Operations Research
Vol. 55, No. 3, May–June 2007, pp. 490–502
ISSN 0030-364X | EISSN 1526-5463 | 07 | 5503 | 0490

DOI 10.1287/opre.1070.0392

Progressive Interval Heuristics for Multi-Item Capacitated Lot-Sizing Problems

Awi Federgruen
Graduate School of Business, Columbia University, 101 Uris Hall, New York, New York 10027,
af7@columbia.edu

Joern Meissner
Department of Management Science, Lancaster University Management School, Room A48, Lancaster, LA1 4YX, United Kingdom,
joe@meiss.com

Michal Tzur
Department of Industrial Engineering, Tel Aviv University, Ramat Aviv, Tel Aviv 69978, Israel,
tzur@eng.tau.ac.il

We consider a family of N items that are produced in, or obtained from, the same production facility. Demands are deterministic for each item and each period within a given horizon of T periods. If in a given period an order is placed, setup costs are incurred. The aggregate order size is constrained by a capacity limit. The objective is to find a lot-sizing strategy that satisfies the demands for all items over the entire horizon without backlogging, and that minimizes the sum of inventory-carrying costs, fixed-order costs, and variable-order costs. All demands, cost parameters, and capacity limits may be time dependent. In the basic joint setup cost (JS) model, the setup cost of an order does not depend on the composition of the order. The joint and item-dependent setup cost (JIS) model allows for item-dependent setup costs in addition to the joint setup costs.

We develop and analyze a class of so-called progressive interval heuristics. A progessive interval heuristic solves a JS or JIS problem over a progressively larger time interval, always starting with period 1, but fixing the setup variables of a progressively larger number of periods at their optimal values in earlier iterations. Different variants in this class of heuristics allow for different degrees of flexibility in adjusting continuous variables determined in earlier iterations of the algorithm.

For the JS-model and the two basic implementations of the progressive interval heuristics, we show under some mild parameter conditions that the heuristics can be designed to be ϵ-optimal for any desired value of $\epsilon > 0$ with a running time that is polynomially bounded in the size of the problem. They can also be designed to be simultaneously asymptotically optimal and polynomially bounded.

A numerical study covering both the JS and JIS models shows that a progressive interval heuristic generates close-to-optimal solutions with modest computational effort and that it can be effectively used to solve large-scale problems.

Subject classifications: inventory, production: multi-item, echelon, approximation, heuristic.
Area of review: Manufacturing, Service, and Supply Chain Operations.
History: Received September 2002; revisions received November 2003, August 2004; accepted November 2004.

1. Introduction

This paper addresses capacitated dynamic lot-sizing models. We consider a family of N items that are produced in the same facility or replenished by the same outside supplier. Demands are specified for each item and each period of a given horizon of T periods. If in a given period an order is placed for some or all of the items, setup costs are incurred. The aggregate order size is constrained by a capacity limit. The objective is to find a lot-sizing strategy that satisfies the demands for all items over the entire horizon without backlogging, and that minimizes the sum of inventory-carrying costs, fixed-order costs, and variable-order costs. All demands, cost parameters, and capacity limits may be time dependent, reflecting, for example, general time series of forecasts, customer orders, seasonal fluctuations of the cost parameters, or changes in the capacity due to new acquisitions or scheduled maintenance.

In the basic model, the setup cost for an order in any given period only depends on the period index, but not on the composition of the order. This assumption is satisfied in many, if not most, practical applications, e.g., when the setup cost represents the fixed cost of dispatching a truck or barge or that of initiating a production run in a batch production facility. We refer to this basic case as the joint setup cost JS model. We extend the model to allow for item-dependent setup costs in addition (or in lieu of) the joint setup costs and refer to this generalized model as the joint and item-dependent setup cost JIS model.

This capacitated dynamic lot-sizing model is one of the most frequently used deterministic inventory-planning models. It needs to be solved repeatedly for each level of a material requirements planning (MRP) or distribution requirements planning (DRP) system, with the orders resulting from the capacitated lot-sizing problem(s) at a given level being used as the demand input parameters for the lot-sizing problem to be solved at the next level. The model represents the fundamental challenge of capacity requirements planning while assessing trade-offs between the costs of holding inventories and the potential of exploiting economies of scale in the procurement costs.

Based on a variety of applications for the BASF and Procter & Gamble corporations as well as a production-distribution problem for the so-called PAMIPS (1995) and MEMIPS (1997) projects, Belvaux and Wolsey (2000, 2001) have developed a prototype optimization system for a class of capacitated multi-item lot-sizing problems that includes the JIS model. The system called *bc-prod* uses the extended modelling and optimization library of XPRESS as its engine, but allows for simplified problem specification and generates various cutting-plane constraints specific to the structure of lot-sizing problems. Bixby (2001) in reviewing the progress and future challenges in CPLEX's mixed-integer programming capabilities emphasizes the importance of supply chain management models and within this area, the class of capacitated multi-item lot-sizing problems as being of prime importance and awaiting algorithmic improvements.

The general model is very complex. Florian et al. (1980) have in fact shown that even the single-item case ($N = 1$) is NP-complete, as opposed to the uncapacitated version that, for a planning horizon of T periods, is solvable in $O(T \log T)$ time (see Federgruen and Tzur 1991, Wagelmans et al. 1992, and Aggarwal and Park 1993), and in $O(T)$ time under some mild assumptions on the data. The difficulty arises in part because under capacity restrictions, it may no longer be optimal to place an order at the last possible time; in other words, it is not possible to confine oneself to so-called zero-inventory ordering policies. Polynomial time algorithms have been developed in the single-item case, but these tend to be time consuming and restricted to special parameter settings only; see Florian and Klein (1971), Bitran and Yanasse (1982), Chung and Lin (1988), and Van Hoesel and Wagelmans (1996). Recently, Van Hoesel and Wagelmans (2001) (and Gavish and Johnson 1990 for a more restricted version of the model) developed a fully polynomial approximation scheme for the general single-item model—i.e., an algorithm that generates an ϵ-optimal solution for any $\epsilon > 0$, in an amount of time that is polynomial in the problem size as well as $1/\epsilon$.

When several items are involved ($N \geqslant 2$), no efficient solution methods are known, with the exception of Anily and Tzur's (2005) dynamic programming algorithm for the case of constant capacities, which is of polynomial complexity when the number of items N is fixed. (This paper also deals with the case where multiple capacitated batches may be ordered in each period. Anily and Tzur 2006 develop an exponential search algorithm for the same problem.) It is for this reason that even the more advanced manufacturing resource planning systems (MRPII) start with the determination of systemwide order releases without consideration of capacity constraints, i.e., on the basis of the solution (for each stage or item) of the uncapacitated single-item dynamic lot-sizing model. It is only in the last phase of the planning process that the elimination of capacity conflicts is attempted by heuristic adaptations of the basic schedules.

Federgruen and Tzur (1994a) have demonstrated for single-item uncapacitated dynamic lot-sizing models that optimal or close-to-optimal initial decisions can be made by truncating the horizon after a relatively small number of periods. A forecast horizon is found in which at most three, and usually only two, orders are placed (the obligatory order in the first period included). It is reasonable to expect similarly short forecast horizons to continue to apply when multiple items are considered and in the presence of capacity constraints, as long as the utilization rate is not very close to one. See Federgruen and Tzur (1994a) for a discussion of how these forecast horizon results relate to capacitated models. This suggests that a close-to-optimal solution may be generated by partitioning or truncating the horizon.

We therefore develop and analyze a new class of so-called *progressive interval heuristics*. A progressive interval heuristic consists of J iterations. In iteration l, the problem is solved to optimality for period 1 to some period T_l, but all integer variables for periods 1 to $T_l - \tau$ (for some $\tau > 0$) and all continuous variables for periods 1 to some $t_l \leqslant T_{l-1}$ are fixed at their optimal values after iteration $l - 1$. When solving a given interval problem, we append, as boundary conditions, the necessary and sufficient conditions for a feasible extension to the remainder of the planning horizon. The horizons are chosen such that $0 = T_0 \leqslant T_1 \leqslant \cdots \leqslant T_J = T$ and $0 = t_1 \leqslant t_2 \leqslant \cdots \leqslant t_J$, while $\tau \geqslant T_l - T_{l-1}$, the number of periods by which the horizon in the lth iteration is expanded. The complexity of any mixed-integer programming method is largely determined by the number of (unrestricted) integer variables. Choosing the parameter τ sufficiently small therefore ensures that the complexity in each iteration grows only modestly. Thus, while the heuristic solves a sequence of progressively larger problem instances, exact solution methods remain viable with only modest increases in computational effort.

We pay special attention to two extreme subsets of this class of heuristics: (i) the *strict partitioning heuristics* (SP): here $t_l = T_{l-1}$ and $T_l - T_{l-1} = \tau$, with the possible exception of the last interval. The planning horizon is thus partitioned into nonoverlapping intervals and in the lth iteration, only the total cost pertaining to the newly appended τ-period interval are minimized, given the boundary conditions (in particular ending inventories) generated in the previous $(l - 1)$st iteration; (ii) the *expanding horizon heuristics* (EH): here $t_l = 0$ for all $l = 1, \ldots, J - 1$. A hybrid implementation would, e.g., set $t_l = [T_l - M]^+$ for some window M. The trade-offs are clear: (EH) [(SP)] provides, within the class of progressive interval heuristics, maximum (minimum) flexibility at the expense of maximum (minimum) incremental computational complexity in adjusting the solution from each iteration to the next.

When applied to the JS model, the (SP) heuristic can be implemented to be, simultaneously, asymptotically optimal as $T \to \infty$ and to run in $O(NT^2 \log\log T)$ time, provided some of the model parameters are uniformly bounded from above or from below. With the same choice of τ and the same interval choices, the (EH) heuristic continues to be asymptotically optimal and runs in $O(NT^3)$ time. Our numerical study reveals, however, that it generally results in significantly better solutions than the (SP) heuristic. Both heuristics can also be designed as polynomial time approximation schemes, i.e., to be of polynomial time complexity and to guarantee an ϵ-optimal solution for any $\epsilon > 0$. To our knowledge, these are the first heuristics for multi-item capacitated lot-sizing problems to possess these properties.

While the above theoretical results refer to the JS model, a comprehensive numerical study shows how in particular the (EH) heuristic can be effectively used for the general JIS model (with period- and item-dependent setup costs) as well. For the latter, it is possible to find the optimal solution for instances with up to 150–200 setup variables (e.g., when $N = 10$ and $T = 15$ or 20). For these problem sizes, the (EH) heuristic generates close-to-optimal solutions with an optimality gap of up to 2% across a large set of parameter combinations. (The (SP) heuristic, while significantly faster, often generates solutions with optimality gaps above 10%.)

While exact optimality gaps cannot be measured for larger problem instances, our theoretical results show that (at least for the JS model) optimality gaps can be expected to be even lower as T, the length of the planning horizon, increases. We systematically evaluate the performance of both the (SP) heuristic and the (EH) heuristic for problem instances with the number of items varying from 10 to 25 and the horizon length varying from 10 to 50 in the JIS model and up to 100 in the JS model. An earlier numerical study for the single-item problem in Federgruen and Tzur (1994b) shows that problems with up to 100 periods can be solved by a slight variant of the (SP) heuristic with an optimality gap of less than 7% and, on average, equal to 2%.

Summarizing, the main contributions of this paper are (i) the design of a new class of heuristics; (ii) the demonstration that, for the JS model, both the (SP) and (EH) heuristics can be designed to be of low polynomial complexity as well as asymptotically optimal; (iii) the proof that for finite T, both the (SP) and (EH) heuristics can be designed to be polynomial time approximation schemes; and (iv) the demonstration that a progressive interval heuristic generates close-to-optimal solutions with modest computational effort, even for large-scale problems.

While our theoretical and numerical analysis are based on the JS and JIS models, we believe that the effectiveness of the progressive interval heuristics bodes well for its use in general multiperiod production and inventory problems.

The remainder of this paper is organized as follows: Section 2 reviews the relevant literature. In §3, we introduce the JS model and its notation. In §4, we describe the new class of heuristics and develop worst-case bounds for their optimality gaps. In §5, we discuss how each interval problem, which arises in an iteration of the heuristic, can be solved effectively via a general-purpose mixed-integer programming method or a tailor-made branch-and-bound method. This allows us to identify implementations that are simultaneously asymptotically optimal as well as of very reasonable and polynomial complexity. Finally, §6 discusses extensions to the general JIS model as well as the numerical study.

2. Literature Review

In this section, we provide a brief review of the existing literature, beyond the papers mentioned in the introduction.

Chen et al. (1994) and Shaw and Wagelmans (1998) developed two relatively efficient pseudopolynomial solution methods for the general single-item model. Their extensions to the multi-item model result in dynamic programs with a state space of dimension N and larger, and are therefore entirely unusable except for the smallest possible number of items N. As mentioned, even for the single-item model, this paper's heuristics are, to our knowledge, the first to be asymptotically optimal and of polynomial complexity.

All other existing methods are based on heuristics, and none has provable bounds for the associated optimality gaps. These heuristics can be divided into simple constructive heuristics and mathematical programming-based heuristics. The constructive heuristics include "greedy methods" in which a specific sequence is proposed to assign the capacity of a given period to satisfy its or later demand, e.g., Eisenhut (1975), Lambrecht and Vander Eecken (1978), Dixon and Silver (1981), and Maes and Van Wassenhove (1986). Other constructive heuristics start with the solution of the uncapacitated model and search for a feasible production schedule by simple shifting routines, e.g., Van Nunen and Wessels (1978), Dogramaci et al. (1981), Nahmias (1989), and Karni and Roll (1982).

The mathematical programming-based heuristics employ linear programming, Lagrangean relaxation, cutting-plane methods, and column generation techniques. Examples include Baker and Dixon (1978), Eppen and Martin (1987), Pochet (1988), Leung et al. (1989), Martin (1987), and Trigeiro et al. (1989). We refer to Maes and van Wassenhove (1988), Salomon (1990), and Kuik et al. (1994) for detailed surveys of these methods up to 1994.

State-of-the-art solution methods include, in addition to the bc-prod system mentioned in §1 (Belvaux and Wolsey

2000, 2001), those of Stadtler (2003) and Suerie and Stadtler (2003). Interestingly, these methods all apply variants of the (EH) heuristic: In the "fix-and-relax" heuristic, each consecutive problem instance expands the horizon of the previous instance by appending the same number (τ) of periods to its tail. The "internally rolling schedule heuristics" in Stadtler (2003) and Suerie and Stadtler (2003) use constant interval increments $\leqslant \tau$. In each problem instance, instead of imposing boundary conditions that are necessary and sufficient for a feasible extension until the end of the full planning horizon, the authors include the periods beyond the end of the current interval, however, with all binary variables in these periods treated either as continuous variables (bc-prod) or set equal to one (Stadtler 2003, Suerie and Stadtler 2003). The heuristics in the latter two papers substitute all cost parameters for the after-the-interval periods by zero, with the possible exception of variable overtime cost rates, in case the capacity constraints may be violated by scheduling overtime. (Additional heuristic changes are applied to an interval's last set of periods.)

Federguen and Tzur (1994c) describe an effective heuristic for the so-called joint replenishment problem (JRP), which is similar to the (SP) heuristic. (This heuristic can be designed to be asymptotically optimal and of polynomial complexity, under specific parameter conditions.) The (JRP) model is the special case of the JIS model, which arises when no capacity constraints prevail. Federgruen and Tzur (1999) describe a general framework for a variant of the (SP) heuristic, with applications to other types of lot-sizing problems.

3. The Multi-Item Model with Joint Setup Cost JS

In this section, we discuss our basic model JS with joint setup costs only. We use the index $i \in \{1, \ldots, N\}$ to distinguish between items and the index $t \in \{1, \ldots, T\}$ to distinguish between periods. For $i = 1, \ldots, N$ and $t = 1, \ldots, T$, we specify the following parameters:

d_{it} = demand for item i in period t ($d_{it} \geqslant 0$);
D_t = aggregate demand in period $t = \sum_{i=1}^{N} d_{it}$;
c_{it} = variable per-unit order cost for item i in period t;
h_{it} = cost of carrying a unit of inventory of item i at the end of period t;
K_t = setup cost incurred when an order is placed in period t; and
C_t = order capacity, i.e., the maximum number of units that can be ordered in period t.

Without loss of generality, we define the units of the items such that ordering one unit of an item consumes one unit of capacity. We define the following decision variables:

x_{it} = order size for item i in period t, $i = 1, \ldots, N$, $t = 1, \ldots, T$;

$$Y_t = \begin{cases} 1 & \text{if } \sum_{i=1}^{N} x_{it} > 0, \\ 0 & \text{otherwise,} \end{cases} \quad t = 1, \ldots, T;$$

I_{it} = ending inventory of item i in period t, $i = 1, \ldots, N$, $t = 1, \ldots, T$.

Let I_t^0 = the minimum aggregate inventory at the end of period t, such that a feasible production/inventory plan exists for periods $t+1, \ldots, T$. These minimum stock levels are easily computed from the following recursion, which can be verified by induction:

$$I_t^0 = (D_{t+1} - C_{t+1} + I_{t+1}^0)^+, \quad t = 1, 2, \ldots, T-1, \qquad \text{with } I_T^0 = 0. \quad (1)$$

The multi-item model can thus be formulated as follows:

$$(\text{P}) \quad z^* = \min \left\{ \sum_{t=1}^{T} \left[K_t Y_t + \sum_{i=1}^{N} (c_{it} x_{it} + h_{it} I_{it}) \right] \right\} \quad (2)$$

$$\text{s.t.} \quad I_{it} = I_{i(t-1)} + x_{it} - d_{it}, \quad i = 1, \ldots, N, \; t = 1, \ldots, T, \quad (3)$$

$$\sum_{i=1}^{N} x_{it} \leqslant C_t Y_t, \quad t = 1, \ldots, T, \quad (4)$$

$$\sum_{i=1}^{N} I_{it} \geqslant I_t^0, \quad t = 1, \ldots, T, \quad (5)$$

$$x_{it} \geqslant 0; \quad I_{it} \geqslant 0; \quad Y_t \in \{0, 1\}. \quad (6)$$

The above formulation is often referred to as the network formulation. The plant location formulation is an alternative that disaggregates the production quantities $\{x_{it}\}$ into $\{x_{ist}\}$ with x_{ist} = the amount of item i ordered in period s to satisfy demand in period t.

4. Progressive Interval Heuristics: Worst-Case Bounds for Optimality Gaps

A progressive interval heuristic solves a sequence of J problem instances. The first instance considers the capacitated lot-sizing problem that arises when restricting oneself to the first T_1 periods, i.e., it solves (P) with T replaced by T_1. In each of the subsequent instances, a given number of periods $\leqslant \tau$ is appended to the tail of the previous planning horizon. In the hth iteration, a lot-sizing problem ($\widetilde{\text{JS}}_h$) is solved on the complete interval $\{1, \ldots, T_h\}$, albeit that all Y variables of periods $1, \ldots, T_h - \tau$ are fixed at their optimal values in the $(h-1)$st iteration, i.e., when solving ($\widetilde{\text{JS}}_{h-1}$). Recall that $T_h - T_{h-1} \leqslant \tau$, i.e., $T_h - \tau \leqslant T_{h-1}$. Thus, the number of unrestricted binary variables in each iteration remains constant, i.e., equal to τ. Moreover, the aggregate ending inventory in period T_h is constrained from below by the I^0-value.

Different progressive interval heuristics give varying amounts of flexibility to the continuous variables in each of the J problem instances. As mentioned, we focus in particular on two extremes: under the strict partitioning heuristics

(SP), all interval increments $(T_l - T_{l-1}) = \tau$, with the possible exception of the last interval. Also, among the continuous variables, only those pertaining to the last τ periods of the current planning horizon are allowed to be chosen freely (i.e., $t_l = T_l - \tau = T_{l-1}$) without any restrictions beyond those implied by the constraints of (P); all other continuous variables are fixed at their optimal value in the previous problem instance.

Under the (EH) heuristics, *all* of the continuous variables are allowed to be varied fully (subject, of course, to constraints (3)–(6)), i.e., $t_l = 0$; moreover, this class allows for $T_l - T_{l-1} < \tau$, $l = 1, \ldots, J-1$. If the step sizes $(T_l - T_{l-1}) < \tau$, even many of the setup decisions determined in one iteration of the algorithm may be revisited in subsequent iterations on the basis of additional demand, cost, and capacity information pertaining to additional periods. (We have observed that it is often effective to append a single period as one progresses from one interval to the next, i.e., $T_l - T_{l-1} = 1$.)

Various intermediate implementations may be envisioned; for example, a (moving) window of $M > \tau$ periods may be used such that the continuous variables in (up to) the last M periods are unrestricted, as opposed to the last τ (SP) or *all* periods under (EH). Federgruen and Tzur (1999) consider a slight variant of (SP) under which the size and the composition of the last-(or several of the last) order periods in the previously solved iteration may be varied, along with the production quantities of the newly appended periods.

Let z^{SP} and z^{EH} denote the cost of the solutions found by the (SP) and the (EH) heuristics for a given choice of $\{T_l, t_l, \tau\}$. We now derive worst-case bounds for their optimality gaps, under mild conditions for the cost, demand, and capacity parameters. We first derive a lower bound for z^* as an explicit function of T. It is quite simple to obtain a lower bound when assuming that *all* periods' demands are uniformly bounded away from zero; however, to allow for sporadic demands, we derive an alternative bound, merely assuming that the cumulative demand over a large enough time interval is uniformly bounded away from zero. Its proof, while similar to that in Federgruen and Tzur (1994c), requires major adjustments to reflect the capacity limits.

Theorem 1. *Assume that there exists a positive integer* $\theta \geqslant 1$, *and for all* $i = 1, \ldots, N$, *positive constants* d_{i*} *such that*

$$d_{it} + \cdots + d_{i(t+\theta-1)} \geqslant \theta d_{i*}, \qquad i = 1, \ldots, N,\ t = 1, \ldots, T-\theta+1, \tag{7}$$

$$\sum_{t=1}^{T} d_{it} \geqslant T d_{i*}, \quad i = 1, \ldots, N. \tag{8}$$

In addition, assume that there exist constants K_* *and* C^*, *and for each* $i = 1, \ldots, N$, *constants* h_{i*} *and* c_{i*} *such that* $K_t \geqslant K_*, C_t \leqslant C^*$, $h_{it} \geqslant h_{i*}$, *and* $c_{it} \geqslant c_{i*}$ *for all* $t = 1, \ldots, T$. *Let* $d_* = \sum_{i=1}^N d_{i*}$, $\kappa = \sum_{i=1}^N c_{i*} d_{i*}$, $H_* = (1/2)\sum_{i=1}^N h_{i*} d_{i*}$. *Then,* $z^* \geqslant \gamma T$, *where*

$$\gamma \overset{\text{def}}{=} \kappa + \begin{cases} \dfrac{K_*}{2\theta} \quad \text{if } \sqrt{\dfrac{H_*}{\widehat{K}}} \geqslant \dfrac{1}{2\theta}, \\[2ex] \left(2\sqrt{(K_* + 2H_*\theta^2)H_*} - 3H_*\theta\right) \\ \qquad \text{if } \dfrac{d_*}{C^*} < \sqrt{\dfrac{H_*}{\widehat{K}}} < \dfrac{1}{2\theta}, \\[2ex] \left(\dfrac{(K_* + 2H_*\theta^2)d_*}{C^*} + \dfrac{H_* C^*}{d_*} - 3H_*\theta\right) \\ \qquad \text{if } \sqrt{\dfrac{H_*}{\widehat{K}}} \leqslant \dfrac{d_*}{C^*} < \dfrac{1}{2\theta}. \end{cases} \tag{9}$$

Proof. We obtain a lower bound by replacing all fixed-order costs by K_*, all capacities by C^*, and for each item $i = 1, \ldots, N$, all variable-order cost rates by c_{i*}, and holding-cost rates by h_{i*}. We refer to the resulting problem as the *transformed problem*. Consider a solution in which $m \geqslant 1$ orders are placed. For $l = 1, \ldots, m$, let n_l denote the number of periods in the lth order cycle, i.e., the interval that contains the lth order period and all subsequent periods prior to the next order interval (if any). (The mth interval terminates with period T.)

We first derive a lower bound for the total holding costs incurred in a single order cycle of n periods in the transformed problem. Note that zero-inventory ordering may fail to be optimal in the capacitated model, i.e., the starting inventory in the first period may be positive for some or all items. However, the holding cost in the order cycle is clearly bounded from below by assuming that the starting inventory equals zero.

Renumber the periods in this cycle as $1, \ldots, n$ and let $n = \psi\theta + \tau$ with $0 \leqslant \tau < \theta$, i.e., $\psi = \lfloor n/\theta \rfloor$. Fix $i = 1, \ldots, N$. Observe by our assumption that in each of the intervals $[(j-1)\theta + \tau + 1, j\theta + \tau]$ for $j = 1, \ldots, \psi$, at least θd_{i*} units are demanded for item i. Being ordered in or after period 1, the lowest holding costs for these demands arise when θd_{i*} units are demanded in period $(j-1)\theta + \tau + 1$ (i.e., in the first period of this interval) and none in the remaining periods of the interval $[(j-1)\theta + \tau + 1, j\theta + \tau]$. It follows that the holding costs in a single order cycle of n periods are bounded from below by

$$\begin{aligned} &\sum_i h_{i*}\theta d_{i*} \sum_{j=0}^{\psi-1} (\tau + j\theta) \\ &\quad = \sum_i h_{i*}\theta d_{i*}\left[\psi\tau + \frac{1}{2}\theta\psi(\psi-1)\right] \\ &\quad = \sum_i h_{i*}\theta d_{i*}\left[\left\lfloor\frac{n}{\theta}\right\rfloor\tau + \frac{1}{2}\theta\left\lfloor\frac{n}{\theta}\right\rfloor\left(\left\lfloor\frac{n}{\theta}\right\rfloor - 1\right)\right] \\ &\quad \geqslant \sum_i \frac{1}{2} h_{i*}\theta^2 d_{i*}\left[\left(\frac{n}{\theta} - 1\right)^+ \left(\frac{n}{\theta} - 2\right)^+\right] = g(n), \end{aligned}$$

where $g(x) \overset{\text{def}}{=} H_*\theta^2((x/\theta) - 1)^+((x/\theta) - 2)^+$ is convex.

This implies the following lower bound for the total cost over the complete horizon:

$$\begin{aligned} z^* &\geqslant \kappa T + \min_m \left\{ K_* m + \min_{n_l}\left[\sum_{l=1}^{m} g(n_l) : \sum_{l=1}^{m} n_l = T\right] \middle| m \geqslant \frac{Td_*}{C^*}\right\} \\ &= \kappa T + \min_m \left\{ K_* m + (H_*\theta^2) m \left(\frac{T}{m\theta}-1\right)^+ \left(\frac{T}{m\theta}-2\right)^+ \middle| \frac{Td^*}{C^*} \right. \\ &\qquad\qquad \left. \leqslant m \leqslant \max\left(\frac{Td^*}{C^*}, \frac{T}{2\theta}\right)\right\}. \end{aligned} \tag{10}$$

The lower bound for m may be imposed because when $mC^* < Td_*$, it is infeasible to satisfy all demand. The equality in (10) follows because, by the convexity of $g(\cdot)$, equal values $n_l = T/m$, $l = 1, \ldots, m$, achieve the minimum to its left. (The upper bound for m may be imposed because the minimand to the right of (10) is increasing for $m > T/2\theta$.) Equation (9) follows from (10), noting that for $m \leqslant T/2\theta$, the $(\cdot)^+$ operators may be ignored. □

We now derive an upper bound for the optimality gap of the (SP) and (EH) heuristics. The bound is established under the parameter conditions of the lower bound Theorem 1, a uniform lower (upper) bound for the capacities (holding-cost rates) and a condition that specifies that a uniform slack capacity exists over any cycle of θ periods, i.e.,

(S) there exists a constant $\sigma > 0$ and an integer ζ such that

$$\sum_{r=t+1}^{t+\zeta} C_r \geqslant \sum_{r=t+1}^{t+\zeta} D_r + \sigma \quad \text{for all } t = 0, \ldots, T-\zeta. \tag{11}$$

We first need the following lemma, which shows that under condition (S) a uniform upper bound prevails for all minimum reserve stocks $\{I_t^0\}$:

Lemma 1. *Let condition* (*S*) *hold and assume that a constant C^* exists such that $C_t \leqslant C^*$. Then,*

$$I_t^0 \leqslant U \overset{\text{def}}{=} \zeta C^* - \sigma, \quad t = 1, \ldots, T. \tag{12}$$

Proof. By repeated substitutions in (1), we get for all $t = 1, \ldots, T$,

$$\begin{aligned} I_t^0 &= \max_{t+1\leqslant s\leqslant T}\left[\sum_{r=t+1}^{s}(D_r - C_r)\right]^+ \\ &= \max_{t+1\leqslant s\leqslant \min(T,\, t+\zeta-1)}\left[\sum_{r=t+1}^{s}(D_r - C_r)\right]^+ \\ &\leqslant \max_{t+1\leqslant s\leqslant \min(T,\, t+\zeta-1)} \sum_{r=t+1}^{s} D_r \leqslant \sum_{t=t+1}^{\min(T,\, t+\zeta-1)} D_r, \end{aligned}$$

where, by (S), the second equality follows from

$$\sum_{r=t+1}^{s}(D_r - C_r) \geqslant \sum_{r=t+1}^{s-\zeta}(D_r - C_r) \quad \text{for } s \geqslant t+\zeta.$$

Thus, I_t^0 can be bounded by a sum of ζ consecutive aggregate demands, hence, by a sum of ζ consecutive capacity values minus σ, given (11). This proves (12). □

Theorem 2. *Let* (S) *hold. Assume that there exists an integer $\theta \geqslant 1$, and for each $i = 1, \ldots, N$, a constant d_{i*} such that*

$$(d_{it} + \cdots + d_{i,\,t+\theta-1}) \geqslant \theta d_{i*}, \quad t = 1, \ldots, T-\theta+1, \tag{13}$$

$$\sum_{t=1}^{T} d_{it} \geqslant Td_{i*}. \tag{14}$$

In addition, assume that there exist constants K^*, K_*, C^*, and C_*, and for each $i = 1, \ldots, N$, constants h_{i*}, h_i^*, c_{i*}, and c_i^* such that for all $t \geqslant 1$, $K_* \leqslant K_t \leqslant K^*$, $C_* \leqslant C_t \leqslant C^*$, $h_{i*} \leqslant h_{it} \leqslant h_i^*$, and $c_{i*} \leqslant c_{it} \leqslant c_i^*$. Let $\Delta c^* = \max_i[c_i^* - c_{i*}]$, $\eta = (K^*/C_*) + \Delta c^*$, $h_* = \min_i h_{i*}$, $\Delta h^* = \max[h_i^* - h_{i*}]$, and $D_* = \sum_{i=1}^{N} d_{i*}$. Let γ be defined as in (9) *and*

$$\rho_1 = K^* + C^*\left(\left\lfloor\frac{\eta}{h_*}\right\rfloor\eta - \frac{1}{2}\left\lfloor\frac{\eta}{h_*}\right\rfloor\left(\left\lfloor\frac{\eta}{h_*}\right\rfloor + 1\right)h_*\right), \tag{15}$$

$$\begin{aligned} \rho_2 = U\Bigg[(\Delta c^* + K^*) + \left(\left\lfloor\frac{U}{\sigma}\right\rfloor + 1\right)\zeta\Delta h^* \\ + \left(\frac{\Delta c^* + K^*}{h_*}\right)\Delta h^*\Bigg], \end{aligned} \tag{16}$$

$$\rho = \rho_1 + \rho_2. \tag{17}$$

Then,

$$\text{(a)}\ \ \frac{z^{\text{SP}} - z^*}{z^*} \leqslant \frac{(J-1)\rho}{\gamma T}, \qquad \text{(b)}\ \ \frac{z^{\text{EH}} - z^*}{z^*} \leqslant \frac{(J-1)\rho}{\gamma T}.$$

Proof. (a) We show that an optimal solution of the complete problem can be transformed, in two phases, into one that is achievable by the (SP) heuristic, adding at most $(J-1)\rho$ to the total cost. In Phase I, the optimal solution is transformed into one with all intervals' ending aggregate inventory equal to their minimum I^0-level. In Phase II, the composition of the reserve stock at the end of each of the intervals is made identical to that of the solution of the (SP) heuristic.

To describe the transformation in Phase I, renumber the periods in the first l intervals from $1, \ldots, T_l$, starting with T_l and going backwards—i.e., period t is now renumbered as $T_l - t + 1$, $t = 1, \ldots, T_l$. With this numbering, period t occurs t periods before the end of the lth interval.

In the optimal solution, let Q_{ir} denote the number of units of item i ordered in period r to satisfy demands in some future period in the $(l+1)$st or later intervals $(i = 1, \ldots, N,\ r = 1, \ldots, T_l)$. Also, let $Q_r = \sum_i Q_{ir}$. The starting aggregate inventory of the $(l+1)$st interval is $= \sum_{r=1}^{T_l} Q_r > I_1^0$. Because a feasible solution exists for (JS_{l+1}) with a starting inventory of I_1^0 only, it is feasible to postpone the orders for $(\sum_{r=1}^{T_l} Q_r) - I_1^0$ units to periods that belong to the $(l+1)$st interval itself. The transfer of these order quantities requires at most $\lceil((\sum_{r=1}^{T_l} Q_r) - I_1^0)/C_*\rceil$ additional setups in the $(l+1)$st interval, and therefore at most $\lceil((\sum_{r=1}^{T_l} Q_r) - I_1^0)/C_*\rceil K^* \leqslant \lceil(\sum_{r=1}^{T_l} Q_r)/C_*\rceil K^* \leqslant K^* + (K^*/C_*)\sum_{r=1}^{T_l} Q_r$ in additional setup costs. An upper bound for the total additional costs due to the transfer of these order quantities is therefore given by

$$\max\left\{\sum_{i=1}^{N}\sum_{r=1}^{T_l}\left[c_i^* - c_{ir} - \sum_{s=1}^{r} h_{i,s}\right]Q_{i,r} + \frac{K^*}{C_*}\sum_{r=1}^{T_l}\sum_{i=1}^{N} Q_{ir} + K^* \,\middle|\, 0 \leqslant \sum_{i=1}^{N} Q_{ir} \leqslant C_r \;\forall\, r\right\}.$$

This linear progam decomposes into T_l single-constraint problems. Each is straightforwardly solved in closed form: For each $r = 1, \ldots, T_l$, set $Q_{ir} = C_r$ for any item i whose objective function coefficient is largest, unless all $\{Q_{ir}: i = 1, \ldots, N\}$ variables have negative coefficients, in which case it is optimal to set $Q_{ir} = 0$ for all $i = 1, \ldots, N$. This results in the upper bound $K^* + \sum_{r=1}^{T_l} \max_i [(K^*/C_*) + c_i^* - (c_{i,r} + h_{i,r} + h_{i,r-1} + h_{i,1})]^+ C_r \leqslant K^* + \sum_{r=1}^{T_l} [\eta - rh_*]^+ C_r \leqslant K^* + C^* \sum_{r=1}^{\Lambda} [\eta - rh_*] \leqslant K^* + C^* \{\Lambda\eta - (1/2) \cdot \Lambda(\Lambda+1)h_*\} = \rho_1$, where $\Lambda = \lfloor \eta/h_* \rfloor$ is an upper bound on the number of periods in which inventory may be held prior to the lth interval for use during the lth or later intervals. (The first equality follows from the fact that the $(\Lambda+1)$st until the T_lth term in the sum to its left vanishes.) Apply the transfer process sequentially to the intervals $l = J-1, J-2, \ldots, 1$, to end up with a solution in which all intervals' ending aggregate inventory equals the minimum I^0-level and whose cost exceeds z^* by at most $(J-1)\rho_1$.

Let L_l denote the longest shelf life of any unit in stock at the end of the lth interval $l = 1, \ldots, J-1$. In Phase II, we transform the Phase I solution by changing the item identity of at most $I_{T_l}^0$ units in stock at the end of period T_l, without any additional changes in the order and inventory plan. This maintains feasibility, leaves total setup costs unaltered, and adds at most

$$\sum_{l=1}^{J-1} I_{T_l}^0 (\Delta c^* + L_l \Delta h^*) \tag{18}$$

variable-order and holding costs. In view of Lemma 1, to show that the summand in (18) is bounded by ρ_2, it suffices to show that $L_l \leqslant (\lfloor U/\sigma \rfloor + 1)\zeta + (\Delta c^* + K^*)/h^* \overset{\text{def}}{=} \bar{L}$.

Assume first that at least one of the periods $t^* \in \{T_l - (\lfloor U/\sigma \rfloor + 1)\zeta + 1, \ldots, T_l\}$ has slack capacity (in the Phase I solution). In this case, if one of the $I_{T_l}^0$ units in the reserve stock has a shelf life of more than $\bar{L}$ periods, the ordering of this unit can be postponed until t^*, thereby reducing inventory costs by at least $h_*(\bar{L} - (\lfloor U/\sigma \rfloor + 1)\zeta) = h_*((\Delta c^* + K^*)/h_*) = \Delta c^* + K^*$, offsetting any increase in the variable-ordering cost (and possibly one setup cost), due to the postponement. Thus, if any of the $I_{T_l}^0$ units has a shelf life larger than $\bar{L}$, a full-capacity order is placed in each period of the interval $[T_l - (\lfloor U/\sigma \rfloor + 1) \cdot \zeta + 1, \ldots, T_l]$, resulting in an ending inventory of at least $\sum_{t=T_l-(\lfloor U/\sigma \rfloor+1)\zeta+1}^{T_l} (C_t - D_t) \geqslant (\lfloor U/\sigma \rfloor + 1)\sigma > U$ units, which contradicts Lemma 1.

(b) Let $I^{(l)}$ denote the N-vector of ending inventories at the end of the lth interval, as determined in the lth iteration of the (EH) heuristic, $l = 1, \ldots, J$, and let $\{Y_t^{\text{EH}}: t = 1, \ldots, T\}$ be the Y-vector chosen by this heuristic. Transform the optimal solution into a solution $\pi^{(\text{II})}$ with cost value $z^{(\text{II})}$ via Phase I and Phase II transformations, as in part (a), except that in Phase II the lth interval's vector of ending inventories is now matched to $I^{(l)}$. With $T_{-1} = T_0 = 0$, let $\pi^{(l)}$ be an optimal solution of the mixed-integer program (P^l), where $l = 0, \ldots, J$:

$$(\text{P}^l)\colon \quad z^{(l)} = \min(2) \tag{19}$$

s.t.

$$(3)\text{–}(7), \tag{20}$$

$$I_{iT_h} = I_i^{(h)}, \quad i = 1, \ldots, N, \qquad h = \max(l-1, 1),\ \max(l, 1),\ l+1, \ldots, J, \tag{21}$$

$$Y_t = Y_t^{\text{EH}}, \quad t = 1, \ldots, T_{l-1}. \tag{22}$$

(P^{l^*+1}) is obtained from (P^{l^*}) by simultaneously adding the constraints $Y_t = Y_t^{\text{EH}}$, $t = T_{l^*-1} + 1, \ldots, T_{l^*}$ and eliminating the constraints $I_{iT_{l^*-1}} = I_i^{(l^*-1)}$, $i = 1, \ldots, N$. Because $\pi^{(l^*)}$ satisfies (21) and (22) for $l = l^*$—i.e., because it maintains the same ending inventories at the end of the l^*th interval as the EH heuristic does at the end of the l^*th iteration, and because it is restricted to the same order periods in the first $(l^* - 1)$ intervals as the (EH) heuristic is in its l^*th iteration—it follows that both $\pi^{(l^*)}$ and the solution obtained by the (EH) heuristic in its l^*th iteration, minimize total costs over the first T_{l^*} periods subject to constraints (21)–(22) with $l = l^*$. This implies that $\pi^{(l^*)}$ can be chosen such that $Y_t = Y_t^{\text{EH}}$, $t = T_{l^*-1} + 1, \ldots, T_{l^*}$, and hence $Y_t = Y_t^{\text{EH}}$ for all $t = 1, \ldots, T_{l^*}$. Thus, $\pi^{(l^*)}$ is a feasible solution of (P^{l^*+1}) so that

$$z^{\text{EH}} = z^{(J)} \leqslant z^{(J-1)} \leqslant \cdots \leqslant z^{(0)} \leqslant z^{(\text{II})} \leqslant z^* + (J-1)\rho, \tag{23}$$

where the equality follows from the (EH) solution optimizing $\text{P}^{(J)}$, the last inequality from part (a), and the one before that from $\pi^{(\text{II})}$ being a feasible solution of $\text{P}^{(0)}$. □

Remark. The proof of Theorem 2 reveals that a tighter bound, with ρ replaced by a smaller value, may be computed in any given instance, once the number of intervals and their lengths have been specified.

5. Solution Methods for a Single-Interval Problem: Polynomial and Asymptotically Optimal Heuristics

We now discuss how a single-interval problem in an iteration of the progressive interval heuristic can be solved effectively. We have found that the general purpose branch-and-bound method embedded in CPLEX is very effective to solve JS problems; see §6 for details. Alternatively, several tailor-made branch-and-bound methods can be used.

Below, we discuss three such methods. Two of them have the distinct advantage over the CPLEX-based algorithm that their complexity, for the (SP) heuristic, is of the order $O(2^{\tau}P(\tau))$ with $P(\cdot)$ a polynomial in τ. (The complexity is $O(2^{\tau}P(T))$ for the (EH) heuristic.) Theorem 2 shows that the two heuristics can be designed to be asymptotically optimal, e.g., by choosing every (except possibly the last) interval increment $T_l - T_{l-1} = \tau$, $l = 1, \ldots, J-1$, with

$$\tau = \lceil \alpha \log T \rceil \quad \text{for some } \alpha > 0, \tag{24}$$

or, more generally, by choosing $\tau = o(T)$ as $T \to \infty$. (The last increment $T_l - T_{l-1} = T - \lfloor (T/\tau)\tau \rfloor$.) Thus, by choosing τ as in (24), we obtain an algorithm that is simultaneously asymptotically optimal and of polynomial complexity.

Our three branch-and-bound methods are based on three bounds for the value of z^*:

$z^{\mathrm{LB}_1} =$ minimum cost value in the uncapacitated model, i.e., ignoring constraints (4),

$z^{\mathrm{LB}_2} = \max_{\lambda \geqslant 0} z(\lambda)$, where
$z(\lambda) = \min\{\sum_{t=1}^{T}(K_tY_t + \sum_{i=1}^{N}(c_{it}x_{it} + h_{it}I_i) + \lambda_t[C_tY_t - \sum_i x_{it}])$ s.t. (3), (5), (6)$\}$.

In other words, z^{LB_2} is the value of the Lagrangean dual associated with the relaxation of the capacity constraints (4). Clearly, $z^{\mathrm{LB}_2} \geqslant z(0) = z^{\mathrm{LB}_1}$.

$z^{\mathrm{LB}_3} = z^{\mathrm{LB}_{\mathrm{var}}} + z^{\mathrm{LB}_{\mathrm{fix}}}$, where

$z^{\mathrm{LB}_{\mathrm{var}}} =$ minimum value of the variable costs, i.e., minimum cost value when all setup costs are reduced to zero, and

$z^{\mathrm{LB}_{\mathrm{fix}}} =$ minimum value of the fixed (setup) costs required to satisfy all demands when in each period t the best observed, and yet unused, setup cost and capacity value can be used (instead of only K_t and C_t being available).

Therefore, $z^{\mathrm{LB}_{\mathrm{fix}}}$ is a lower bound on the minimum value of the fixed costs, i.e.,

$$\begin{aligned} z^* &\geqslant \min\left\{\sum_{t=1}^{T}\left[\sum_{i=1}^{N} c_{it}x_{it} + \sum_{i=1}^{N} h_{it}I_{it}\right] \text{ s.t. (3)–(6)}\right\} \\ &\quad + \min\left\{\sum_{t=1}^{T} K_tY_t \text{ s.t. (3)–(6)}\right\} \\ &= z^{\mathrm{LB}_{\mathrm{var}}} + z^{\mathrm{LB}_{\mathrm{fix}}} = z^{\mathrm{LB}_3}. \end{aligned} \tag{25}$$

In the single-item case ($N = 1$), z^{LB_1} can clearly be evaluated via any of the solution methods for the single uncapacitated model. (This can be done in $O(T \log T)$ time; see the introduction.) In the multi-item case, evaluation of z^{LB_1} reduces to the solution of the joint replenishment problem (JRP) without item-specific setup costs. In the important special case where no speculative motives for carrying inventory prevail, the complexity of this method is easily verified to be $O(NT^2)$; see Federgruen and Tzur (1994c). For general variable holding and order costs, any of the known lower bounds for the JRP can be invoked, e.g., the bound in Federgruen and Tzur (1994c), which requires $O((N + K^*)T \log T)$ time where $K^* = \max_t K_t$.

To evaluate z^{LB_2}, the above methods need to be embedded in an unconstrained optimization technique that searches for the maximizing vector λ.

z^{LB_3} is the sum of two components: $z^{\mathrm{LB}_{\mathrm{var}}}$ is the minimum cost network flow in a network of special structure. Ahuja and Hochbaum's (2004, §6.3) algorithm solves this problem in $O(NT \log T)$ time. To compute $z^{\mathrm{LB}_{\mathrm{fix}}}$, observe that it is optimal to sequentially postpone setups until the last feasible period because in any given period, any prior (unused) capacity and setup cost value may be chosen. Thus, assume that the first j setup periods $t(1), t(2), \ldots, t(j)$ have been determined, together with their adopted capacities and setup cost values; the next setup period $t(j+1)$ (if any) is then obtained as the first period t after $t(j)$ for which $\sum_{s=1}^{t} D_s$ is in excess of the sum of the adopted capacities for periods $t(1), \ldots, t(j)$; it is then optimal to assign to this setup period the best observed, and yet unused, setup cost and capacity value. This sequence of setup periods (and associated setup costs and capacity values) can be determined in $O(T \log T)$ time by maintaining two ordered lists of unused capacity and setup cost parameters. Thus, z^{LB_3} can be computed in $O(NT \log T)$ time.

5.1. Branch-and-Bound Methods

Our branch-and-bound (b&b) algorithm bears the following similarities to that in Federgruen and Tzur (1994c): (1) it implicitly enumerates all possible subsets of the τ undetermined order periods; (2) it characterizes each node of the b&b tree by a partition of the periods into sets S^+, S^-, and S^0, with S^+ the set of periods in which one is committed to place an order, S^- the set in which no order is allowed, and S^0 the set of periods where no decision is fixed yet; (3) the root of the tree has all τ periods in the set S^0 and every nonterminal node has two successor nodes, one with an additional period shifted from S^0 to S^+ and one with the same period shifted to S^-. (This period is selected according to a specific branching rule.) At any of the leaf nodes, for a given set of order periods, the problem reduces to a polynomially solvable network problem.

Compared to Federgruen and Tzur (1994c), a different lower bound is used to evaluate each node of the b&b tree. For all $r = 1, 2, 3$, and a given node characterized by S^+, S^-, S^0 let $Z^{\mathrm{LB}_r} = \sum_{i \in S^+} K_i +$ the value of z^{LB_r} when the setup cost for periods $i \in S^+$ (S^-) is changed to zero (∞) and the capacity for periods $i \in S^-$ is changed to zero. Each of the values Z^{LB_1}, Z^{LB_2}, and Z^{LB_3} can be used as a lower bound for any node in the tree; Z^{LB_3} gives the optimal solution value for nodes at the bottom of the tree, where $S^0 = \varnothing$.

We now conclude that both the (SP) and (EH) heuristics can be implemented as an asymptotically optimal and polynomially bounded heuristic, e.g., if all intervals are chosen as in (24).

Corollary 1. *Consider the* (SP) *heuristic with interval lengths specified by* (24) *and with each interval problem solved by the above branch-and-bound procedure.*

(a) *In the general multi-item case, the heuristic has complexity* $O(NT^2 \log\log T)$ *if each node is evaluated by the value* Z^{LB_3} *and* $O((N+K^*)T^2 \log\log T)$ *if evaluated by* Z^{LB_1}.

(b) *In the multi-item case without speculative motives, the heuristic has complexity* $O(NT^2 \log T)$ *if each node in the branch-and-bound tree is evaluated by* Z^{LB_1}.

(c) *In the single-item case* $(N=1)$, *the heuristic has complexity* $O(T^2 \log\log T)$ *if each node in the branch-and-bound tree is evaluated by* Z^{LB_1} *or* Z^{LB_3}.

(d) *Assume that the parameter conditions of Theorem* 2 *are satisfied. The heuristic is asymptotically optimal as T increases to infinity*; *the convergence of the optimality gap to zero is uniform in N.*

Proof. Parts (a)–(c): The (SP) heuristic requires, to compute its solution for any given interval, at most $2^{\tau-1}$ exact evaluations, one for each leaf of the branch-and-bound tree, and $2^{\tau-1}$ lower-bound evaluations of the other nodes of the tree. Exact evaluation of a leaf takes $O(N\tau\log\tau)$ time, as shown when discussing $z^{\mathrm{LB}_{\mathrm{var}}}$. Also, $\tau = O(\log T)$ and $J = O(T/\log T)$. The complexity bounds in parts (a)–(c) thus follow from those associated with a single evaluation of Z^{LB_1} or Z^{LB_3} in the nonleaf nodes of the branch-and-bound tree, i.e., $O(N\tau\log\tau)$, $O((N+K^*)\tau\log\tau)$, $O(N\tau^2)$, and $O(\tau\log\tau)$, respectively. Part (d) follows from the discussion at the start of §5. □

Thus, the (SP) heuristic can be designed to be asymptotically optimal with a complexity that grows only somewhat faster than quadratically in T, and linearly in the number of items N. The (EH) heuristic has larger complexity. For example, when implemented with interval increments of size τ and τ given by (24), its complexity is $O(NT^3)$ when each interval problem is solved by the above branch-and-bound procedure based on the lower bound Z^{LB_3}. On the other hand, the (EH) heuristic tends to generate significantly superior solutions, as we shall demonstrate in the next section.

The heuristics can also be designed as polynomial approximation schemes.

Corollary 2. *Assume that the parameter conditions of Theorem* 2 *are satisfied. For any given* $\epsilon \geqslant 0$, *choose* $\tau = \min\{T, \rho/\epsilon\gamma\}$ *and all interval increments* $T_l - T_{l-1} = \tau$ (*with the possible exception of the last interval increment, which is of length* $T - \lfloor T/\tau \rfloor \tau$). *Assume that each interval problem is solved by the above branch-and-bound procedure, with each node evaluated by* z^{LB_3}. *The* (SP) *and* (EH) *heuristics result in an* ϵ-*optimal solution with a complexity bound that is* $O(NT)$ *and* $O(NT^2 \log T)$, *respectively.*

Proof. The optimality gap result is obvious if $\tau = T$. Otherwise, by Theorem 2, for PI = SP and PI = EH,

$$\frac{z^{PI} - z^*}{z^*} \leqslant \frac{(J-1)\rho}{T\gamma} \leqslant \frac{(\lceil T/\tau \rceil - 1)}{T}\frac{\rho}{\gamma} \leqslant \frac{(T/\tau)\rho}{\gamma T} = \epsilon.$$

The complexity counts are immediate from the proof of Corollary 1. □

6. The General JIS Model and Numerical Results

In this section, we consider a generalization in which the fixed setup cost associated with an order depends on the specific items included in that order. More specifically, we assume that in addition to the period-dependent (joint) setup cost K_t, incurred for any order in period t, an item-specific setup cost is incurred for any item included in the order. Thus, let

$\kappa_{it} =$ setup cost incurred when ordering item i in period t,
$i = 1, \ldots, N,\ t = 1, \ldots, T.$

The mixed-integer programming formulation in §2 is easily adjusted to incorporate these item-specific setup costs. Add a new set of zero-one variables

$$y_{it} = \begin{cases} 1 & \text{if } x_{it} > 0, \\ 0 & \text{otherwise,} \end{cases}$$

as well as constraints

$$x_{it} \leqslant C_t y_{it}, \quad i = 1, \ldots, N,\ t = 1, \ldots, T, \tag{26}$$

$$y_{it} \leqslant Y_t, \quad i = 1, \ldots, N,\ t = 1, \ldots, T. \tag{27}$$

The new objective function becomes

$$z^* = \min\left\{\sum_{t=1}^{T}\left[K_t Y_t + \sum_{i=1}^{N} c_{it}x_{it} + h_{it}I_{it} + \kappa_{it}y_{it}\right]\right\}. \tag{28}$$

The mechanisms of both the (SP) and the (EH) heuristics are easily generalized, as well. Note that it is not necessary to solve each interval problem to optimality; To accelerate the procedure, one may terminate as soon as a solution is found within a given precision (δ%) of a lower bound. While it is unknown how the bounds for the heuristics' optimality gaps can be extended or how the heuristics can be designed to be asymptotically optimal and polynomially bounded, in practice we find that in particular the (EH) heuristic generates close-to-optimal solutions in a modest amount of time. To show this, we have conducted a numerical study, coding our heuristics in C++ and running them on a Sun 4000 work station with Solaris 7 and 2 GB of RAM.

In designing our study, we have followed the design of Maes and Van Wassenhove (1986, 1988), one of the most comprehensive comparisons of known heuristics, except that they confined themselves to instances with $N = T = 12$ items and periods, while we have systematically varied the number of items between 10 and 25 and the number of periods from 10 to 50. (Maes and Van Wassenhove restrict themselves to the case where only item-specific setup costs prevail, which remain constant across the complete planning horizon.) An additional difference is that, at the end of

Table 1. JIS: Gaps and running times of the EH heuristic with $N = 10$, $T = 15$ (limit six hours per instance).

	Density								
	Low			Medium			High		
TBO period	Low	Medium	High	Low	Medium	High	Low	Medium	High
Low-item TBO (%)	0.00	0.23	−1.10	0.73	0.80	0.31	0.29	0.79	0.06
Running time	30/18	75/19	—/126	452/30	56/23	60/29	6,154/55	—/126	6,370/98
Medium-item TBO (%)	0.72	1.73	1.52	0.37	0.32	0.20	0.79	0.04	0.49
Running time	—/80	2,551/54	—/110	252/34	—/126	—/117	—/145	—/150	—/167
High-item TBO (%)	−1.33	0.96	0.01	0.17	0.89	−0.43	1.09	0.01	0.49
Running time	—/119	—/98	—/52	—/238	—/210	—/232	—/448	—/170	—/170

the eighties, no solution method was capable of solving the model to optimality, even for moderate-size problems with $N = T = 12$. As a consequence, the quality of the proposed heuristics was gauged by their gap with respect to the best solution found after evaluation of (up to) 1,000 nodes in a tailor-made branch-and-bound tree. Today, we can solve these and many larger problems to optimality, enabling us to gauge the actual optimality gaps.

Our base set of problems has $N = 10$ items and a horizon of $T = 15$ periods. As in Maes and Van Wassenhove (1988), all demands $\{d_{it}\}$ are independently generated from a normal distribution with mean 100 and standard deviation of 10. With constant capacity levels C, we consider three levels for the "problem density," defined as the ratio $\sum_{t=1}^{T} C_t / \sum_{t=1}^{T} D_t = TC / \sum_{t=1}^{T} D_t$: *low density* where the ratio equals 2, *medium density* where it equals 4/3, and *high density* where it is 10/9. We set all variable cost rates $h_{it} = c_{it} = 1$. For each item $i = 1, \ldots, N$, we determine the fixed (item-specific) setup cost indirectly by first choosing the EOQ-cycle time, "time between orders (TBO)" $= \sqrt{2\kappa/hd} = \sqrt{2\kappa/100} = \sqrt{\kappa/50}$, and determining the κ value from this identity. The TBO-value is generated from a uniform distribution on the interval $[1, 3]$ when considering low TBO-values, the interval $[2, 6]$, when considering medium TBO-values, and $[5, 10]$ for the case of high TBO-values. The joint setup cost is calculated in the same way, i.e., from the identity $\text{TBO} = \sqrt{2K/100N}$.

We start by evaluating the (EH) heuristic with respect to its optimality gap and running time, compared to the complete horizon method (CHM)—the solution obtained by the standard CPLEX MIP-solver when applied to the full problem. We consider all 27 combinations that arise when combining the three problem densities, three product TBO values, and three period TBO values. For each of these 27 combinations, we have generated five distinct problem instances. We report in Table 1 the average running times in CPU seconds when solving the problem with CHM and with the (EH) heuristic, implemented with $\tau = 5$, $T_l = l$, $l = 1, \ldots, J = T$, and $\delta = 1\%$. We also report the optimality gap of the solution generated by this heuristic. A hyphen indicates that (one or more) problem instances could not be solved to optimality within six hours, in which case the reported optimality gap refers to the best found solution by CPLEX so far. (Some of the optimality gaps are negative, implying that the (EH) heuristic terminates with a better solution than CHM after six hours of running time!) We note that all optimality gaps are below 1.75%.

Where comparable, the CPU times appear to be of the same order of magnitude as those in state-of-the-art heuristics such as Stadtler (2003), even though differences between the problem instances and platforms make a precise comparison impossible.

Unless specified otherwise, when CHM is used, we employ the plant location formulation. Confirming prior experience with the JIS model, we have noticed that this formulation usually, although not necessarily, results in faster solutions. (In contrast, we use the network formulation, unless specified otherwise, for progressive interval

Table 2. JIS: Gaps and CPU seconds of four exact and two EH heuristic solutions (limit one hour per instance) with $N = 10$, $T = 15$.

Density	TBO	Net	Net w/cuts	Plant	Plant w/cuts	EH w/net	EH w/plant
Low	Low	* (30)	* (244)	* (67)	* (131)	1.85% (16)	2.35% (28)
Low	Medium	*	0.75%	1.03%	0.25%	1.15% (38)	1.42% (106)
Low	High	*	1.35%	0.90%	0.02%	5.32% (53)	2.19% (145)
Medium	Low	0.34%	0.64%	*	0.13%	1.17% (25)	1.16% (32)
Medium	Medium	0.82%	1.03%	0.46%	0.48%	0.12% (67)	* (297)
Medium	High	1.02%	0.67%	0.59%	0.14%	0.15% (136)	* (548)
High	Low	0.11%	0.40%	*	0.34%	0.31% (51)	0.18% (98)
High	Medium	0.78%	1.01%	0.62%	0.91%	0.20% (96)	* (288)
High	High	0.91%	1.15%	1.08%	1.16%	0.04% (392)	* (947)

heuristics, as it typically runs faster for these heuristics.) As a further benchmark for the (EH) heuristic, we have verified whether exact solutions (via CPLEX 7.1) could be significantly sped up if the problem formulation is strengthened by adding the cutting-plane constraints (see Barany et al. 1984a, b)

$$\sum_{t\in S} x_{it} \leqslant \sum_{t\in S}\left(\sum_{u=t}^{l} d_{iu}\right) y_{it} + I_{il}, \qquad i=1,\ldots,N \text{ and } l=1,\ldots T \;\; \forall\, S\subseteq\{1,\ldots,l\} \tag{29}$$

to the network formulation (and the same constraints, with x_{it} replaced by $\sum_{w=t}^{T} x_{itw}$, for the plant location formulation). More specifically, we have added the violated constraints in (29) after solving the LP relaxation of the complete problem and before invoking the CPLEX MIP solver. Table 2 revisits the nine categories (of five problem instances each) in Table 1, in which the item and period TBO are of the same type, i.e., in which they are both low, medium, or high. Each of the last six columns reports on one of six solution methods described below and executed with a one-hour time limit. The first reported number is the optimality gap with respect to the best among the six solutions, with a * denoting a 0% gap; where the CPU time is less than one hour, we report this measure within parentheses (in seconds). The six methods are: (1) CHM using the network flow formulation by itself; (2) CHM using the network flow formulation with the addition of violated cuts; (3) CHM with the plant location formulation by itself; (4) CHM using the plant location formulation with the addition of the above violated cuts; (5) the (EH) heuristic where each interval problem is solved with the network flow formulation; and (6) the (EH) heuristic with the plant location formulation. We conclude that the cuts in (29) do not result in major improvements either in terms of CPU time or in terms of the quality of the generated solutions. (Frequently, both attributes deteriorate, in fact.)

In Table 3, we show that the (EH) heuristic, again implemented with $\tau = 5$ and $T_l = l$, $l = 1, \ldots, J$, can be effectively used for significantly larger problem instances. Varying N from 5 to 25 and T from 10 to 50, we report the CPU running time in seconds. We specify the parameters as above, confining ourselves to the case where the problem density is medium, as is the "item TBO" and "period TBO" value. Three problem instances are generated for every combination of N and T.

Table 3. JIS: Running times for the (EH) heuristic.

Periods	10	25	50
5 item	7	42	124
10 item	29	184	524
15 item	416	2,694	4,310
20 item	1,600	9,372	16,159
25 item	20,335	66,634	58,264

Table 4. JIS: Gaps and CPU seconds for the CHM (within 1% of LB), the (EH) heuristic, and the (SP) heuristic with $N = 10$, $T = 15$.

TBO period	Low	Medium	High
Low-item TBO	0.9%/3.9%	0.1%/7.0%	0.3%/8.6%
Running time	9/7/1	9/8/1	13/6/1
Medium-item TBO	1.3%/11.3%	0.8%/11.5%	0.5%/11.0%
Running time	262/19/1	390/19/1	208/18/1
High-item TBO	2.8%/33.8%	2.9%/25.9%	1.2%/19.7%
Running time	7,854/23/1	5,235/24/1	6,750/26/1

As mentioned in §4, the (SP) heuristic is considerably faster than the (EH) heuristic, but it generally generates solutions with significantly larger optimality gaps. Table 4 illustrates this for a set of 27 problem instances, all with $N = 10$ and $T = 15$ and parameters as specified in our basic set. Focusing on the medium problem density case, we consider all nine combinations of product TBO and period TBO-values, generating three instances for each. We report on the running times of CHM (terminated when a solution is found within 1% of the best lower bound), the (EH) heuristic, and the (SP) heuristic. We also report both heuristics' average optimality gaps. While the optimality gap for the (EH) heuristic is never in excess of 3%, and on average equals 1.2%, that of the (SP) heuristic may be as high as 33% and is on average 14.7%.

In Table 5, we evaluate the optimality gaps for the JS problem with period-dependent setup costs only. To this end, we consider a set of 45 problems with $N = 10$ and $T = 30$ periods; we again consider all nine combinations of TBO and problem density values and generate five problem instances for each for these combinations. We report the CPU times of the CHM and the (EH) and the (SP) heuristics, along with the optimality gaps associated with both heuristics. Once again, the (EH) heuristic generates solutions within 1% of optimality and does so within approximately 20 seconds of CPU time. The CHM often requires several thousands of CPU seconds (i.e., many hours of CPU time); its solution times depend greatly on the parameters of the problem. The (SP) heuristic is an order of magnitude faster than the (EH) heuristic, but may generate solutions with optimality gaps as large as 15%. Clearly, the (EH) heuristic can be employed for far larger problem instances.

Table 5. JS: Gaps and CPU seconds for CHM, the (EH) heuristic, and the (SP) heuristic with $N = 10$, $T = 30$.

Density	Low	Medium	High
Low TBO	0.2%/6.0%	0.5%/1.6%	0.2%/0.5%
Running time	44/17/1	48/17/1	21/17/1
Medium TBO	0.8%/12.1%	0.1%/3.4%	0.1%/5.0%
Running time	742/29/3	712/19/1	85/19/1
High TBO	0.5%/15.0%	0.1%/3.9%	0%/6.0%
Running time	1,150/22/2	3,973/21/1	127/19/1

Table 6. JIS: Gaps and CPU seconds for CHM and the (EH) heuristic with item-dependent fixed costs only ($K_t = 0$, $N = 10$, $T = 15$).

Density	Low	Medium	High
Low-item TBO (%)	0.5	0.5	0.4
Running time	2/7	7/7	-/13
Medium-item TBO (%)	1.7	1.6	−1.1
Running time	-/28	-/44	-/103
High-item TBO (%)	1.0	−0.7	−2.5
Running time	-/41	-/126	-/358

Finally, we consider the case with item-dependent setup costs only. Table 6 compares the (EH) heuristic and CHM for the nine relevant item TBO and problem density values in Table 1 (as in Table 2, the CHM is terminated after one hour). All of our conclusions regarding the quality of the (EH) heuristic solutions and the running times continue to apply for this special case of the JIS model.

Returning to the general JIS model, Belvaux and Wolsey (2000, 2001) observe that in many applications, at most one or two items may be ordered per period. The authors refer to such models as "small bucket models." Once again, the mechanics of the (SP) and (EH) heuristics are straightforwardly adjusted to accommodate this restriction. For small bucket models, even the branch-and-bound methods of §5 are easily adjusted. Choosing $\tau = \lceil \alpha \log T \rceil$, as in (24), this gives rise to a polynomial time implementation of the heuristics for the JIS model, where the complexity bound is a factor $O(N)$ or $O(N^2)$ larger than the corresponding complexity bound for the JS model.

Similarly, the mechanics of the (SP) and (EH) heuristics are easily adjusted to (i) add capacity limits for individual items in each period; (ii) allow for multiple capacitated order batches in every period, as in Anily and Tzur's (2005, 2006) MIMV-problem; (iii) address the hierarchical planning problems in Graves (1982) or Van Roy and Wolsey (1987), which differ from the JIS model with capacity limits for each item only by allowing the (joint) capacity to be increased with overtime at a linear penalty cost; or (iv) handle any of the other variants mentioned in Belvaux and Wolsey (2000, 2001).

References

Aggarwal, A., J. Park. 1993. Improved algorithms for economic lot size problems. *Oper. Res.* **41** 549–571.

Ahuja, R., D. Hochbaum. 2004. Solving linear cost dynamic lot sizing problems in $O(n \log n)$ time. *Oper. Res.* Forthcoming.

Anily, S., M. Tzur. 2005. Shipping multiple-items by capacitated vehicles—An optimal dynamic programming approach. *Transportation Sci.* **39**(2) 233–248.

Anily, S., M. Tzur. 2006. Algorithms for the multi-item, multi-vehicles capacitated dynamic lot sizing problem. *Naval Res. Logist.* **53**(2) 157–169.

Baker, K., P. Dixon. 1978. An algorithm for the dynamic lot size problem with time-varying production capacity constraints. *Management Sci.* **24** 710–720.

Barany, I., T. Van Roy, L. Wolsey. 1984a. Strong formulations for multi-item capacitated lot-sizing. *Management Sci.* **30** 1255–1261.

Barany, I., T. Van Roy, L. Wolsey. 1984b. Uncapacitated lot-sizing: The convex hull of solutions. *Math. Programming Stud.* **22** 32–43.

Belvaux, G., L. Wolsey. 2000. *bc-prod*—A specialized branch-and-cut system for lot-sizing problems. *Management Sci.* **46** 724–738.

Belvaux, G., L. Wolsey. 2001. Modelling practical lot-sizing problems as mixed-integer programs. *Management Sci.* **47** 993–1007.

Bitran, G., H. Yanasse. 1982. Computational complexity of the capacitated lot size problem. *Management Sci.* **28** 1174–1186.

Bixby, R. 2001. MIP: A progress report. Talk at 1st Columbia Optimization Day. Columbia University, New York.

Chen, H., D. Hearn, C. Lee. 1994. A new dynamic programming method for the single capacitated dynamic lot size model. *J. Global Optim.* **4** 285–300.

Chung, C., M. Lin. 1988. An $O(T^2)$ algorithm for the NI/G/NI/ND capacitated single item lot size problem. *Management Sci.* **34** 420–426.

Dixon, P., E. Silver. 1981. A heuristic solution for the multi-item, single level, limited capacity lot sizing problem. *J. Oper. Management* **2** 23–39.

Dogramaci, A., J. Panayiotopoulos, N. Adam. 1981. The dynamic lotsizing problem for multiple items under limited capacity. *IEE Trans.* **13** 294–303.

Eisenhut, P. 1975. A dynamic lotsizing algorithm with capacity constraints. *Trans. Amer. Inst. Electr. Engrgs.* **7** 170–176.

Eppen, G., R. Martin. 1987. Solving multi-item capacitated lot sizing problems using variable redefinitions. *Oper. Res.* **35** 832–848.

Federgruen, A., M. Tzur. 1991. A simple forward algorithm to solve general dynamic lot-sizing models with n periods in $O(n \log n)$ or $O(n)$ time. *Management Sci.* **37** 909–925.

Federgruen, A., M. Tzur. 1994a. Minimal forecast horizons and a new planning procedure for the general dynamic lot sizing model: Nervousness revisited. *Oper. Res.* **42** 456–469.

Federgruen, A., M. Tzur. 1994b. Capacitated lot-sizing models. Working paper, Graduate School of Business, Columbia University, New York.

Federgruen, A., M. Tzur. 1994c. The joint replenishment problem with time-varying parameters: Efficient, optimal and ϵ-optimal solutions. *Oper. Res.* **42** 1067–1087.

Federgruen, A., M. Tzur. 1999. Time-partitioning heuristics: Application to one-warehouse, multiitem, multiretailer lot-sizing problems. *Naval Res. Logist.* **46** 463–486.

Florian, M., M. Klein. 1971. Deterministic production planning with concave costs and capacity constraints. *Management Sci.* **18** 12–20.

Florian, M., J. Lenstra, A. Rinnooy Kan. 1980. Deterministic production planning: Algorithms and complexity. *Management Sci.* **26** 669–679.

Gavish, B., R. Johnson. 1990. A fully polynomial approximation scheme for single-product scheduling in a finite capacity facility. *Oper. Res.* **38** 70–83.

Graves, S. 1982. Using Lagrangean techniques to solve hierarchical production planning problems. *Management Sci.* **28** 260–275.

Karni, R., Y. Roll. 1982. A heuristic algorithm for the multi-item lotsizing problem with capacity constraints. *Amer. Inst. Indust. Engrgs. Trans.* **13** 249–256.

Kuik, R., M. Solomon, L. van Wassenhove. 1994. Batching decisions: Structures and models. *Eur. J. Oper. Res.* **18** 213–263.

Lambrecht, M., J. Vander Eecken. 1978. A capacity constrained single-facility dynamic lot-size model. *Eur. J. Oper. Res.* **2** 132–136.

Leung, J., T. Magnanti, R. Vachani. 1989. Facets and algorithms for capacitated lot sizing. *Math. Programming* **45** 331–359.

Maes, J., L. van Wassenhove. 1986. Multi item single level capacitated dynamic lotsizing heuristics: A computational comparison (Part I: Static case; Part II: Rolling horizon). *IIE Trans.* **18** 114–129.

Maes, J., L. van Wassenhove. 1988. Multi-item single-level capacitated dynamic lot-sizing heuristics: A general review. *J. Oper. Res. Soc.* **39** 991–1004.

Martin, R. 1987. Generating alternative mixed-integer programming models using variable redefinitions. *Oper. Res.* **35** 820–831.

MEMIPS. 1997. Model enhanced solution methods for integer programming software. Esprit Project 20118, Public report reference DR1.1.10.

Nahmias, S. 1989. *Production and Operations Analysis.* Homewood, Boston, MA.

PAMIPS. 1995. Development of parallel algorithms and software for mixed-integer programming in industrial scheduling. Esprit Project 8755, Public report reference DR4.3.5.

Pochet, Y. 1988. Valid inequalities and separation for capacitated economic lot sizing. *Oper. Res. Lett.* **7** 109–116.

Salomon, M. 1990. Deterministic lotsizing models for production planning. Ph.D. dissertation, Erasmus University, Rotterdam, The Netherlands.

Shaw, D., A. Wagelmans. 1998. An algorithm for single-item capacitated economic lot sizing with piecewise linear production costs and general holding costs. *Management Sci.* **44** 831–838.

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.

Suerie, C., H. Stadtler. 2003. The capacitated lot-sizing problem with linked lot sizes. *Management Sci.* **49** 1039–1054.

Trigeiro, W., L. Thomas, J. McClain. 1989. Capacitated lot sizing with setup times. *Management Sci.* **35** 353–366.

Van Hoesel, C., A. Wagelmans. 1996. An $O(T^3)$ algorithm for the economic lot-sizing problem with constant capacities. *Management Sci.* **42** 142–150.

Van Hoesel, C., A. Wagelmans. 2001. Fully polynomial approximation schemes for single-item capacitated lot-sizing problems. *Math. Oper. Res.* **26** 339–357.

Van Nunen, J., J. Wessels. 1978. Multi item lot size determination and scheduling under capacity constraints. *Eur. J. Oper. Res.* **2** 36–41.

Van Roy, T., L. Wolsey. 1987. Solving mixed integer programming problems using automatic reformulation. *Oper. Res.* **35** 45–48.

Wagelmans, A., C. van Hoesel, A. Kolen. 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.