCLSP, the number of relaxed binary variables with fractional (non-integer) values is often much higher than for single-level CLSPs. For this reason, effective CG-based heuristics for the MLCLSP have not yet been presented.

3.2 Lagrangian heuristics

Lagrangian heuristics (LH) are iterative solution approaches applying Lagrangian relaxation (LR). In LR, the complicating constraints of a problem are relaxed and their violation is punished at penalty cost in the objective function. At each iteration, a lower bound is computed based on a Lagrangian relaxation or decomposition and given values of the Lagrangian multipliers. A feasible solution is constructed and serves as the new upper bound. Finally, the Lagrangian multipliers are updated.

Lagrangian heuristics are based on the convergence of the lower and the upper bound through the adaptation of the Lagrangian multipliers. The basic idea of LH is that with the adequate Lagrangian multipliers the solution to the relaxed problem will be very close to the optimal solution of the original problem and only small modifications will have to be made to obtain a close-to-optimal feasible solution.

To compute the lower bounds, the complicating constraints are relaxed. In the case of the CLSP, the only complicating constraints are the capacity constraints. When they are relaxed, the remaining problem decomposes into K problems of the SLULSP type. The inventory balance equations of the MLCLSP represent a second type of interdependencies between products, which results from the input–output relationships between component items and their successors in the bill-of-material structure.

Relaxing both constraints again leads to K problems of the SLULSP type. An alternative consists of relaxing only the capacity constraints and trying to solve the remaining MLULSP. For the CLSPL, the capacity constraints and the setup carryover constraints are relaxed, i.e. the restriction to at most one setup carryover per resource and period. The remaining problem is of the SLULSPL type. A number of exact and heuristic algorithms have been introduced to solve the uncapacitated lot-sizing problems efficiently, e.g. Wagner and Whitin (1958), Evans (1985), Federgruen and Tzur (1991), Wagelmans et al. (1992), Silver and Meal (1969, 1973) and Groff (1979).

Lagrangian multipliers are updated via subgradient optimization, until a stopping criterion is met. The subgradient indicates the direction, in which the Lagrangian multipliers have to be altered, to achieve the greatest possible improvement of the objective value. Generally, in the case of optimization problems with multiple constraints, the subgradients can be based on the violation of the constraints themselves. Table 3 gives an overview.

Lagrangian relaxation Thizy and van Wassenhove (1985) developed a solution algorithm to the CLSP, which was later extended by Diaby et al. (1992b) to a CLSP with setup times and overtime. Based on the LR of the capacity constraints, the setup decisions are determined. Once the setup decision has been made, the resulting problem can be formulated as a transportation problem (TP). A perturbation scheme is developed based on this formulation.

Trigeiro (1987) solved the CLSP without setup times. Later, Trigeiro et al. (1989) extended their heuristic to problems with setup times. To generate feasible solutions, they apply a smoothing procedure which was adapted by other authors. Its objective is to minimize total opportunity cost divided by the quantity of overtime eliminated. The smoothing routine consists of up to two backward and two forward passes, starting either from the end or from the beginning of the planning horizon. In the case of the backward passes, lots are shifted backward either to the immediately preceding period or to the closest preceding period with a corresponding setup. Either the complete lot or as much as necessary to eliminate a capacity violation is shifted. For the forward passes, the target period is always the next period and the quantity shifted is always the inventory of the respective item. At most one lot per period may be split. The heuristic moves on to the next period when all overtime has been eliminated in

Table 3 Lagrangian heuristics

Reference	DC	RC	SP	SOL	FEAS	MU	MOD
Brahimi et al. (2006a)	LR	C, TW	(TW) SLU	DP	FC, B, F	SO	SL, TW
Campbell and Mabert (1991)	LR	C, II	CYCSLULSP	ENUM	F, B	SO	SL, CS, II, MINQ
Chen and Chu (2003)	LR	BIN	LMLCLSP	LP	SET1	SSG	ML
Diaby et al. (1992b)	LR	C	SLU	DP	TP	SO	SL, ST, OV
Hindi et al. (2003)	LR	C	SLU	DP	FC, B, F	SO	SL, ST
Millar and Yang (1994)	LR	SET	TP, INT	LCINT	TP	SO	SL, BO
Moorkanat (2000)	LR	C, I	SLU	DP	F, B, M	SO	ML, ST
Özdamar and Barbarosoglu (1999)	LR	PM	SLU, LOAD	GR	SA	SSG	ML, ST, PM, OV, BO
Özdamar and Barbarosoglu (2000)	LR	C, (I)	M(S)LULSP	DP	SA, LS, GS	SSG	ML, ST
Sambasivan and Yahya (2005)	LR	C	MPSLULSP	B&B	F, B, M	SO	SL, ST, MP
Sox and Gao (1999)	LR	C, S	SLULSPL	DP	FC, F, B, SC	SO	SL, SC
Tempelmeier and Derstroff (1993)	LR	C, I	SLU	DP	FC, B, F	SO	ML, ST
Tempelmeier and Derstroff (1996)	LR	C, I	SLU	DP	FC, B, F	SO	ML, ST
Thizy and van Wassenhove (1985)	LR	C	SLU	DP	TP	SO	SL
Trigeiro (1987)	LR	C	SLU	DP	FC, B, F	SO	SL
Trigeiro et al. (1989)	LR	C	SLU	DP	FC, B, F	SO	SL, ST
Millar and Yang (1994)	LD	DEC	TP, SLU	Zangwill (1966)	TP	SO	SL, BO

Abbreviations

DC Decomposition approach (LD = Lagrangian decomposition, LR = Lagrangian relaxation)

RC Relaxed constraints (BIN = binary setup variables, C = capacity constraints, DEC = coupling constraint between duplicates and the corresponding original variables in decomposition approaches, I = inventory balance constraints, II = initial inventory constraints, PM = coupling between production on each resource and total production for an item, S = setup carryover constraints, SET = coupling constraint between production and setup, TW = time window constraints)

SP Subproblem solved in the case of decomposition (CYCSLULSP = SLULSP with cyclical schedules, INT = integer problem to determine setup scheme, LMLCLSP = linear MLCLSP, LOAD = capacitated loading problem, MPSLULSP = multi-plant SLULSP, SLU = SLULSP, TP = transportation problem, TWSLU = SLULSP with time windows)

SOL Solution method applied to solve the subproblem (B&B = branch and bound, DP = dynamic programing, ENUM = enumeration of all possible schedules, GR = greedy, LCINT = least cost search to determine setups, LP = linear programing)

FEAS Type of feasibility check (B = backward, BC = backward shifting of cumulative overtime, F = forward, FC = forward shifting of cumulative overtime, GS = global search, LS = local search, M = machine-by-machine, SA = simulated annealing, SC = feasibility of setup carryover constraints, SET1 = all positive setup variables are set to 1, TP = fix setups and solve transportation problem)

MU Type of multiplier update (SO = Subgradient Optimization, SSG = Surrogate subgradient method)

MOD Problem solved (ML = multi-level, SL = single-level, BO = backorders, CS = cyclical schedules, II = initial inventory, MINQ = minimum production quantity, MP = multi-plant, OV = overtime, PM = parallel machines, SC = setup carryover, SD = sequence dependency, ST = setup time, TW = time windows)

the incumbent period. It is guided by the constraints, the cost and the Lagrangian multipliers. Among the forward passes, the first considers cumulative overtime only, while the second eliminates overtime in every period. When a feasible solution is found, unnecessary inventory for the given setup pattern is eliminated in a final step by postponing production to the extent possible (that is limited by demand and capacity constraints).

Hindi et al. (2003) solved the CLSP with setup times. Their heuristic consists of three parts. First, a smoothing heuristic is applied, which is similar to that of

Trigeiro et al. (1989). Second, after each pass, the solution is optimized further by solving a capacitated transshipment problem by the dual network complexity method for the given setup schedule. The transshipment model equals the transportation model of Diaby et al. (1992b), but it is claimed to be much smaller. The smoothing heuristic is followed by a variable neighborhood search based on simple cost comparisons.

Tempelmeier and Derstroff (1993, 1996) developed a solution procedure to the MLCLSP with setup times and positive lead times based on the formulation of Billington et al. (1986). To account for the derived demands in the bill-of-material structure, a sequence of SLULSPs is solved according to a low-level code sorting of the items. Then they apply a smoothing procedure similar to Trigeiro et al. (1989) to ensure capacity feasibility. However, as they solve a multi-level problem, they applied both single- and multi-item shifts. A multi-item shift affects a subset of the product structure simultaneously. Their procedure is extremely fast, but relatively complex and difficult to implement.

Moorkanat (2000) solved the MLCLSP with setup times using the single- and multi-item shifts introduced by Tempelmeier and Derstroff (1996). The major difference to the implementation by Tempelmeier and Derstroff (1996) is that back-logging is allowed for the end items. Hence, while the schedule may not be feasible with respect to the inventory balance constraints, it is guaranteed to be feasible with respect to the capacity constraints.

Campbell and Mabert (1991) developed an LH similar to Trigeiro et al. (1989), but solved the CLSP with cyclic schedules in which the times between production periods of an item are constant. To solve the relaxed problems, all combinations of a first production period and cycle length are evaluated and the least cost solution is chosen. Sox and Gao (1999) developed an LR heuristic (which they refer to as Lagrangian decomposition) with relaxation of the capacity and setup carryover constraints.

Özdamar and Barbarosoglu (1999) developed a hybrid algorithm, combining LR and simulated annealing to solve a multi-level production system with serial product structures for several end items. Production takes place on parallel machines. They relax the coupling constraint between production quantities on each machine and the respective total production quantity. The problem therefore decomposes into a set of SLULSPs and a set of capacitated loading problems for given total production quantities per item and period. Both subproblems are solved only approximately. From a generated inventory feasible solution, capacity feasibility is finally achieved through a specialized procedure based on simulated annealing.

Özdamar and Barbarosoglu (2000) applied a similar procedure to solve the standard MLCLSP with general product structures. They developed two relaxation schemes. The first is called hierarchical relaxation, as only the capacity constraints are relaxed. The second relaxes both the capacity and the inventory balance constraints. Neither relaxation is solved to optimality and therefore the solutions do not represent lower bounds to the original problem. First, they generate a feasible solution by an iterative procedure shifting production to eliminate capacity violations. Then, a simulated annealing approach is applied. It searches the direct neighborhood, defined by shifting a randomly selected partial lot to the next resource or the next period. Finally, another procedure is based on the same neighborhood definition, but allows cost reductions only.

Chen and Chu (2003) developed a solution algorithm to the MLCLSP, in which the binary constraints for the setup variables are relaxed. The remaining linear model is solved by an iterative linear programing algorithm, which does not lead to the exact solution. Feasibility is achieved trivially by rounding up all non-zero relaxed setup variables. To update the Lagrangian multipliers, they used a surrogate subgradient method, which is a subgradient method adapted to the fact that the relaxed problem is only solved approximately. At each iteration, the setup states are computed for a given set of production quantities and vice versa. The procedure terminates when both computations lead to the same solution. Finally, the solution is improved through a local search procedure.

Sambasivan and Yahya (2005) applied LR to a multi-plant version of the CLSP with setup times. This extension is reflected through several resources, item transfer in the inventory balance constraints and the corresponding cost factor in the objective function. They also altered the problem to forbid inter-plant transfers. The resulting problem corresponds to the CLSP with parallel resources. A lot shifting–splitting–merging routine is used to generate a feasible solution. Initially, capacity violating lots are shifted within the same period to another plant where the corresponding demand arises. If capacity violations persist, production is shifted across periods within the same plant. The shifts are executed forward and backward without incurring capacity violations in the target periods.

Brahimi et al. (2006a) proposed several LR schemes for the CLSP with time windows. In the standard CLSP, production can take place not later than the "due" demand period. Here, production is additionally limited to be not earlier than a given "release" period.

Lagrangian decomposition Lagrangian decomposition (LD) via variable duplication is a variant of LR. Instead of relaxing the complicating constraints, all original constraints are kept. However, the original problem is decomposed into subproblems via variable duplication, where each subproblem only contains some of the constraints. For the solutions to the subproblems to be a valid solution to the original problem, the duplicates must equal the corresponding original variables. These coupling constraints are finally relaxed.

Millar and Yang (1994) developed both an LR and an LD approach to solve the CLSP with backorders. Both decompose the original problem into two subproblems. In both cases, one of them is a TP. In the case of the LR, the other is an integer problem to determine setups which is solved by inspection of the given cost coefficients. In the case of the LD, the second subproblem is a set of SLULSPs, which is solved by the algorithm introduced by Zangwill (1966). To compute the upper bound, again a TP is solved for the setup decisions given by the relaxation.

3.3 Decomposition and aggregation approaches

The idea of decomposition and aggregation approaches is to solve subproblems of reduced size and then coordinate the individual solutions. Aggregation approaches

Table 4 Decomposition and aggregation heuristics

Reference	AG/DC	SP	SOL	MOD
Blackburn and Millen (1984)	DC (K)	SLU	MRP	ML
Helber (1994)	DC (K)	SL	DS	ML, ST
Helber (1995)	DC (K)	SL	DS	ML, ST
Kirca and Kökten (1994)	DC (K)	SL	DP	SL
Newson (1975a)	DC (K)	SLU	DP	SL, ST
Newson (1975b)	DC (K)	SLU	DP	SL, ST, OV
Sambasivan and Schmidt (2002)	DC (K)	SLU	EX	SL, ST, MP
Tempelmeier and Helber (1994)	DC (K)	SL	DS	ML
Bourjolly et al. (2001)	DC (T)	SL	TS	SL, ST, SC
Boctor and Poulin (2005)	AG (K)	SL	GR	ML, ST (OV), MM
Özdamar and Bozyel (2000)	AG (K)	SI	EX	SL, ST, OV

Abbreviations
AG/DC Aggregation or decomposition approach (AG = aggregation, DC = decomposition, K = item-based, T = period-based)
SP Subproblem that is solved in the case of decomposition (SI = single-item CLSP, SL = CLSP, SLU = SLULSP)
SOL Solution method applied to solve the (sub)problem (DP = dynamic programing, DS = modified Dixon–Silver heuristic, EX = exact solution, GR = greedy, MRP = capacitated MRP, TS = Tabu Search)
MOD Problem solved (ML = multi-level, SL = single-level, MM = multi-machine, MP = multi-plant, OV = overtime, SC = setup carryover, ST = setup time)

reduce the problem size by omitting details first and breaking the solution down later. Decomposition approaches split the original problem into subproblems and coordinate the schedules later. In item-based decomposition capacity restrictions may be neglected for the subproblems. Time-based decomposition is mostly combined with rolling schedules. The idea is to split the planning horizon into shorter (usually overlapping) time windows. Once a solution is found for the current window, the solution is fixed for some periods at the beginning of the window and the problem solved again for the next time window. The F&R algorithms presented above are closely related. Table 4 gives an overview of decomposition and aggregation approaches.

Item-based decomposition approaches Newson (1975a,b) developed a heuristic to the CLSP without and with overtime, respectively. The idea is to neglect the capacity constraint first, which decomposes the CLSP into single-item problems of the SLULSP type. The latter are solved with the help of the Wagner–Whitin algorithm. Then infeasible combinations of production schedules are eliminated from the set of possible solutions. The reduced set is resolved until a feasible solution is achieved.

Kirca and Kökten (1994) developed an item-by-item heuristic to the CLSP. In an iterative algorithm, one of the yet unscheduled items is selected with respect to a cost criteria. In the next step, production quantities are bounded, with the intention to ensure feasible plans for the remaining items. Finally, for the selected item the modified single-item CLSP with inventory restrictions is solved via an algorithm based on dynamic programing developed by Kirca (1990).

Sambasivan and Schmidt (2002) solved a multi-plant CLSP with inter-plant transfers by first solving SLULSPs and then applying a smoothing procedure which shifts production quantities across plants and periods to remove capacity violations. The uncapacitated subproblems are reformulated as shortest-route problems and solved exactly.

One decomposition principle for the MLCLSP is based on the assumption that the only link between production levels is via derived demand. The implication is that lot-sizing decisions for a given item may have an impact on the lot-sizing decisions for all predecessor items and hence the arising cost. Thus, one way of reflecting these interdependencies is via cost modification. Blackburn and Millen (1984) suggested a set of algorithms for the MLULSP for assembly type bill-of-material structures. Firstly, holding and setup cost are modified based on the assumption that the order interval of a component is always an integer multiple of its successor. Secondly, based on a low-level-code ordering of the items, independent SLULSPs are solved.

Tempelmeier and Helber (1994) solved the MLCLSP by decomposing the MLCLSP into a sequences of CLSPs. They proposed four versions, which arise by combining two ways to construct the CLSPs and two types of cost adjustments. The CLSPs are solved with a modified version of the Dixon–Silver heuristic which includes a multi-level feasibility check. Helber (1994, 1995) extended the algorithm to cover problems with setup times.

Period-based decomposition approaches Bourjolly et al. (2001) extended the method of Gopalakrishnan et al. (2001) for the solution of the CLSPL by applying it in a rolling horizon fashion. Based on the idea of dynamic recursion, first the problem for period 1, then for periods 1 and 2, etc., are solved. The best schedule obtained for a given subproblem is used to find the best schedule for the next subproblem. Setup carryovers are introduced later by linking lots for an item, if it is produced in two consecutive periods.

Aggregation approaches Özdamar and Bozyel (2000) proposed a planning approach, which is based on the aggregation of the demand of all items in a period. Setups are not accounted for explicitly but via a setup allowance percentage, which reduces capacity. Finally, individual lots that respect the aggregated lot-sizes are determined via a filling procedure. As a good estimation of the setup allowance is not trivial, the authors also proposed an iterative approach, in which the estimate is updated based on setup time required in the previous solution.

Boctor and Poulin (2005) solved an MLCLSP for a serial bill-of-material structure with item-specific resources. With the underlying assumption that lot-sizes are the same on each stage, they treated the multi-level problem by constructing a surrogate single-level problem which is then solved by a greedy heuristic.

3.4 Metaheuristics

Metaheuristics are general strategies guiding the process to solve optimization problems. They may make use of domain-specific knowledge in the form of problem-specific heuristics that are controlled by the upper level strategy. The hope behind this approach is to gain flexibility and the ability to handle large and complex problems. Metaheuristics are usually non-deterministic and may incorporate mechanisms to avoid getting trapped in confined areas of the search space. Furthermore, the search space may also include infeasible solutions, where the violation of constraints is charged with penalty cost.

Metaheuristics belong to the group of improvement procedures starting from a given initial solution. The two basic principles that largely determine the behavior of a metaheuristic are intensification and diversification. The latter enhances the exploration of the search space, while the former allows for the exploitation of the accumulated search experience.

Most of the metaheuristics for lot-sizing problems use a direct solution representation, e.g. binary variables for setups, as well as continuous variables for production decisions. Furthermore, some heuristics are restricted to finding a close-to-optimal binary grid. The respective production plan can then be derived via linear programing, heuristic approaches or dual reoptimization, as described earlier. Alternatively, in genetic algorithms, an indirect encoding can be used, e.g. by Kohlmorgen et al. (1999). They used a priority rule for selecting items to build a feasible production plan.

Metaheuristic algorithms range from simple local search procedures to complex learning processes. Presently, more advanced metaheuristics use search experience to guide the search. Examples include simulated annealing (SA), Tabu search (TS), variable neighborhood search (VNS), genetic algorithms (GA) and ant colony optimization (ACO). Table 5 gives an overview on papers dealing with metaheuristics for capacitated lot-sizing problems.

Local search It is arguable whether local search (LS) itself is a metaheuristic. However, it is described here as all procedures based on neighborhood search (e.g. simulated annealing) follow the principles of LS. Also called iterative improvement or hill-climbing methods, LS methods are based on the idea that at least a local optimum can be found by starting from a given solution and iteratively trying to find a better one in an appropriately defined neighborhood. Here, the neighborhood is defined as the set of solutions which can be obtained from the current one by performing simple modifications, called moves. In the field of lot-sizing, possible moves include lotshifting, lotsplitting, as well as elimination and creation of setups. LS procedures also vary according to whether they explore the complete or only a part of the neighborhood of a solution and the stopping criterion applied.

Haase (1994) applied a stochastic backward-oriented scheduling procedure based on a randomized regret measure to solve the CLSPL. Haase (1998) embedded the same method in an LS procedure where the regret measure is determined iteratively resulting in the best solution value.

Chen and Chu (2003) combined an LR approach with LS. In every iteration, the feasible solution constructed during the LR phase is further improved by changing the values of two setup variables simultaneously during a subsequent local search.

As mentioned earlier, simulated annealing (SA), Tabu search (TS) or variable neighborhood search (VNS) can be seen as extensions to LS as they tried to escape from

Table 5 Metaheuristics

Reference	NHS	PB	MOD
Chen and Chu (2003)	LS		ML
Haase (1994)	LS		SL, SC
Haase (1998)	LS		SL, SC
Hindi et al. (2003)	VNS		SL, ST
Barbarosoglu and Özdamar (2000)	SA		ML, ST
Berretta et al. (2005)	SA, TS		ML, ST, LT
Helber (1994)	SA, TS	GA	ML, ST
Helber (1995)	SA, TS	GA	ML, ST
Hung and Chien (2000)	SA, TS	GA	ML, ST, BO
Özdamar and Barbarosoglu (1999)	SA	GA	ML, ST, PM, OV, BO
Özdamar and Barbarosoglu (2000)	SA		ML, ST
Özdamar and Bozyel (2000)	SA		ML, ST, OV
Özdamar et al. (2002)	SA, TS	GA	SL, ST, OV
Kuik et al. (1993)	SA, TS		ML
Salomon (1991)	SA, TS		ML
Salomon et al. (1993)	SA, TS		ML
Gopalakrishnan et al. (2001)	TS		SL, SC
Hindi (1996)	TS		SL
Hung et al. (2003)	TS		SL, ST, BO
Gutierrez et al. (2001)		GA	ML
Haase and Kohlmorgen (1995)		GA	SL
Hung et al. (1999)		GA	SL, ST, BO
Kohlmorgen et al. (1999)		GA	SL
Xie and Dong (2002)		GA	ML, ST, OV
Berretta and Rodrigues (2004)		MA	ML, ST
Pitakaso et al. (2006)		ACO	ML, ST

Abbreviations
NHS Neighborhood search (LS = local search, SA = simulated annealing, TS = Tabu search, VNS = variable neighborhood search)
PB Population-based heuristics (ACO = ant colony optimization, GA = genetic algorithms, MA = memetic algorithms)
MOD Problem solved (ML = multi-level, SL = single-level, BO = backorders, OV = overtime, LT = lead times, PM = parallel machines, SC = setup carryover, ST = setup time)

local optima to find a better solution in yet unexplored parts of the solution space. However, a sensible choice of the stopping criterion is required, when the algorithm is designed to overcome local optima.

Variable neighborhood search VNS was proposed in Hansen and Mlaydenović (1999, 2001). The basic idea is that the process consists of three phases of shaking, local search and move. During the shaking phase, a neighboring solution is randomly selected from each current solution. The size of the neighborhood increases if no improvement is made. The neighboring solution then becomes the starting point of an LS method. Finally, the resulting solution is compared to the current. If it is beneficial, the move is executed.

In their solution approach, Hindi et al. (2003) combined LR and VNS. Starting with the solution obtained by the LR, a smoothing heuristic similar to that presented by Trigeiro et al. (1989) is used to eliminate capacity infeasibility and to improve the current best solution during the VNS.

Simulated annealing Simulated annealing was proposed for combinatorial optimization by Kirkpatrick et al. (1983) and can be seen as a combination of iterative improvement and random walk. Neighboring solutions are randomly selected and the search is guided by a decreasing probability to accept such a neighboring solution even though it is worse than the current solution.

Kuik et al. (1993) combined their RH as described in Sect. 3.1 with an SA approach for the MLCLSP without setup times. Helber (1995) investigated different cooling schemes for SA approaches to determine a good setup pattern for the MLCLSP (see also Helber 1994). Furthermore, he examined the impact of different initial solutions on the overall solution quality. Further SA algorithms have been presented by Salomon (1991), Salomon et al. (1993), Hung and Chien (2000), Barbarosoglu and Özdamar (2000), Özdamar and Barbarosoglu (2000), Özdamar et al. (2002), Özdamar and Bozyel (2000) and Berretta et al. (2005).

Tabu search Tabu search was first introduced by Glover (1986). It can be either a deterministic or a stochastic procedure. The distinct characteristic giving the name to Tabu search is that it uses so-called Tabu lists. Tabu lists store information on recent moves to prevent their reversal. The Tabu list length is decisive for the size of the section of the solution space that is searched. More advanced algorithms apply adaptive Tabu lists, which depend on the quality of the recently visited solutions (improvement or deterioration). To prevent the exclusion of beneficial moves, so-called aspiration criteria can be defined (e.g. if the corresponding solution is the best found so far), which allow Tabu moves to be executed.

Salomon et al. (1993) presented a TS and SA approach to solve the subproblems of their CG heuristic for the CLSP without setup times. Hung et al. (2003) solved the CLSP with backlogging with an improved TS method. Neighboring solutions are generated by minor changes to the current setup pattern. The remaining LP is solved to optimality and post-optimization information is used to conduct the next move. Gopalakrishnan et al. (2001) developed a TS heuristic for the CLSP with setup carryover using different move types for the sequencing and lot-sizing decisions. TS was also applied by Salomon (1991), Kuik et al. (1993), Helber (1994, 1995), Hindi (1996), Hung and Chien (2000), Özdamar et al. (2002), and Berretta et al. (2005).

Genetic algorithms Genetic algorithms belong to the category of evolutionary algorithms (EA). They are population-based and incorporate a learning component via the recombination of solutions.

Genetic algorithms are inspired by the principles of natural selection. They operate simultaneously on a population of individuals encoded as "chromosomes" by creating new generations of offsprings through an iterative process until some convergence criterion is met. The best chromosome generated is then decoded, providing the corresponding solution. The underlying reproduction process is mainly aimed at improving the fitness of individuals, i.e. the cost to be minimized while exploring the solution space. The algorithm applies stochastic operators such as selection, crossover and mutation on an initially random population, to compute a new generation of individuals. A major challenge is to represent solutions in such a way that the offspring of two different feasible solutions is again feasible.

Özdamar and Barbarosoglu (1999) introduced a hybrid algorithm, which incorporates SA and GA into an LR scheme. Haase and Kohlmorgen (1995) and Kohlmorgen et al. (1999) implemented a parallel GA for the CLSP using an indirect encoding. Here, a solution consists of a string of real values, which is decoded by a heuristic to create a production plan. Further genetic algorithms have been proposed for lot-sizing by Helber (1994, 1995), Hung et al. (1999), Hung and Chien (2000), Özdamar and Bozyel (2000), Gutierrez et al. (2001), Özdamar et al. (2002) and Xie and Dong (2002).

Memetic algorithms Other evolutionary algorithms that are similar to GA are memetic algorithms (MA). Basically they combine LS heuristics with crossover operators of genetic algorithms. Although population-based, memetic algorithms are not restrained by a biological metaphor. The aim is to exploit during the LS all the available knowledge about the problem. This includes optimal solution properties, heuristics and truncated exact algorithms. Berretta and Rodrigues (2004) proposed a memetic algorithm for the MLCLSP based on the work of França et al. (1997).

Ant colony optimization Ant colony optimization techniques (ACO) are inspired by the real world's behavior of ant colonies. During their search for food, ants lay down pheromone trails. Following ants usually choose the trail with the highest concentration of pheromone. The reason is that satisfactory paths possess a higher concentration of pheromone than unsatisfactory ones.

Pitakaso et al. (2006) combined an ant colony optimization algorithm with an F&R heuristic for the MLCLSP. During the F&R heuristic only a subset of the constraints and variables are considered relating to a reduced number of products and periods. The corresponding MIP is solved to optimality and the binary variables are fixed. The ACO uses virtual pheromone concentrations that have been built up during previous iterations to single out promising subsets for the next iteration. Subsequently, the pheromone information is updated.

3.5 Problem-specific greedy heuristics

Finally, a set of rather intuitive heuristic algorithms are the greedy heuristics (GH). Starting from scratch and working period-by-period or starting from a given initial solution, they usually increase lot-sizes successively to achieve cost savings. GH usually consist of a feasibility routine and a priority index to select the best candidate for such a move. While the former ensures feasibility of the overall schedule,

Table 6 Problem-specific greedy heuristics

Reference	TGR	DIR	FEAS	CC	MOD
Dixon and Silver (1981)	C	F	LA	SM	SL
Dogramaci et al. (1981)	C	F	LA	SM	SL
Eisenhut (1975)	C	F	FB	PPB	SL
Gupta and Magnusson (2005)	C	F	FB	CD	SL, ST, SC
Lambrecht and Vanderveken (1979)	C	F	FB	SM	SL
Maes and van Wassenhove (1986c)	C	F	LA	SM, LUC, LTC, PPB	SL
Boctor and Poulin (2005)	I (L4L)	F	FB	PPB, MC, AC	SL, ST, OV, MM
Clark and Armentano (1995)	I (WW)	F, B	SP	CPO	ML, ST
Dogramaci et al. (1981)	I (L4L)	G	SP	LTC	SL
França et al. (1997)	I (WW)	F, B	SP	CPO	ML, ST
Günther (1987)	I (L4L)	F	LA	CPP (G)	SL
Karni and Roll (1982)	I (WW)	G	SP	LTC	SL
van Nunen and Wessels (1978)	I (WW)	B	1P	CPP	SL, ST, OV
Selen and Heuts (1989)	I (L4L)	F	LA	CPP (G)	SL
Trigeiro (1989)	I (L4L)	F, B	SP	CPP (mod SM)	SL, ST, OV

Abbreviations
TGR Type of greedy algorithm (C = constructive, I = improving, L4L = lot-for-lot, WW = Wagner–Whitin)
DIR Direction of greedy algorithm (B = backward, F = forward, G = considering the complete schedule at each step)
FEAS Type of feasibility check (1P = lots are respectively shifted back one period—slack capacity is shifted forward, FB = feedback, LA = look-ahead, SP = smoothing procedure)
CC Cost criterion for choice of item (AC = exact average cost savings, CD = cumulative demand, CPO = cost per overtime reduction, CPP = cost per setup and production time, G = Groff, LTC = least total cost, LUC = least unit cost, MC = marginal cost, PPB = part period balancing, SM = silver-meal)
MOD Problem solved (ML = multi-level, SL = single-level, MM = multi-machine, OV = overtime, SC = setup carryover, ST = setup time)

i.e. that all demand is served without backlogging and capacity violation, the latter serves as a cost criterion to manipulate scheduling decisions.

Two types of feasibility routines can be distinguished: feedback mechanisms and look-ahead mechanisms. The former are usually used in backward routines and push infeasible production quantities to earlier periods. Look-ahead mechanisms, normally used in forward routines, compute a priori the minimum inventory needed to avoid future capacity violations and schedule production lots accordingly. The priority index is often some criterion taken from uncapacitated dynamic lot-sizing heuristics, such as Silver Meal (SM), see Silver and Meal (1969, 1973), least unit cost (LUC), Groff (GR), see Groff (1979), and part period balancing (PPB), see DeMatteis (1968). Table 6 gives an overview.

Constructive heuristics One set of Greedy Heuristics—also referred to as single-resource heuristics, common-sense or specialized heuristics—is constructive, i.e. they generate a solution from scratch by adding components to an initially empty partial solution, until the solution is complete. Most work period-by-period either forward or backward. They are myopic, so that at each step the current lot-size is increased ignoring future cost. This myopic approach can have the beneficial effect of stability if only the decisions for the first periods are implemented in a planning system with rolling horizons. The first heuristic of this type was introduced by Eisenhut (1975). However, the use of a feasibility routine is only suggested, rather than implemented here. Further heuristics of this type include Lambrecht and Vanderveken (1979), Dixon and Silver (1981) and Dogramaci et al. (1981).

Maes and van Wassenhove (1986c) suggested a so-called ABC heuristic. It consists of 72 alternative algorithms based on the combination of (a) six a priori orderings of items, (b) four cost criteria and (c) three search strategies. They suggested applying various combinations and then choosing the best solution.

Gupta and Magnusson (2005) solved the CLSPL with setup times. First, they generated a feasible solution to the CLSP without setup times with the help of a greedy algorithm. Then setup times are introduced under the assumption that a setup is carried over for each period and a smoothing procedure is applied. Finally, a backward-oriented improvement procedure is applied.

Improvement heuristics The other set of Greedy heuristics consists of improvement heuristics that generate a better feasible solution from a usually infeasible inferior starting solution by simple shifting routines. Heuristics of this type have been presented amongst others by van Nunen and Wessels (1978), Günther (1987), Selen and Heuts (1989) and Trigeiro (1989). Selen and Heuts (1989) modified the priority index used by Günther (1987) to account for setup time savings in case a complete lot is shifted.

Boctor and Poulin (2005) solved a CLSP with multiple capacity restrictions. It arises when a serial multi-stage system is aggregated to a single-stage system by the assumption that lot-sizes are the same on each stage. They improved on a lot-for-lot schedule. Initially, a feasible schedule is aimed to be achieved in a forward pass. Through all stages, residual capacity after meeting demand is calculated. If it is positive, the lot is further increased, if negative, production is shifted into earlier periods. Finally, the solution is improved by lot merging (backward shifting of complete lots).

Dogramaci et al. (1981) developed a “four-step algorithm”, which improves a lot-for-lot schedule by shifting complete or partial lots to reduce setup cost, eliminating capacity violations, and finally reducing holding cost. Karni and Roll (1982) initially solved the SLULSP optimally for each item. Then they applied an algorithm similar to the one presented by Dogramaci et al. (1981).

Clark and Armentano (1995) solved the MLCLSP. While the starting routine computes a solution to the MLULSP by solving a series of SLULSPs along the product structure, the smoothing procedure shifts production backward and forward to obtain a feasible schedule. The priority rule is based on the cost changes incurred by the shift (including penalty cost for the overall capacity violations) over the incurred reduction in capacity violation. Penalty cost for capacity violation are gradually increased at each shifting cycle. França et al. (1997) extended the algorithm by adding improvement and merging. The smoothing procedure again shifts production quantities forward and backward, but only if this decreases cost without violating the capacity restriction.

A merging procedure shifts production quantities to periods with an existing setup regardless of the capacity violation. Smoothing, improvement and merging are repeated until a given number of iterations is reached.

Maes and van Wassenhove (1986a,b) compared the heuristics of Lambrecht and Vanderveken (1979), Dixon and Silver (1981) and Dogramaci et al. (1981) when applied on static and rolling horizon problems.

4 Conclusions

In this paper, we presented a structured survey of the broad literature on dynamic capacitated lot-sizing. We focused on models and algorithms for problems of the CLSP type, thus ignoring the literature in sequencing and detailed scheduling.

Five major streams of research were identified with respect to the general algorithmic approach. More than 100 papers from four decades were classified. In many cases, our classification is arguable as authors often creatively combine ideas from different streams of research. A substantial part of the discussed research was published during the past 10 years. This shows how vivid this field of research is. Two streams that appear to be particularly active are those that are based on mathematical programing and those that work with metaheuristics. These two approaches offer the flexibility to treat a broad variety of problems that arise in practice. They require and use the increased computing power that is nowadays available. Most papers present numerical results to address the question how accurate and efficient the proposed method is. However, it is extremely difficult to assess the relative performance of these methods as no agreed-upon standard test bed has been established and accepted in the field and it is doubtful whether such a data set can be established which covers all the different lot-sizing problems arising in reality.

It appears to be impossible to make a precise statement about problem dimensions that can be solved nowadays. It is our impression that the single-level CLSP without setup-times is particularly well-solved by column-generation-based approaches for any relevant problem size, see Haase (2005). Setup times, a setup carryover or multi-level product structures can make the problem much harder to solve, in particular, when capacity limits are tight and/or setups are costly. In these cases, problem instances with $10^3 - 10^4$ binary setup variables can still be solved heuristically within seconds or a few minutes. However, it can be difficult to assess the quality of these solutions as the optimal objective function values are usually unknown.

Although some so-called advanced planning systems (APS) provide the possibility to address lot-sizing issues, the solution methods available in a typical APS are often too generic and not always applicable to realistic problem sizes. Thus, dynamic lot-sizing still appears to be an open question in the field of advanced planning. Nevertheless, the authors' experience shows that an increasing number of companies is aware of the improvement potential inherent in lot-sizing and that planners are open to the application of model-based solution approaches. Methods to solve the MLCLSP or variants thereof are required as layout-type specific components in a hierarchically structured advanced planning system. The objective of the current review was to provide an overview over the methods that are currently available.

References

Akartunalı K, Miller AJ (2008) A heuristic approach for big bucket multi-level production planning problems. Eur J Oper Res. doi:10.1016/j.ejor.2007.11.033

Alfieri A, Brandimarte P, D'Orazio S (2002) LP-based heuristics for the capacitated lot-sizing problem: the interaction of model formulation and solution algorithm. Int J Prod Res 40:441–458

Armentano VA, França PM, de Toledo FMB (1999) A network flow model for the capacitated lot-sizing problem. Omega 27:275–284

Bahl HC (1983) Column generation based heuristic algorithm for multi-item scheduling. IIE Trans 15:136–141

Bahl HC, Ritzman LP, Gupta JND (1987) Determining lot sizes and resource requirements: a review. Oper Res 35:329–345

Barany I, van Roy TJ, Wolsey LA (1984) Strong formulations for multi-item capacitated lot sizing. Manage Sci 30:1255–1261

Barbarosoglu G, Özdamar L (2000) Analysis of solution space-dependent performance of simulated annealing: the case of the multi-level capacitated lot sizing problem. Comput Oper Res 27:895–903

Belvaux G, Wolsey LA (2000) bc-prod: a specialized branch-and-cut system for lot-sizing problems. Manage Sci 46:724–738

Belvaux G, Wolsey LA (2001) Modelling practical lot-sizing problems as mixed-integer programs. Manage Sci 47:993–1007

Berretta R, Rodrigues LF (2004) A memetic algorithm for a multistage capacitated lot-sizing problem. Int J Prod Econ 87:67–81

Berretta R, França PM, Armentano VA (2005) Metaheuristic approaches for the multilevel resource-constrained lot-sizing problem with setup and lead times. Asia Pac J Oper Res 22:261–286

Billington PJ, McClain JO, Thomas LJ (1983) Mathematical programming approaches to capacity-constrained MRP systems: review, formulation and problem reduction. Manage Sci 29:1126–1141

Billington PJ, McClain JO, Thomas LJ (1986) Heuristics for multilevel lot-sizing with a bottleneck. Manage Sci 32:989–1006

Bitran GR, Matsuo H (1986) The multi-item capacitated lot size problem: error bounds of Manne's formulations. Manage Sci 32:350–359

Bitran GR, Yanasse HH (1982) Computational complexity of the capacitated lot size problem. Manage Sci 28:1174–1186

Blackburn JD, Millen RA (1984) Simultaneous lot-sizing and capacity planning in multi-stage assembly processes. Eur J Oper Res 16:84–93

Boctor FF, Poulin P (2005) Heuristics for N-product, M-stage, economic lot sizing and scheduling problem with dynamic demand. Int J Prod Res 43:2809–2828

Bourjolly J-M, Ding K, Gopalakrishnan M, Gramani MCN, Mohan S (2001) On a tactical/operational production planning problem in supply chain management: balancing inventory and setup costs. In: MIC'2001—4th metaheuristics international conference, pp 569–571

Brahimi N, Dauzère-Pérès S, Najid NM (2006a) Capacitated multi-item lot-sizing problems with time windows. Oper Res 54:951–967

Brahimi N, Dauzère-Pérès S, Najid NM, Nordli A (2006b) Single item lot sizing problems. Eur J Oper Res 168:1–16

Campbell GM, Mabert VA (1991) Cyclical schedules for capacitated lot sizing with dynamic demands. Manage Sci 37:409–427

Cattrysse D, Maes J, van Wassenhove LN (1990) Set partitioning and column generation heuristics for capacitated dynamic lotsizing. Eur J Oper Res 46:38–47

Chen H, Chu C (2003) A lagrangian relaxation approach for supply chain planning with order/setup costs and capacity constraints. J Syst Sci Syst Eng 12:98–110

Chen W-H, Thizy J-M (1990) Analysis of relaxations for the multi-item capacitated lot-sizing problem. Ann Oper Res 26:29–72

Clark AR, Armentano VA (1995) A heuristic for a resource-capacitated multi-stage lot-sizing problem with lead times. J Oper Res Soc 46:1208–1222

Dantzig GB, Wolfe P (1960) Decomposition principle for linear programs. Oper Res 8:101–111

Degraeve Z, Jans R (2003) A new Dantzig–Wolfe reformulation and branch-and-price algorithm for the capacitated lot sizing problem with set up times. (April 2003, 03). In: ERIM report series reference no. ERS-2003-010-LIS. Available at SSRN: http://ssrn.com/abstract=411647

DeMatteis JJ (1968) An economic lot-sizing technique i—the part-period algorithm. IBM Syst J 7:30–38
Denizel M, Altekin FT, Süral H, Stadtler H (2008) Equivalence of the LP relaxations of two strong formulations for the capacitated lot-sizing problem with setup times. OR Spectr 30:773–785
Diaby M, Bahl HC, Karwan MH, Zionts S (1992a) Capacitated lot-sizing and scheduling by Lagrangean relaxation. Eur J Oper Res 59:444–458
Diaby M, Bahl HC, Karwan MH, Zionts S (1992b) A lagrangean relaxation approach for very-large-scale capacitated lot-sizing. Manage Sci 38:1329–1340
Dillenberger C, Escudero LF, Wollensak A, Zhang W (1993) On solving a large-scale resource allocation problem in production planning. In: Fandel G, Gulledge T, Jones A (eds) Operations research in production planning and control. Springer, Heidelberg, pp 105–119
Dillenberger C, Escudero LF, Wollensak A, Zhang W (1994) On practical resource allocation for production planning and scheduling with period overlapping setups. Eur J Oper Res 75:275–286
Dixon PS, Silver EA (1981) A heuristic solution procedure for the multi-item, single-level, limited capacity, lot-sizing problem. J Oper Manage 2:23–39
Dogramaci A, Panayiotopoulos JC, Adam NR (1981) The dynamic lot-sizing problem for multiple items under limited capacity. AIIE Trans 13:294–303
Drexl A, Haase K (1995) Proportional lotsizing and scheduling. Int J Prod Econ 40:73–87
Drexl A, Kimms A (1997) Lot sizing and scheduling: survey and extensions. Eur J Oper Res 99:221–235
Dzielinski BP, Gomory RE (1965) Optimal programming of lot sizes, inventory and labor allocations. Manage Sci 11:874–890
Eisenhut PS (1975) A dynamic lot sizing algorithm with capacity constraints. AIIE Trans 7:170–176
Eppen GD, Martin RK (1987) Solving multi-item capacitated lot-sizing problems using variable redefinition. Oper Res 35:832–848
Evans JR (1985) An efficient implementation of the Wagner-Whitin algorithm for dynamic lot-sizing. J Oper Manage 5:229–235
Federgruen A, Tzur M (1991) A simple forward algorithm to solve general dynamic lot sizing models with n periods in $o(n \log n)$ or $o(n)$ time. Manage Sci 37:909–925
Federgruen A, Meissner J, Tzur M (2007) Progressive interval heuristics for multi-item capacitated lot-sizing problems. Oper Res 55:490–502
Fleischmann B (1990) The discrete lot-sizing and scheduling problem. Eur J Oper Res 44:337–348
Fleischmann B, Meyr H (1997) The general lotsizing and scheduling problem. OR Spektrum 19:11–21
Florian M, Lenstra J, Kan AR (1980) Deterministic production planning: algorithms and complexity. Manage Sci 26:669–679
França PM, Armentano VA, Berretta RE, Clark AR (1997) A heuristic method for lot-sizing in multi-stage systems. Comput Oper Res 24:861–874
Garey M, Johnson D (1979) Computers and intractability: a guide to the theory of NP-completeness. Freeman, New York
Gelders LF, Maes J, van Wassenhove LN (1986) A branch and bound algorithm for the multi item single level capacitated dynamic lotsizing problem. In: Axsäter S, Schneeweiss C, Silver E (eds) Multi-stage production planning and inventory control. Springer, Berlin, pp 92–108
Glover F (1986) Future paths for integer programming and links to artificial intelligence. Comput Oper Res 13:533–549
Gopalakrishnan M, Ding K, Bourjolly J-M, Mohan S (2001) A tabu-search heuristic for the capacitated lot-sizing problem with set-up carryover. Manage Sci 47:851–863
Groff G (1979) A lot sizing rule for time-phased component demand. Prod Inventory Manage 20:47–53
Günther H-O (1987) Planning lot sizes and capacity requirements in a single stage production sytem. Eur J Oper Res 31:223–231
Gupta YP, Keung Y (1990) A review of multi-stage lot-sizing models. Int J Oper Prod Manage 10:57–73
Gupta D, Magnusson T (2005) The capacitated lot-sizing and scheduling problem with sequence-dependent setup costs and setup times. Comput Oper Res 32:727–747
Gutierrez E, Hernández W, Süer GA (2001) Genetic algorithms in capacitated lot sizing decisions. In: Computer research conference
Haase K (1994) Lotsizing and scheduling for production planning. Lecture notes in economics and mathematical systems, vol 408. Springer, Berlin
Haase K (1998) Capacitated lot-sizing with linked production quantities of adjacent periods. In: Drexl A, Kimms A (eds) Beyond manufacturing resource planning (MRP II)—advanced models and methods for production planning. Springer, Berlin, pp 127–146

Haase K (2005) Solving large-scale capacitated lot-sizing problems close to optimality. Working paper, Technische Universität Dresden

Haase K, Kohlmorgen U (1995) Parallel genetic algorithm for the capacitated lot-sizing problem. In: Kleinschmidt P, Bachem A, Derigs U, Fischer D, Leopold-Wildburger U, Möhring R (eds) Operations research proceedings. Springer, Berlin, pp 370–375

Hansen P, Mlaydenović N (1999) An introduction to variable neighborhood search. In: Voß S, Martello S, Osman I, Roucairol C (eds) Metaheuristics: advances and trends in local search paradigms for optimization. Kluwer, Dordrecht, pp 433–458

Hansen P, Mlaydenović N (2001) Variable neighborhood search: principles and applications. Eur J Oper Res 130:449–467

Harrison TP, Lewis HS (1996) Lot sizing in serial assembly systems with multiple constrained resources. Manage Sci 42:19–36

Helber S (1994) Kapazitätsorientierte Losgrößenplanung in PPS-Sytemen. M&P

Helber S (1995) Lot sizing in capacitated production planning and control systems. OR Spektrum 17:5–18

Hindi KS (1995) Computationally efficient solution of the multi-item, capacitated lot-sizing problem. Comput Ind Eng 28:709–719

Hindi KS (1996) Solving the CLSP by a tabu search heuristic. J Oper Res Soc 47:151–161

Hindi KS, Fleszar K, Charalambous C (2003) An effective heuristic for the CLSP with set-up times. J Oper Res Soc 54:490–498

Huisman D, Jans R, Peeters M, Wagelmans APM (2003) Combining column generation and Lagrangian relaxation (1 January 2003). In: ERIM report series reference no. ERS-2003-092-LIS. Available at SSRN: http://ssrn.com/abstract=496704

Hung Y-F, Chien KL (2000) A multi-class multi-level capacitated lot sizing model. J Oper Res Soc 51: 1309–1318

Hung Y-F, Hu Y-C (1998) Solving mixed integer programming production planning problems with setups by shadow price information. Comput Oper Res 25:1027–1042

Hung Y-F, Shih C-C, Chen C-P (1999) Evolutionary algorithms for production planning problems with setup decisions. J Oper Res Soc 50:857–866

Hung Y-F, Chen C-P, Shih C-C, Hung M-H (2003) Using tabu search with ranking candidate list to solve production planning problems with setups. Comput Ind Eng 45:615–634

Jacobs F, Khumawala B (1982) Multi-level lot sizing in material requirements planning: an empirical investigation. Comput Oper Res 9:139–144

Jans R, Degraeve Z (2007) Meta-heuristics for dynamic lot sizing: a review and comparison of solution approaches. Eur J Oper Res 177:1855–1875

Karimi B, Ghomi SMTF, Wilson JM (2003) The capacitated lot sizing problem: a review of models and algorithms. Omega 31:365–378

Karmarkar US, Schrage L (1985) The deterministic dynamic product cycling problem. Oper Res 66: 326–345

Karni R, Roll Y (1982) A heuristic algorithm for the multi-item lot sizing problem with capacity constraints. IIE Trans 14:249–256

Katok E, Lewis HS, Harrison TP (1998) Lot sizing in general assembly systems with setup costs, setup times, and multiple constrained resources. Manage Sci 44:859–877

Kirca Ö (1990) An efficient algorithm for the capacitated single dynamic lot size problem. Eur J Oper Res 45:15–24

Kirca Ö, Kökten M (1994) A new heuristic approach for the multi-item dynamic lot sizing problem. Eur J Oper Res 75:332–341

Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. Science 220:671–680

Kohlmorgen U, Schmeck H, Haase K (1999) Experiences with fine-grained parallel genetic algorithms. Ann Oper Res 90:203–219

Krarup J, Bilde O (1977) Plant location, set covering and economic lot size: An $o(mn)$-algorithm for structured problems. In: Collatz L, Meinardus G, Wetterling W (eds) Numerische Methoden bei Optimierungsaufgaben Band 3. Birkhäuser, Basel, pp 155–180

Kuik R, Salomon M (1990) Multi-level lot-sizing problem: evaluation of a simulated-annealing heuristic. Eur J Oper Res 45:25–37

Kuik R, Salomon M, van Wassenhove LN, Maes J (1993) Linear programming, simulated annealing and tabu search heuristics for lotsizing in bottleneck assembly systems. IIE Trans 25:62–72

Kuik R, Salomon M, van Wassenhove LN (1994) Batching decisions: structure and models. Eur J Oper Res 75:243–263

Lambrecht MR, Vanderveken H (1979) Heuristic procedures for the single operation, multi-item loading problem. AIIE Trans 11:319–326
Lasdon LS, Terjung RC (1971) An efficient algorithm for multi-item scheduling. Oper Res 19:946–969
Maes J, van Wassenhove LN (1986a) Multi item single level capacitated dynamic lotsizing heuristics: a computational comparison (part I: static case). IIE Trans 18:114–123
Maes J, van Wassenhove LN (1986b) Multi item single level capacitated dynamic lotsizing heuristics: a computational comparison (part II: rolling horizon). IIE Trans 18:124–129
Maes J, van Wassenhove LN (1986c) A simple heuristic for the multi item single level capacitated lotsizing problem. Oper Res Lett 4:265–273
Maes J, van Wassenhove L (1988) Multi-item single-level capacitated dynamic lot-sizing heuristics: a general review. J Oper Res Soc 39:991–1004
Maes J, McClain JO, van Wassenhove LN (1991) Multilevel capacitated lotsizing complexity and LP-based heuristics. Eur J Oper Res 53:131–148
Manne AS (1958) Programming of economic lot sizes. Manage Sci 4:115–135
Mercé C, Fontan G (2003) MIP-based heuristics for capacitated lotsizing problems. Int J Prod Econ 85: 97–111
Millar HH, Yang M (1994) Lagrangian heuristics for the capacitated multi-item lot-sizing problem with backordering. Int J Prod Econ 34:1–15
Miller AJ, Nemhauser GL, Savelsbergh MWP (2000) Solving multi-item capacitated lot-sizing problems with setup times by branch-and-cut. Université Catholique de Louvain, Belgium. Core discussion paper 00/39
Moorkanat J (2000) Studies in certain resource loading, scheduling and production control problems in multi-stage production control problems in multi-stage production inventory systems. Ph.D. thesis, Indian Institute of Technology, Bombay
Newson EFP (1975a) Multi-item lot size scheduling by heuristic, part I: with fixed resources. Manage Sci 21:1186–1193
Newson EFP (1975b) Multi-item lot size scheduling by heuristic, part II: with variable resources. Manage Sci 21:1194–1203
Özdamar L, Barbarosoglu G (1999) Hybrid heuristics for the multi-stage capacitated lot sizing and loading problem. J Oper Res Soc 50:810–825
Özdamar L, Barbarosoglu G (2000) An integrated Lagrangean relaxation-simulated annealing approach to the multi-level multi-item capacitated lot sizing problem. Int J Prod Econ 68:319–331
Özdamar L, Bozyel MA (2000) The capacitated lot sizing problem with overtime decisions and setup times. IIE Trans 32:1043–1057
Özdamar L, Birbil ŞI, Portmann M-C (2002) Technical note: new results for the capacitated lot sizing problem with overtime decisions and setup times. Prod Plann Control 13:2–10
Pitakaso R, Almeder C, Doerner KF, Hartl RF (2006) Combining exact and population-based methods for the constrained multilevel lot sizing problem. Int J Prod Res 44:4755–4771
Pochet Y, Wolsey LA (1991) Solving multi-item lot-sizing problems using strong cutting planes. Manage Sci 37:53–67
Quadt D, Kuhn H (2008) Capacitated lot-sizing with extensions: a review. 4OR Q J Oper Res 6:61–83
Rogers J (1958) A computational approach to the economic lot scheduling problem. Manage Sci 4:264–291
Rosling K (1986) Optimal lot-sizing for dynamic assembly systems. In: Axsäter S, Schneeweiss C, Silver E (eds) Multi-stage production planning and inventory control. Springer, Berlin, pp 119–131
Salomon M (1991) Deterministic lotsizing models for production planning. Lecture notes in economics and mathematical systems, vol 355. Springer, Berlin
Salomon M, Kroon LG, Kuik R, van Wassenhove LN (1991) Some extensions of the discrete lotsizing and scheduling problem. Manage Sci 37:801–812
Salomon M, Kuik R, van Wassenhove LN (1993) Statistical search methods for lotsizing problems. Ann Oper Res 41:453–468
Sambasivan M, Yahya S (2005) A Lagrangean-based heuristic for multi-plant, multi-item, multi-period capacitated lot-sizing problems with inter-plant transfers. Comput Oper Res 32:537–555
Sambasivan M, Schmidt CP (2002) A heuristic procedure for solving multi-plant, multi-item, multi-period capacitated lot-sizing problems. Asia Pac J Oper Res 19:87–105
Selen WJ, Heuts RM (1989) A modified priority index for Günther's lot-sizing heuristic under capacitated single stage production. Eur J Oper Res 41:181–185
Silver EA, Meal HC (1969) A simple modification of the EOQ for the case of a varying demand rate. Prod Inventory Manage 10:51–55

Silver EA, Meal HC (1973) A heuristic for selecting lot size quantities for the case of a deterministic time-varying demand rate and discrete opportunities for replenishment. Prod Inventory Manage 14:64–74
Sox CR, Gao Y (1999) The capacitated lot sizing problem with setup carry-over. IIE Trans 31:173–181
Stadtler H (1996) Mixed integer programming model formulations for dynamic multi-item multi-level capacitated lotsizing. Eur J Oper Res 94:561–581
Stadtler H (1997) Reformulations of the shortest route model for dynamic multi-item multi-level capacitated lotsizing. OR Spektrum 19:87–96
Stadtler H (2000) Improved rolling schedules for the dynamic single-level lot-sizing problem. Manage Sci 46:318–326
Stadtler H (2003) Multilevel lot sizing with setup times and multiple constrained resources: internally rolling schedules with lot-sizing windows. Oper Res 51:487–502
Staggemeier AT, Clark AR (2001) A survey of lot-sizing and scheduling models. In: 23rd annual symposium of the Brazilian Operational Research Society (SOBRAPO), Campos do Jordao SP, Brazil, pp 938–947
Sürie C, Stadtler H (2003) The capacitated lot-sizing problem with linked lot sizes. Manage Sci 49: 1039–1054
Tempelmeier H, Derstroff M (1993) Mehrstufige Mehrprodukt-Losgrößenplanung bei beschränkten Ressourcen und genereller Erzeugnisstruktur. OR Spektrum 15:63–73
Tempelmeier H, Derstroff M (1996) A Lagrangean-based heuristic for dynamic multilevel multiitem constrained lotsizing with setup times. Manage Sci 42:738–757
Tempelmeier H, Helber S (1994) A heuristic for dynamic multi-item multi-level capacitated lotsizing for general product structures. Eur J Oper Res 75:296–311
Thizy J-M, van Wassenhove LN (1985) Lagrangean relaxation for the multi-item capacitated lot-sizing problem: A heuristic implementation. IIE Trans 17:308–313
Trigeiro WW (1987) A dual-cost heuristic for the capacitated lot sizing problem. IIE Trans 19:67–72
Trigeiro WW (1989) A simple heuristic for lot sizing with setup times. Decis Sci 20:294–303
Trigeiro WW, Thomas LJ, McClain JO (1989) Capacitated lot sizing with setup times. Manage Sci 35: 353–366
van Nunen JAEE, Wessels J (1978) Multi-item lot size determination and scheduling under capacity constraints. Eur J Oper Res 2:36–41
Wagelmans A, van Hoesel S, Kolen A (1992) Economic lot sizing: An o (n log n) algorithm that runs in linear time in the Wagner–Whitin case. Oper Res 40:145–156
Wagner HM, Whitin TM (1958) Dynamic version of the economic lot size model. Manage Sci 5:89–96
Wolsey LA (1995) Progress with single-item lot-sizing. Eur J Oper Res 86:395–401
Xie J, Dong J (2002) Heuristic genetic algorithms for general capacitated lot-sizing problems. Comput Math Appl 44:263–276
Zangwill WI (1966) A deterministic multi-period production scheduling model with backlogging. Manage Sci 13:105–119