Available online at www.sciencedirect.com
ScienceDirect
Omega 37 (2009) 3–15
omega
The International Journal of Management Science
www.elsevier.com/locate/omega

Review

Coordinated deterministic dynamic demand lot-sizing problem: A review of models and algorithms☆

Powell Robinson[a,*], Arunachalam Narayanan[b], Funda Sahin[c]

[a] Information and Operations Management, Texas A&M University, 320 Wehner Building, MS-4217, College Station, TX 77843-4217, USA
[b] Department of Enigneering Technology and Industrial Distribution, Texas A&M University, MS-3367, College Station, TX 77843-3367, USA
[c] Department of Marketing and Logistics, The University of Tennessee, Knoxville, TN 37996-0530, USA

Received 21 July 2006; accepted 16 November 2006
Available online 9 January 2007

Abstract

Due to their importance in industry and mathematical complexity, dynamic demand lot-sizing problems are frequently studied. In this article, we consider coordinated lot-size problems, their variants and exact and heuristic solutions approaches. The problem class provides a comprehensive approach for representing single and multiple items, coordinated and uncoordinated setup cost structures, and capacitated and uncapacitated problem characteristics. While efficient solution approaches have eluded researchers, recent advances in problem formulation and algorithms are enabling large-scale problems to be effectively solved. This paper updates a 1988 review of the coordinated lot-sizing problem and complements recent reviews on the single-item lot-sizing problem and the capacitated lot-sizing problem. It provides a state-of-the-art review of the research and future research projections. It is a starting point for anyone conducting research in the deterministic dynamic demand lot-sizing field.

Keywords: Coordinated lot-sizing; Integer programming; Heuristics; Inventory control

Contents

1. Introduction 4
2. Overview of lot-sizing problems and paper scope 4
   2.1. Deterministic static demand models 4
   2.2. Deterministic dynamic demand models 5
3. Coordinated uncapacitated lot-sizing problem (CULSP) 5
   3.1. Problem formulations 5
      3.1.1. Traditional (TRAD) product unit formulation 6
      3.1.2. Shortest path (SPATH) formulation 6
      3.1.3. Arborescent network (ARBNET) formulation 7
      3.1.4. Exact requirements (EXREQ) formulation 7
   3.2. Solving CULSP with general-purpose software 8

☆ This manuscript was processed by Associate Editor Metters.
* Corresponding author. Tel.: +1 979 845 1148.
E-mail address: p-robinson@mays.tamu.edu (P. Robinson).

0305-0483/$ - see front matter
doi:10.1016/j.omega.2006.11.004

3.3. CULSP heuristics . . . 8
3.3.1. Common sense or specialized heuristics for CULSP . . . 8
3.3.2. Meta-heuristics for CULSP . . . 9
3.3.3. Mathematical programming-based heuristics . . . 9
3.3.4. Comparison of CULSP heuristics . . . 9
4. Coordinated capacitated lot-sizing problem (CCLSP) . . . 10
4.1. CCLSP problem formulations . . . 10
4.1.1. TRAD-C formulation . . . 10
4.1.2. ARBNET-C1 and ARBNET-C2 formulations . . . 11
4.1.3. EXREQ-C formulation . . . 11
4.2. Solving CCLSP with general-purpose software . . . 12
4.3. Mathematical programming-based heuristics . . . 12
4.4. Common sense or specialized heuristics . . . 13
5. Conclusions and directions for future research . . . 13
References . . . 14

1. Introduction

The dynamic demand, coordinated lot-size problem determines the time-phased replenishment schedule (i.e., timing and order quantity) that minimizes the sum of inventory and ordering costs for a family of items. A joint shared fixed setup cost is incurred each time one or more items of the product family are replenished, and a minor setup cost is charged for each item replenished. In addition, a unit cost is applied to each item ordered. Demand is assumed to be deterministic but dynamic over the planning horizon and must be met through current orders or inventory. Coordinated lot-size problems are often encountered in production, procurement, and transportation planning [1–4].

The mathematical complexity of the coordinated lot-size problem is *NP*-complete indicating that it is unlikely that a polynomial bound algorithm will be discovered for its solution. For this reason, a significant literature base detailing alternative mathematical formulations and exact solution approaches for the problem is rapidly evolving in an effort to solve large industry problems, which may include over one hundred items and time periods. However, the most recent review of this rapidly evolving literature is by Aksoy and Erenguc [5] in 1988. This review focuses on the literature for the deterministic, dynamic demand, coordinated lot-sizing problems since 1988. We begin by providing a brief overview of lot-sizing research, which positions this study within the broad context of the lot-sizing literature. Next, we examine alternative problem formulations, exact and heuristic solution approaches, and experimental findings for the uncapacitated and capacitated coordinated lot-size problems, respectively. We conclude with future research directions. This research extends the recent surveys by Brahimi et al. [6] on single-item lot-sizing problems and Karimi et al. [7] on the capacitated lot-sizing problem.

2. Overview of lot-sizing problems and paper scope

A variety of taxonomies are proposed for classifying lot-sizing problems (see [5,7–9]). Karimi et al. [7] review the problem characteristics that affect classifying, modeling and solving lot-sizing problems. An important problem characteristic is the nature of demand. Static demand problems assume a stationary or constant demand pattern, while dynamic demand problems permit demand to vary. If all demand values are known for the duration of the planning horizon, the demand stream is defined as deterministic. Otherwise, the demand is considered to be stochastic. Due to the vastness of the lot-sizing literature, this review only addresses single-level lot-sizing decisions with deterministic demand.

2.1. Deterministic static demand models

Harris [10] introduces the single-item economic order quantity (EOQ) model, which assumes deterministic static demand, continuous time, and an unlimited replenishment lot-size. The objective is to minimize the sum of ordering and inventory holding costs. Numerous extensions of this basic model are proposed including models with gradual replenishment of stock, quantity discounts, and periodic setup costs, among others (see Silver et al. [9]).

The economic lot scheduling problem (ELSP) generalizes the EOQ model to consider multiple items that share a constrained resource. While the EOQ model is simple to solve, the ELSP is *NP*-hard [11,12].

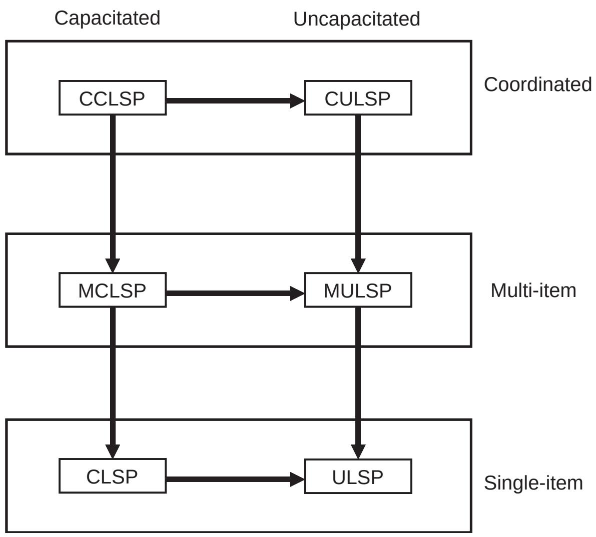

Fig. 1. Taxonomy of deterministic dynamic demand lot-sizing problems.

The joint replenishment problem (JRP) considers a setup cost that is shared among all replenished items, in addition to individual item setup costs. The replenishment quantity of each item is unlimited. Goyal and Satir [13] review the optimal and heuristic procedures proposed for solving the problem up to 1989. More recent and effective heuristic approaches are discussed in [14–17], optimization procedures are proposed in [18,19], and genetic algorithms are investigated in [20].

2.2. Deterministic dynamic demand models

Fig. 1 presents the six most commonly researched deterministic dynamic demand lot-sizing models. The problems are classified according to three factors: (1) single or multiple items, (2) capacitated or uncapacitated replenishment quantities, and (3) joint or independent setup cost structures. The problem classes are represented by nodes and their structural relationships by arcs, where a problem node originating an arc is a generalization of the problem node terminating the arc.

The coordinated capacitated lot-sizing problem (CCLSP) is the most general problem class considering multiple items, product family and individual item setup costs, and a capacity limitation on the maximum number of items that can be replenished in a time period. Relaxing the capacity constraint yields the coordinated uncapacitated lot-sizing problem (CULSP), while removing the joint setup cost yields the multi-item capacitated lot-sizing problem (MCLSP). Limiting the number of items to one, while simultaneously relaxing the joint setup cost and capacity constraint yields the uncapacitated lot-sizing problem (ULSP), the most elemental dynamic demand lot-sizing problem class.

Moving to a more general problem representation complicates the problem's mathematical structure, which requires a more robust solution approach and increases the computational difficulty. The logical goal is to develop an efficient exact solution approach for the CCLSP, which would provide a comprehensive modeling and algorithmic approach for the six problem classes. However, as Karimi et al. [7] indicate, "there is little literature for problems such as CLSP with single-family or multi-family joint setup, in both capacitated and uncapacitated cases. Developing heuristics with reasonable speed and solution quality for these kinds of problems is another interesting research area."

Brahimi et al. [6] provide a current literature review of the single-item uncapacitated (ULSP) and capacitated (CLSP) lot-size problem. Karimi et al. [7] survey the literature for single and multiple item capacitated lot-sizing problems (CLSP and MCLSP). We synthesize the literature on the CULSP focusing on research since the 1988 survey by Aksoy and Erenguc [5] and then examine the rapidly developing body of work on the CCLSP.

3. Coordinated uncapacitated lot-sizing problem (CULSP)

The CULSP's objective is to minimize total system costs, which includes a joint setup cost for each time period any item in the product family is replenished, an item setup cost for each item replenished in each time period and inventory costs. The joint setup cost complicates the solution of the CULSP, which is known to be *NP*-complete [21,22]. However, researchers have exploited specialized problem formulations to obtain efficient algorithms. We present the four most significant problem formulations; the algorithms associated with each, and summarize studies comparing their solution efficiency.

3.1. Problem formulations

Boctor et al. [23] and Narayanan and Robinson [24] study alternative formulations for the CULSP. The more effective formulations use 'disaggregate' variable upper bound constraints on the setup decision variables, which yields a 'tight' linear programming (LP) relaxation and lower bound on the problem.

3.1.1. Traditional (TRAD) product unit formulation

Consider a T-period planning horizon. For $i = 1, \ldots, I$ and $t = 1, \ldots, T$, define, d_{it}, the demand for the item i in period t; s_{it}, setup cost for item i in period t; S_t, joint setup cost in period t; c_{it}, variable per unit cost for item i in period t; and h_{it}, the per unit inventory holding cost for item i in period t. The decision variables include: x_{it}, order size for item i in period t; I_{it}, ending inventory of item i in period t; $Y_{it}=1$ if item i is replenished in period t and $Z_t = 1$ if a joint setup occurs in period t. The mathematical formulation is

$$v(TRAD) = \text{Minimize } Z = \sum_{t=1}^{T} S_t Z_t + \sum_{i=1}^{I}\sum_{t=1}^{T} s_{it} Y_{it} + \sum_{i=1}^{I}\sum_{t=1}^{T} c_{it} x_{it} + \sum_{i=1}^{I}\sum_{t=1}^{T} h_{it} I_{it} \quad (1)$$

$$\text{Subject to } I_{it} = I_{it-1} + x_{it} - d_{it} \quad (i = 1, \ldots, I; t = 1, \ldots, T), \quad (2)$$

$$x_{it} \leqslant \sum_{r=t}^{T} d_{ir} Y_{it} \quad (i = 1, \ldots, I; t = 1, \ldots, T), \quad (3)$$

$$\sum_{i=1}^{I} Y_{it} \leqslant I Z_t \quad (t = 1, \ldots, T), \quad (4)$$

$$X_{it} \geqslant 0, \quad I_{it} \geqslant 0 \quad (i = 1, \ldots, I; t = 1, \ldots, T), \quad (5)$$

$$Y_{it} \in \{0 \text{ or } 1\} \quad (i = 1, \ldots, I; t = 1, \ldots, T), \quad (6)$$

$$Z_t \in \{0 \text{ or } 1\} \quad (t = 1, \ldots, T). \quad (7)$$

The *TRAD* formulation models product unit flows and employs the 'aggregate' or 'weak' variable upper bound constraint sets (3) and (4) to prevent replenishment unless the appropriate setup costs are incurred. Early researchers exploit the 'single sourcing' and 'exact requirement' characteristics of an optimal problem solution with dynamic programming procedures [25–27]. However, computation times increase exponentially with problem size limiting usefulness to problems with only a few items [1]. Haseborg [28] proposes joint ordering policies to mitigate the impact of the number of items. Raghavan [29] proposes a branch-and-cut procedure for a slight modification of the above formulation. Erenguc [30] develops a combined branch-and-bound/dynamic programming procedure based on Veinott's [26] 'major setup pattern' concepts. Federgruen and Tzur [31] describe a branch-and-bound technique whose upper bound is generated by a greedy-add heuristic and a tight lower bound is provided by a partitioning heuristic.

3.1.2. Shortest path (SPATH) formulation

Joneja [22] proposes an integer programming formulation, which models the problem as I independent ULSPs that are coupled by the joint setup decision variables. The formulation exploits the 'exact requirements' property of Wagner and Whitin [32] to provide a more compact model than the *TRAD* formulation. For a specified setting of the joint setup variables, the resulting ULSPs are easily solved as I independent shortest path problems. Building upon earlier defined parameters and variables, the problem considers I items over a T-period planning horizon with $T' = T + 1$. The demand for item i in periods t' through $t - 1$ is $D_{it't}$. The total cost of ordering $D_{it't}$ units in period t' and serving demand through period $t-1$ for item i is $C_{it't} = s_{it'} + c_{it'} D_{it't} + \sum_{r=t'+1}^{t-1} h_{i,t'-1} D_{it't}$. The decision variable $Y_{it't} = 1$ if $D_{it't}$ units of item i are ordered in period t', and 0 otherwise. $Z_{t'} = 1$ if a major setup is scheduled in period t', and 0 otherwise. The *SPATH* formulation is

$$v(SPATH) = \text{Minimize } Z = \sum_{t'=1}^{T} S_{t'} Z_{t'} + \sum_{i=1}^{I}\sum_{t'=1}^{T}\sum_{t=t'+1}^{T'} C_{it't} Y_{it't} \quad (8)$$

$$\text{Subject to } \sum_{t'=1}^{t-1} Y_{it't} - \sum_{t=t'+1}^{T'} Y_{it't} = 0 \quad (i = 1, \ldots, I; t = 2, \ldots, T), \quad (9)$$

$$\sum_{t'=1}^{T} Y_{it'T'} = 1 \quad (i = 1, \ldots, I), \quad (10)$$

$$Z_{t'} - \sum_{t=t'+1}^{T'} Y_{it't} \geqslant 0 \quad (i = 1, \ldots, I; t' = 1, \ldots, T), \quad (11)$$

$$Y_{it't} \in \{0 \text{ or } 1\} \quad (i = 1, \ldots, I; t' = 1, \ldots, T; t = t' + 1, \ldots, T'), \quad (12)$$

$$Z_{t'} \in \{0 \text{ or } 1\} \quad (t' = 1, \ldots, T). \quad (13)$$

A notable feature of the *SPATH* formulation is the 'disaggregate' upper bound constraint set (11). This yields more constraints than the 'aggregate' formulation, but provides a tighter LP relaxation. Kirca [33] develops a dual-based branch-and-bound procedure solving problems with 24 time periods and 50 items.

3.1.3. Arborescent network (ARBNET) formulation

Robinson and Gao [34] present an arborescent fixed-charge network programming formulation for the problem. Supplementing earlier notation, define $h_{it't} = \sum_{r=t'}^{t-1} h_{ir}$ as the per unit inventory holding cost for serving demand for item i in period t from a replenishment order in period t'; $Y_{it'} = 1$ if item i is replenished period t', and 0 otherwise; and $X_{it't}$ is the portion of demand for item i in period t that is served from a replenishment order in period t'. The *ARBNET* formulation is

$v(ARBNET)$

$$=\text{Minimize}\ \ Z = \sum_{t'=1}^{T} S_{t'} Z_t + \sum_{i=1}^{I}\sum_{t'=1}^{T} s_{it'} Y_{it'} + \sum_{i=1}^{I}\sum_{t'=1}^{T-1}\sum_{t=t'}^{T} (c_{it'} + h_{it't}) d_{it} X_{it't} \quad (14)$$

$$\text{Subject to}\ \ \sum_{t'=1}^{t} X_{it't} = 1 \quad (i = 1, \ldots, I, t = 1, \ldots, T), \quad (15)$$

$$Y_{it'} \leqslant Z_{t'} \quad (i = 1, \ldots, I, t' = 1, \ldots, T), \quad (16)$$

$$X_{it't} \leqslant Y_{it'} \quad (i = 1, \ldots, I, t' = 1, \ldots, T, t = t', \ldots, T), \quad (17)$$

$$Z_{t'} \in \{0 \text{ or } 1\} \quad (t' = 1, \ldots, T), \quad (18)$$

$$Y_{it'} \in \{0 \text{ or } 1\} \quad (i = 1, \ldots, I, t' = 1, \ldots, T), \quad (19)$$

$$X_{it't} \in \{0 \text{ or } 1\} \quad (i = 1, \ldots, I, t' = 1, \ldots, T, t = t', \ldots, T). \quad (20)$$

The attractive feature of this model is the hierarchical linking of the decision variables by the disaggregate variable upper bound constraints (16)–(18). This tightly constrains the setup variables to take on a value of 0 or 1 in the optimal solution of the LP relaxation. Robinson and Gao [34] exploit this structure with a dual-ascent-based branch-and-bound method and solve problems with up to 24 (36) time periods and 40 (20) items. Including backorders requires a 30% increase in computational time. Raghavan [29] proposes a Dantzig–Wolfe decomposition approach for the formulation but provides limited results.

3.1.4. Exact requirements (EXREQ) formulation

Boctor et al. [23] also view the CULSP as I linked Wagner and Whitin [32] problems. Their formulation further exploits the 'exact requirements' property of Wagner and Whitin [32] yielding a more compact problem representation than the *SPATH* model. The binary decision variable $w_{it't} = 1$ if and only if a replenishment is scheduled in time t' to cover the demand for item i from period t' through period t. Define $C'_{it't} = s_{it'} + c_{it'} \sum_{r=t'}^{t} d_{ir} + \sum_{r=t'+1}^{t} (\sum_{k=t'}^{r-1} h_{ik}) d_{ir}$ as the sum of the item production and inventory costs associated with $w_{it't}$. Boctor et al. [23] formulation, with disaggregate variable upper bound constraints (23) for tightening the LP lower bound as suggested by Narayanan and Robinson [24] is

$v(EXREQ)$

$$=\text{Minimize}\ \ Z = \sum_{t'=1}^{T} S_{t'} Z_{t'} + \sum_{i=1}^{I}\sum_{t'=1}^{T}\sum_{t=t'}^{T} C_{it't} w_{it't} \quad (21)$$

$$\text{Subject to}\ \ \sum_{r=1}^{t'}\sum_{t=t'}^{T} w_{irt} = 1 \quad (i=1, \ldots, I, t'=1, \ldots, T), \quad (22)$$

$$\sum_{t=t'}^{T} w_{it't} \leqslant Z_{t'} \quad (i=1, \ldots, I, t'=1, \ldots, T), \quad (23)$$

$$w_{it't} \in \{0 \text{ or } 1\} \quad (i = 1, \ldots, I, t' = 1, \ldots, T, t = 1, \ldots, T), \quad (24)$$

$$Z_{t'} \in \{0 \text{ or } 1\} \quad (t'=1, \ldots, T). \quad (25)$$

EXREQ tightly constrains the joint setup variable $Z_{t'}$ to take on a value of 1 when any item is replenished in time t' and is a potentially attractive, but unexplored, formulation for algorithm development.

3.2. Solving CULSP with general-purpose software

In contrast to specialized solution methods, general-purpose mathematical programming software provides an off-the-shelf approach for the problem. Boctor et al. [23], Narayanan and Robinson [24] and Gao et al. [35] evaluate the effectiveness of solving CULSP with CLPEX and Xpress-MP finding that the particular problem formulation has a significant impact on computational efficiency. Specifically, the aggregate representation of the variable upper bound constraints, such as in *TRAD*, provides a computationally inefficient problem representation due to the weak lower bound provided by the LP relaxation. On the other hand, the equally tight *SPATH*, *ARBNET*, and *EXREQ* formulations, each of which provides a different approach for defining the convex-hull relaxation of CULSP, have an average LP optimality gap of 0.001% with approximately 97.75% of the LP solutions being integer and thus optimal [24]. However, solutions times for the tight formulations also vary considerably.

When using general-purpose software, Gao et al. [35] find that the *ARBNET* formulation is more efficient than the *SPATH* formulation, while Narayanan and Robinson [24] document the superiority of *EXREQ* over *ARBNET*. Using XpressMP running on a Pentium 4 at 1.9 GHz to solve problem sizes with (I, T) ranging from (5, 6) to (40, 48) average solution times for *EXREQ* are 0.53 CPU seconds while the longest solution time for a specific combination of factors is 12.44 CPU seconds.

Comparing general-purpose and specialized solution approaches, general-purpose software requires approximately 35 times more effort than Robinson and Gao's [34] dual-ascent-based procedure, which is the best known exact approach for CULSP [35].

3.3. CULSP heuristics

Heuristic approaches for CULSP are classified as common sense or specialized heuristics, meta-heuristics and mathematical programming-based heuristics. Table 1 lists the major articles reviewed in this section and their associated research methodologies.

3.3.1. Common sense or specialized heuristics for CULSP

Common sense heuristics include myopic single-strategy, period-by-period and improvement heuristics.

Myopic single-strategy heuristics: Recognizing that the allocation of the joint setup cost among the items is minimized when all items are simultaneously replenished, the myopic single-strategy heuristics order each item every time there is a joint setup. Fogarty and Barringer [36] propose a Wagner and Whitin [32] type model and dynamic programming algorithm assuming all items are ordered at each joint setup. As expected, the heuristic produces high quality solutions under relatively high fixed cost ratios (i.e., the joint setup cost divided by the sum of the item setup costs), but solution quality suffers under low fixed cost ratios. Atkins and

Table 1
Coordinated uncapacitated lot-sizing problem—exact and heuristic methods

Authors	Solution method
Zangwill [25]	Dynamic programming using dominant extreme point sets
Veinott [26]	Leontief substitution models and dynamic programming
Kao [27]	Dynamic programming using regenerations points to limit the search for optimality
Silver [1]	Dynamic programming using four optimality properties to limit the search
Haseborg [28]	Dynamic programming
Fogarty and Barringer [36]	Dynamic programming
Silver and Kelle [37]	Right shift improvement procedure
Atkins and Iyogun [38]	Silver and Meal-based forward-pass heuristic
Erenguc [30]	Branch-and-bound with shortest path sub-problems
Joneja [22]	Branch-and-bound using cost covering heuristics
Iyogun [39]	Part-period balancing-based forward-pass heuristic
Raghavan [29]	Branch-and-cut and Dantzig/Wolfe
Federgruen and Tzur [31]	Branch-and-bound using a partitioning heuristic for lower bounding and a greedy-add heuristic for upper bounding
Kirca [33]	Dual-based branch-and-bound
Robinson and Gao [34]	Dual-ascent-based branch-and-bound
Boctor et al. [23]	Perturbation meta-heuristic
Robinson et al. [40]	Forward-pass, two-phase improvement, and simulated annealing meta-heuristic

Iyogun [38] extend the Silver and Meal [41] single-item heuristic to the problem assuming that each item is produced at every joint setup. Similarly, Iyogun [39] adapts the part-period balancing method [42] to the problem and provides an improved version of the Atkins and Iyogun [38] heuristic.

Federgruen and Tzur's [31] greedy-add heuristic starts with a replenishment in period 1 that covers the demand for all items over the planning horizon. Next, a greedy-add procedure iteratively schedules additional replenishment periods, if they yield cost savings. Boctor et al. [23] propose a greedy-drop heuristic, which starts with a replenishment in each time period and iteratively eliminates the replenishments generating the greatest cost savings.

Period-by-period: Period-by-period heuristics shift orders into earlier time periods if the setup cost saving exceeds the increase in inventory holding costs. Forward-pass heuristics begin in period 1 and construct replenishment schedules moving forward through time. A savings-based decision criterion guides the selection of which demand to move into the lot-sizes being scheduled. Robinson et al. [40] propose forward-pass heuristics based on modified Eisenhut [43] and Lambrecht and Vanderveken [44] decision criterion.

Joneja [22] develops a forward-pass 'cost covering' heuristic that places an item order in time t when the inventory holding cost for serving the demand of a candidate item in time t exceeds its ordering cost. Similarly, a joint order is scheduled in time t when the total inventory holding costs of all the candidate items exceeds their total ordering cost plus the joint setup cost.

Improvement heuristics: Improvement approaches attempt to improve upon an existing solution. Robinson et al. [40] propose a two-phase greedy heuristic extending the concepts of Dogramaci et al. [45] to consider joint setup costs. The heuristic begins with an initial lot-for-lot solution. Next, operating in a greedy iterative manner, Phase I attempts to shift orders into earlier time periods, not necessarily moving sequentially through time. Upon completion of Phase I, Phase II attempts to improve upon the solution by right-shifting orders later in time to lower inventory costs.

Silver and Kelle [37] describe an improvement heuristic, considers whether a cost saving can be achieved by incorporating the production of each item into its prior scheduled order. When applied as an improvement step for the myopic Fogarty and Barringer [36] heuristic, solution quality improves considerably.

3.3.2. Meta-heuristics for CULSP

Traditional heuristics tend to converge at a local optimum, leaving neighborhoods of the problem's state space unexplored. In such cases, meta-heuristics, which coordinate the search process to escape from local optima and perform a more robust search of the problem's feasible region, are attractive. Boctor et al. [23] propose a perturbation meta-heuristic based on the Fogarty and Barringer (FB) and Silver and Kelle (SK) heuristics. The perturbation meta-heuristic contains four basic components: (1) the FB–SK heuristics provide a starting solution, (2) a perturbation procedure jumps to other regions of the solution state space, (3) a greedy-drop heuristic eliminates joint setups that are not economically justified, and (4) the SK improvement heuristic. The perturbation procedure provides a 5.5% improvement over the stand alone FB–SK heuristic [40].

Robinson et al. [40] propose a simulated annealing meta-heuristic that uses the two-phase heuristic to generate the starting solution and to find a high quality solution in each newly generated neighborhood of the feasible region. The simulated annealing meta-heuristic is particularly effective improving upon the initial two-phase solution by 63.6%. Koulamas et al. [46] survey simulated annealing applications to other operations research problems.

3.3.3. Mathematical programming-based heuristics

Robinson and Gao [34] propose using their dual-ascent-based procedures as a heuristic by terminating the solution procedure at node zero of the branch-and-bound tree and reporting the candidate solution. The heuristic provides a feasible and lower bound solution, thereby permitting a statement of the worst case quality of the heuristic solution. Other mathematical programming heuristics are described in [22,31,33].

3.3.4. Comparison of CULSP heuristics

Boctor et al. [23] and Robinson et al. [40] evaluate the performance of the CULSP heuristics. The optimality gaps and the number of optimal solutions found by the best performing heuristics as presented in [40] are summarized in Table 2 for 1600 problem instances, which vary from 5 to 40 items and 6 to 48 time periods. The simulated annealing meta-heuristic's performance is superior on all performance metrics with an average optimality gap of 0.21%, a maximum optimality gap of 3.95%, and finding 780 optimal solutions to the 1600 test problems. Even though Robinson and Gao [34] dual-ascent heuristic finds the second highest number of optimal solutions, it has the worst overall average and maximum optimality gaps. Using a personal

Table 2
Experimental results for CULSP heuristic procedures

CULSP heuristic	Optimality gap[a]		Number of optimal solutions[c]
	Average (%)	Maximum[b] (%)	
FB–SK heuristic	0.92	12.64	502
Dual-ascent heuristic	1.95	35.43	725
Two-phase heuristic	0.56	9.12	602
Perturbation meta-heuristic	0.87	9.56	517
Simulated annealing meta-heuristic	0.21	3.95	780

[a]Optimality gap = 100(heuristic objective value − optimal objective value)/optimal objective value.
[b]Largest optimality gap associated with a combination of experimental factors.
[c]Each cell represents the results for 1600 test problems.

computer with a Pentium 4 processor at running at 1.9 GHz with the Windows 2000 Professional operating system, solution times for each of the stand alone heuristics average less than 0.05 CPU seconds per problem. The simulated annealing meta-heuristic is the most time intensive procedure and averages 0.18 CPU seconds with a maximum of 1.8 CPU seconds.

4. Coordinated capacitated lot-sizing problem (CCLSP)

The CCLSP contains both the complicating constraints associated with capacitated replenishment and the joint setup decision variables resulting in a *NP*-complete problem. We present four alternative mathematical formulations, the algorithms associated with each, and compare their computational efficiency.

4.1. CCLSP problem formulations

The mixed integer programming formulations discussed in this section build upon the notation described in the previous section.

4.1.1. TRAD-C formulation

Federgruen et al. [47] extend the *TRAD* formulation to consider capacity constraints on the replenishment quantity available in each time period. Define, $P_{t'}$ as the available capacity in period t' and $D_t = \sum_{i=1}^{I} d_{it}$ as the aggregate demand in period t. $I_{t'}^0$ is the minimum total inventory on hand at the end of period t' that is required to guarantee that a feasible replenishment schedule exists for periods $t'+1, \ldots, T$. The values for $I_{t'}^0$ are calculated by recursion starting from period T and moving backwards using the following definition, $I_{t'}^0 = (D_{t'+1}^0 - P_{t'+1} + I_{t'+1}^0)^+$ for all $t' = 1, \ldots, T$ with $I_T^0 = 0$. The *TRAD-C* formulation is

$$v(TRAD\text{-}C) = \text{Minimize } Z = \sum_{t'=1}^{T} S_{t'} Z_{t'} + \sum_{i=1}^{I} \sum_{t'=1}^{T} s_{it'} Y_{it'} + \sum_{i=1}^{I} \sum_{t'=1}^{T} c_{it'} x_{it'} + \sum_{i=1}^{I} \sum_{t'=1}^{T} h_{it'} I_{it'} \quad (26)$$

$$\text{Subject to } I_{it'} = I_{it'-1} + x_{it'} - d_{it'} \quad (i = 1, \ldots, I;\ t' = 1, \ldots, T), \quad (27)$$

$$x_{it'} \leqslant P_{t'} Y_{it'} \quad (i = 1, \ldots, I;\ t' = 1, \ldots, T), \quad (28)$$

$$\sum_{i=1}^{I} x_{it'} \leqslant P_{t'} Z_{t'} \quad (t'=1, \ldots, T), \quad (29)$$

$$Y_{it'} \leqslant Z_{t'} \quad (i = 1, \ldots, I;\ t' = 1, \ldots, T), \quad (30)$$

$$\sum_{i=1}^{I} I_{it'} \geqslant I_{t'}^0 \quad (t' = 1, \ldots, T), \quad (31)$$

$$I_{it'} \geqslant 0 \quad (i = 1, \ldots, I;\ t' = 1, \ldots, T), \quad (32)$$

$$x_{it'} \geqslant 0 \quad (i = 1, \ldots, I;\ t' = 1, \ldots, T), \quad (33)$$

$$Y_{it'} \in \{0 \text{ or } 1\} \quad (i = 1, \ldots, I;\ t' = 1, \ldots, T), \quad (34)$$

$$Z_{t'} \in \{0 \text{ or } 1\} \quad (t'=1, \ldots, T). \quad (35)$$

4.1.2. ARBNET-C1 and ARBNET-C2 formulations

Robinson and Lawrence [4] append the following capacity constraint set to the *ARBNET* formulation to obtain *ARBNET-C*1:

$$\sum_{t=t'}^{T}\sum_{i=1}^{I} X_{it't}d_{it} \leqslant P_{t'} \quad (t'=1,\ldots,T). \tag{36}$$

Gao and Robinson [48] tighten the formulation by incorporating the joint setup decision variable, $Z_{t'}$, into the right-hand side of the capacity constraint yielding constraint set (37) and the *ARBNET-C*2 formulation

$$\sum_{t=t'}^{T}\sum_{i=1}^{I} X_{it't}d_{it} \leqslant P_{t'}Z_{t'} \quad (t'=1,\ldots,T). \tag{37}$$

*ARBNET-C*2 provides the convex envelope relaxation, E(C2), as shown below, which gives a tighter LP relaxation than the *ARBNET-C*1 convex envelope relaxation, E(C1). Gao and Robinson [48] and Denizel et al. [49] provide addition details:

$$\begin{aligned}\mathrm{E}(C2) = \mathrm{Min} \sum_{t'=1}^{T} S_{t'} \,\mathrm{Max}\Bigg\{ & X_{it't}, \forall i=1,\ldots,I, \\ & t=t',\ldots,T, \frac{1}{P_{t'}}\sum_{t=t'}^{T}\sum_{i=1}^{I} X_{it't}d_{it}\Bigg\} \\ & + \sum_{i=1}^{I}\sum_{t'=1}^{T} s_{it}\,\mathrm{Max}\{X_{it't}, \forall i=1,\ldots,I\} \\ & + \sum_{i=1}^{I}\sum_{t'=1}^{T-1}\sum_{t=t'}^{T}(c_{it'}+h_{it't})d_{it}X_{it't},\end{aligned}$$

$$\begin{aligned}\mathrm{E}(C1) = \mathrm{Min} \sum_{t'=1}^{T} S_{t'} \,\mathrm{Max}\{ & X_{it't}, \forall i=1,\ldots,I, \\ & t=t',\ldots,T\} \\ & + \sum_{i=1}^{I}\sum_{t'=1}^{T} s_{it}\mathrm{Max}\{X_{it't}\forall i=1,\ldots,I\} \\ & + \sum_{i=1}^{I}\sum_{t'=1}^{T-1}\sum_{t=t'}^{T}(c_{it'}+h_{it't})d_{it}X_{it't}.\end{aligned}$$

4.1.3. EXREQ-C formulation

We propose a new problem formulation for CCLSP. Item setup cost $s_{it'}$ and decision variable $Y_{it'}$ are introduced to decouple the item and family setup constraints (40) and (41). Define $w'_{it't}$ as the fraction of the total demand for item i from period t' to period t that is served from an order in period t' and $\widehat{C}_{it't}$ as the sum of the variable per unit order and inventory holding costs for producing item i in period t' and covering its demand from period t' through t, where $\widehat{C}_{it't}=\sum_{q=t'+1}^{t}(c_{it'}+\sum_{k=t'}^{q-1}h_{ik})d_{iq}$. The formulation is

$$\begin{aligned} v(EXREQ\text{-}C) &= \mathrm{Minimize}\ Z = \sum_{t'=1}^{T} S_{t'}Z_{t'} + \sum_{t'=1}^{T}\sum_{i=1}^{I} s_{it'}Y_{it'} \\ &\quad + \sum_{i=1}^{I}\sum_{t'=1}^{T}\sum_{t=t'}^{T} \widehat{C}_{it't}\, w'_{it't} \end{aligned} \tag{38}$$

Subject to
$$\sum_{t'=1}^{r}\sum_{t=r}^{T} w'_{it't} = 1 \quad (i=1,\ldots,I, r=1,\ldots,T), \tag{39}$$

$$\sum_{t=t'}^{T} w'_{it't} \leqslant Y_{it'} \quad (i=1,\ldots,I, t'=1,\ldots,T), \tag{40}$$

$$Y_{it'} \leqslant Z_{t'} \quad (i=1,\ldots,I, t'=1,\ldots,T), \tag{41}$$

$$\sum_{i=1}^{I}\sum_{t=t'}^{T} w'_{it't}\left(\sum_{q=t'}^{t} d_{iq}\right) \leqslant P_{t'}Z_{t'} \quad (t'=1,\ldots,T), \tag{42}$$

$$\sum_{t'=1}^{q}\sum_{t=q+1}^{T}\sum_{i=1}^{I} w'_{it't}\left(\sum_{r=t'+1}^{t} d_{ir}\right) \geqslant I_q^0 \quad (q=1,\ldots,T), \tag{43}$$

$$0 \leqslant w'_{it't} \leqslant 1 \quad (i=1,\ldots,I, t'=1,\ldots,T, t=1,\ldots,T), \tag{44}$$

$$Y_{it'} \in \{0 \text{ or } 1\} \quad (i=1,\ldots,I, t'=1,\ldots,T), \tag{45}$$

$$Z_{t'} \in \{0 \text{ or } 1\} \quad (t'=1,\ldots,T). \tag{46}$$

The two important modeling features are compact structure of constraint set (39), which insures that all demand is met, and the surrogate aggregate inventory constraint (43).

4.2. Solving CCLSP with general-purpose software

We conducted a computational efficiency study of CCLSP following the design in [24] and [47]. The experiments include three factors that are known to impact solution difficulty. The factors include two levels of demand density (DD), four levels of capacity utilization (CU), and three levels of time-between-orders (TBO) resulting in 24 different combinations of factor settings. Demand density, DD $\in$ {0.50, 1.0}, is the fraction of time periods an item experiences demand. CU, defined as the ratio of total demand divided by the total available capacity over the planning horizon, is taken from the set, CU $\in$ {0.2, 0.4, 0.6, 0.8}. Item and family TBOs are calculated as follows: for item i, $\mathrm{TBO}_i = \sqrt{2s_{it'}/h_i\overline{d}_i}$, where h_i is the per unit inventory holding cost for item i, $\overline{d}_i$ is the expected demand for item i over the planning horizon, and $s_{it'}$ is the item setup cost. For each item, the TBO_i values are randomly drawn from a uniform distribution on the interval [2,6] and then the corresponding value of $s_{it'}$ is determined. For all items, $h_i = 1$ to insure $s_{it'}$ corresponds to the randomly generated TBO_i values. The joint setup cost, $S_{t'}$ is similarly computed with $\mathrm{TBO}_{\mathrm{family}} = \sqrt{2S_{t'}/h_i\overline{D}}$, where $\overline{D}$ is the expected demand for the product family over the planning horizon. $\mathrm{TBO}_{\mathrm{family}} \in$ {low, medium and high}, where the values are randomly generated from a uniform distribution on the intervals [1,3], [2,6] and [5,10], respectively. Within a test problem $S_{t'}$ and $s_{it'}$ are constant over time.

Each test problem has 10 items and a 12-period planning horizon. The demand, d_{it}, is normally distributed and varies by item and time period. Odd numbered items have a mean demand of 50 units and a standard deviation of 20 units; even numbered items have a mean demand of 100 units and a standard deviation of 20 units. Unit production costs are set equal to zero.

For each combination of factors, three test problems are randomly generated and solved to control for random effects. The experiments are conducted on a personal computer with a Pentium®4 processor running at 1.9 GHz with the Windows 2000 Professional operating system, and solved using Xpress-MP version 2005A (Xpress Optimizer v16.01.02), a state-of-the art general-purpose optimization software. For those problems not solving within 120 CPU minutes, the associated 'ending' gap (i.e., 100 (best *integer* solution −LB)/LB) is reported as a metric for evaluating solution quality.

Table 3 summarizes the results by DD, where each cell is associated with 36 test problems. While *ARBNET-C*2 and *EXREQ-C* provide the tightest and identical average LP gaps (i.e., lower bound on the optimal objective function value), *ARBNET-C*2, with an average time of 281 CPU seconds, is the only formulation which solves all of the test problems. *TRAD-C* yields the worst lower bound (55%) and could not solve 30 of the 72 test problems. Other findings not reported here, reveal that CPU times increase rapidly with an increase in I or T for all of the formulations. These findings document the importance of selecting the best formulation for general-purpose software and developing specialized exact and heuristic algorithms for the problem. More detailed experimental results are available from the authors on request.

4.3. Mathematical programming-based heuristics

Federgruen et al. [47] develop a strict partitioning (SP) and a progressive interval/expanding horizon (EH) heuristic. Higher levels of CU and longer TBOs adversely impact the quality of the heuristic solutions. The SP heuristic is computationally efficient, but has an average optimality gap of 14.7%. In contrast, the EH heuristic finds solutions with an average optimality gap of 1.2%, but computational time increases rapidly with problem size. For example, a 10-item and 10-period problem requires 30 s, while a 25-item and 10-period problem requires 5 h and 30 min.

Altay [50] proposes a cross decomposition procedure for the problem. Experimental findings show that the problem is easier to solve when setup costs are negligible and becomes substantially difficult when the ratio of joint setup cost to total cost increases. The algorithm is only capable of solving very small problems.

Robinson and Lawrence [4] develop a Lagrangian relaxation heuristic for *ARBNET-C*1 with backorders. The algorithm relaxes the assignment constraint (Eq. (15)) and the capacity constraint (Eq. (36)) yielding easily solvable sub-problems. The best found solution at node zero of the branch-and-bound tree is reported as the heuristic solution. Computational experiments yield heuristic solutions with average optimality gaps of 0.44%, 3.9%, and 4.72% at the 5%, 45% and 85% CU levels, respectively.

Lawrence [51] proposes a second Lagrangian relaxation heuristic using *ARBNET-C*1, in which only the assignment constraint is relaxed. The resulting Lagrangian sub-problem is a fixed-charge knapsack problem, which produces a tighter lower bound but is more difficult to solve than the Robinson and Lawrence's [4] sub-problems. This heuristic finds solutions with average optimality gaps of 0.15%, 1.73%, and 1.60% at the 5%, 45% and 85% CU levels, respectively. However, CPU requirements increase rapidly with problem size and

Table 3
Summary of CCLSP formulations performance by demand density

CCLSP formulation	Average LP gap[a]		Average CPU time in seconds		No. of problems not solved[b]		Average ending gap[c]	
	DD = 0.5 (%)	DD = 0.1 (%)	DD = 0.5	DD = 0.1	DD = 0.5	DD = 0.1	DD = 0.5 (%)	DD = 0.1 (%)
TRAD-C	55.40	55.04	548.04	6159.31	2	28	3.25	8.32
*ARBNET-C*1	14.67	17.80	357.73	102.77	1	0	0.83	—
*ARBNET-C*2	3.85	3.19	465.98	96.46	0	0	—	—
EXREQ-C	3.85	3.19	553.00	49.90	1	0	2.72	—

[a]LP gap = 100(optimal objective value—LP objective value)/optimal objective value.
[b]Number of problems out of 36 test problems that did not solve to optimality within 120 CPU minutes.
[c]Ending gap = 100(best integer solution objective value—lower bound)/lower bound.

Table 4
Coordinated capacitated lot-sizing problem—exact and heuristic methods

Authors	Solution method
Lawrence [51]	Lagrangian relaxation of assignment constraint
Altay [50]	Cross decomposition
Robinson and Lawrence [4]	Lagrangian relaxation of assignment and capacity constraint
Gao and Robinson [48]	Lagrangian relaxation of capacity constraint
Federgruen et al. [47]	Progressive interval heuristics
Narayanan and Robinson [52]	Six-phase construction heuristic and simulated annealing meta-heuristic

item setup cost making the procedures ineffective for problems with more than 10 items and 12 time periods.

Gao and Robinson [48] describe a Lagrangian dual-ascent heuristic based on relaxing the capacity constraint in *ARBNET-C*2. The heuristic finds solutions with an average optimality gap of 0.67%. While performance declines as CU and joint setup cost increase, it still yields high quality solutions. Problems with an 80% CU are solved in 2.5 CPU seconds with an optimality gap of 1.36%.

4.4. Common sense or specialized heuristics

Narayanan and Robinson [52] propose a six-phase construction heuristic that builds upon [24,45]. The heuristic finds solutions superior to those found by the Lagrangian heuristics in [4,48]. The authors also incorporate the six-phase heuristic into a simulated annealing meta-heuristic that improves solutions by approximately 50%. The EH heuristic of Federgruen et al. [47] provides the highest quality solutions of any known heuristic, but its high computational requirements make it impractical for industry size problems. In contrast, CPU requirements for the simulated annealing meta-heuristic are relatively invariant across problem sizes requiring approximately 0.25 CPU seconds on average with an average optimality gap of 0.43%.

Table 4 lists the major articles, by author, for CCLSP and the associated research methodologies.

5. Conclusions and directions for future research

Due to their importance in industry and mathematical complexity, deterministic, dynamic demand lot-sizing problems are frequently studied. Researchers typically develop specialized formulations and solution procedures for each particular lot-sizing problem class. However, the CCLSP provides a comprehensive modeling framework for single and multiple items, coordinated and uncoordinated scheduling, and capacitated and uncapacitated problem variants. This paper synthesizes the research on this important problem class updating the survey by [5] to consider recent modeling and algorithmic advancements. The paper complements the recent reviews by [6] on the single-item lot-sizing problem and [7] for the capacitated lot-sizing problem to provide a complete picture of state-of-the-art research in this emerging area. This review is a starting point for anyone conducting research in the deterministic dynamic demand coordinated lot-sizing field.

The literature review indicates the existence of several efficient and effective problem formulations, heuristics, and exact approaches for the CULSP, but the CCLSP still poses many challenges for researchers. However, the literature presents a variety of formulations, whose mathematical structure can potentially be exploited leading to an efficient exact approach for the problem. In addition we propose a new formulation, *EXREQ-C*. The *EXREQ-C* and *ARBNET-C*2 formulations provide equally tight LP relaxations defining the convex-hull linear relaxation of CCLSP. As such, they provide the strongest known lower bound for possible inclusion in an exact solution methodology. In addition, several heuristic procedures are identified, which can provide high quality candidate solutions for upper bounding in a branch-and-bound procedure.

Other promising research areas are available. While genetic algorithms, tabu search [53] and capacitated network flow models [54] are successfully applied to solve other lot-size problems, their potential to solve CCLSP is unknown. Research examining sensitivity analysis of dynamic lot-sizing heuristics within the context of CCLSP is also worthwhile [55]. Finally, extending the CCLSP problem representation to capture the impact of equipment downtime on capacity during item changeover and multiple product families are important research areas (see [49,56]).

References

[1] Silver EA. Coordinated replenishments of items under time varying demand: dynamic programming formulation. Naval Research Logistics Quarterly 1979;26:141–51.

[2] Stowers CL, Palekar US. Lot sizing with strong set-up interactions. IIE Transactions 1997;29:167–79.

[3] Shapiro R, Rosenfield D, Stecke KE. Baker Precision Instruments, Inc. HBS-9-687-052. Boston, MA: Harvard Business School Publishing; 2002.

[4] Robinson EP, Lawrence FB. Coordinated capacitated lot-sizing problem with dynamic demand: a Lagrangian heuristic. Decision Sciences 2004;35:25–54.

[5] Aksoy Y, Erenguc SS. Multi-item inventory models with coordinated replenishments: a survey. International Journal of Production Management 1988;8:63–73.

[6] Brahimi N, Dauzere-Peres S, Najid NM, Nordli A. Single item lot sizing problems. European Journal of Operational Research 2006;168:1–16.

[7] Karimi B, Fatemi Ghomi SMT, Wilson JM. The capacitated lot sizing problem: a review of models and algorithms. Omega 2003;31:365–78.

[8] Drexl A, Kimms A. Lot sizing and scheduling—survey and extensions. European Journal of Operational Research 1997;99:221–35.

[9] Silver EA, Pyke BF, Peterson R. Inventory management and production planning and scheduling. 3rd ed, New York: Wiley; 1998.

[10] Harris FW. How many parts to make at once. Factory, The Magazine of Management 1913;10:135–36, 152.

[11] Hsu WL. On the general feasibility test of scheduling lot sizes for several products on one machine. Management Science 1983;29:93–105.

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

[13] Goyal SK, Satir AT. Joint replenishment inventory control: deterministic and stochastic models. European Journal of Operational Research 1989;38:2–13.

[14] Kaspi M, Rosenblatt MJ. On the economic ordering quantity for jointly replenished items. International Journal of Production Research 1991;29:107–14.

[15] Goyal SK, Deshmukh SG. A note on 'the economic ordering quantity for jointly replenished items'. International Journal of Production Research 1993;31:109–16.

[16] Hariga M. Two new heuristic procedures for the joint replenishment problem. Journal of Operational Research Society 1994;45:463–71.

[17] Ben-Daya M, Hariga M. Comparative study of heuristics for the joint replenishment problem. Omega 1995;23:341–4.

[18] Viswanathan S. A new optimal algorithm for the joint replenishment problem. Journal of the Operational Research Society 1996;47:936–44.

[19] Wildeman RE, Frenk JBG, Dekker R. An efficient optimal solution method for the joint replenishment problem. European Journal of Operational Research 1997;99:433–44.

[20] Khouja M, Michalewicz Z, Satoskar SS. A comparison between genetic algorithms and the RAND method for solving the joint replenishment problem. Production Planning and Control 2000;11:556–64.

[21] Arkin E, Joneja D, Roundy R. Computational complexity of uncapacitated multi-echelon production planning problems. Operations Research Letters 1989;8:61–6.

[22] Joneja D. The joint replenishment problem: new heuristics and worst case performance bounds. Operations Research 1990;38:711–23.

[23] Boctor FF, Laporte G, Renaud J. Models and algorithms for the dynamic demand joint replenishment problem. International Journal of Production Research 2004;42:2667–78.

[24] Narayanan A, Robinson EP. More on 'Models and algorithms for the dynamic demand joint replenishment problem'. International Journal of Production Research 2006;44:383–97.

[25] Zangwill WI. A deterministic multiproduct, multifacility production and inventory system. Operations Research 1966;14:486–507.

[26] Veinott Jr. AF. Minimum concave-cost solution of Leontief substitution models of multi-facility inventory systems. Operations Research 1969;17:262–91.

[27] Kao EPC. A multi-product dynamic lot-size model with individual and joint setup costs. Operations Research 1979;27:279–89.

[28] Haseborg F. On the optimality of joint ordering policies in a multi-product dynamic lot size model with individual and joint set-up costs. European Journal of Operational Research 1982;9:47–55.

[29] Raghavan P. The multi-item lot sizing problem with joint replenishments. PhD dissertation, Leonard N. Stern School of Business, New York University; 1993.

[30] Erenguc SS. Multi-product dynamic lot-sizing model with coordinated replenishments. Naval Research Logistics 1988;35:1–22.
[31] Federgruen A, Tzur M. The joint replenishment problem with the time-varying costs and demands: efficient, asymptotic and ε-optimal solutions. Operations Research 1994;42:1067–86.
[32] Wagner HM, Whitin TM. Dynamic version of economic lot size model. Management Science 1958;5:89–96.
[33] Kirca O. A primal-dual algorithm for the dynamic lot-sizing problem with joint setup costs. Naval Research Logistics 1995;42:791–806.
[34] Robinson EP, Gao L-L. A dual ascent procedure for multi-product dynamic demand coordinated replenishment with backlogging. Management Science 1996;42:1556–64.
[35] Gao L-L, Altay N, Robinson EP. A comparative study of exact algorithms for the coordinated lot-sizing problem with dynamic demand. Working paper, Hofstra University, Long Island, NY; 2004.
[36] Fogarty D, Barringer R. Joint order release decisions under dependent demand. Production and Inventory Management Journal 1987;28:55–61.
[37] Silver EA, Kelle P. More on 'Joint order release decisions under dependent demand'. Production and Inventory Management Journal 1988;29:71–2.
[38] Atkins D, Iyogun P. A heuristic with lower bound performance guarantee for the multi-product lot-size problem. IIE Transactions 1988;20:369–73.
[39] Iyogun P. Heuristic methods for the multi-product dynamic lot size problem. Journal of Operational Research Society 1991;42:889–94.
[40] Robinson EP, Narayanan A, Gao L-L. Effective heuristics for the dynamic demand joint replenishment problem. Journal of Operational Research Society 2006, published online April 19, 2006.
[41] Silver EA, Meal HC. A heuristic for selecting lot size quantities for the case of a deterministic time varying demand rate and discrete opportunities for replenishment. Production and Inventory Management Journal 1973;14:64–74.
[42] De Matteis JJ, Mendoza AG. An economic lot-sizing technique: the part-period balancing algorithm. IBM Systems Journal 1968;7:30–46.
[43] Eisenhut PS. A dynamic lot sizing algorithm with capacity constraints. AIIE Transactions 1975;7:170–6.
[44] Lambrecht MR, Vanderveken H. Heuristic procedures for the single operations, multi-item loading problem. AIIE Transactions 1979;11:319–26.
[45] Dogramaci A, Panayiotopoulos JC, Adam NR. The dynamic lot sizing problem for multiple items under limited capacity. AIIE Transactions 1981;13:294–303.
[46] Koulamas C, Antony S, Jaen R. A survey of simulated annealing applications to operations research problems. Omega 1994;22:41–56.
[47] Federgruen A, Meissner J, Tzur M. Progressive interval heuristics for multi-item lot-sizing problems. Operations Research 2004, forthcoming.
[48] Gao L-L, Robinson EP. A Lagrangian/dual-ascent based heuristic for the capacitated coordinated replenishment with dynamic demand and backlogging. Working paper, Hofstra University, Long Island, NY; 2004.
[49] Denizel M, Erenguc SS, Sherali HD. Convex envelope results and strong formulations for a class of mixed-integer programs. Naval Research Logistics 1996;43:503–18.
[50] Altay N. A cross decomposition algorithm for the capacitated dynamic demand coordinated replenishment problem. PhD dissertation, Texas A&M University, College Station, TX; 2001.
[51] Lawrence FB. A dual solution procedure for the capacitated dynamic demand coordinated replenishment problem. PhD dissertation, Texas A&M University, College Station, 2 TX; 1999.
[52] Narayanan A, Robinson EP. An efficient heuristic for the coordinated capacitated dynamic demand lot-sizing problem. Working paper, Texas A&M University, College Station, TX; 2006.
[53] Karimi B, Fatemi Ghomi SMT, Wilson JM. A tabu search heuristic for solving the CLSP with back logging and set-up carry-over. Journal of the Operational Research Society 2006;57:140–7.
[54] Armentano VA, França PM, Toledo FMB. A network flow model for the capacitated lot-sizing problem. Omega 1999;27: 275–84.
[55] Pan C-H. Sensitivity analysis of dynamic lot-sizing heuristics. Omega 1994;22:251–61.
[56] Erenguc SS, Mercan HM. A multifamily dynamic lot-sizing model with coordinated replenishments. Naval Research Logistics 1990;37:539–58.