European Journal of Operational Research 75 (1994) 332–341
North-Holland

A new heuristic approach for the multi-item dynamic lot sizing problem

Ömer Kırca * and Melih Kökten
Middle East Technical University, Ankara, Turkey

Received April 1992; revised September 1992

Abstract: In this paper a framework for a new heuristic approach for solving the single level multi-item capacitated dynamic lot sizing problem is presented. The approach uses an iterative item-by-item strategy for generating solutions to the problem. In each iteration a set of items are scheduled over the planning horizon and the procedure terminates when all items are scheduled. An algorithm that implements this approach is developed in which in each iteration a single item is selected and scheduled over the planning horizon. Each item is scheduled by the solution of a bounded single item lot sizing problem where bounds on inventory and production levels are used to ensure feasibility of the overall problem. The performance of this algorithm is compared to some well-known heuristics over a set of test problems. The computational results demonstrated that on the average our algorithm outperforms other algorithms. The suggested algorithm especially appears to outperform other algorithm for problems with many periods and few items. In the literature these problems are considered as hard.

Keywords: Lot sizing; Production planning; Heuristics

1. Introduction

In this paper the multi-item single level lot sizing problem (CLSP) is considered. The problem involves scheduling of n items over a planning horizon of T periods. Demand for each item in every period is known and has to be satisfied without backlogging. In each period there exists a bottleneck production resource with known capacity. The problem is to schedule these items such that the sum of the fixed and variable production costs and inventory holding cost over the horizon is minimized, subject to the demand and capacity constraints in each period. A mathematical formulation of the CLSP is given below:

CLSP($\mathcal{N}$)

$$Z = \text{Min} \sum_{i \in \mathcal{N}} \sum_{t=1}^{T} (A_{it} y_{it} + h_{it} I_{it} + c_{it} x_{it}) \tag{1}$$

* *Correspondence to:* Prof. Dr. Ömer Kırca, Currently with the Department of Decision Sciences, National University of Singapore, 0511 Singapore.

SSDI 0377-2217(93)E0197-6

$$\text{subject to} \quad I_{i,t-1} + x_{it} - I_{it} = d_{it}, \quad i \in \mathcal{N},\ t = 1,\ldots,T, \tag{2}$$

$$\sum_{i \in \mathcal{N}} a_i x_{it} \le C_t, \quad t = 1,\ldots,T, \tag{3}$$

$$x_{it} \le \min\left(C_t/a_i,\ \sum_{r=t}^{T} d_{ir}\right) y_{it}, \quad i \in \mathcal{N},\ t = 1,\ldots,T, \tag{4}$$

$$I_{it},\ x_{it} \ge 0, \quad i \in \mathcal{N},\ t = 1,\ldots,T, \tag{5}$$

$$y_{it} \in \{0, 1\}, \quad i \in \mathcal{N},\ t = 1,\ldots,T, \tag{6}$$

where: *Parameters (input)*:

n : Number of items to be produced.
$\mathcal{N}$: Set of items to be produced ($= \{1,\ldots,n\}$).
T : Number of production periods.
d_{it} : Demand for item i in period t ($i \in \mathcal{N}; t = 1,\ldots,T$).
C_t : Capacity of the resource in period t ($t = 1,\ldots,T$).
a_i : Capacity requirement for one unit of item i ($i \in \mathcal{N}$).
A_{it} : Set-up cost for item i in period t ($i \in \mathcal{N}$; $t = 1,\ldots,T$).
c_{it} : Production cost for one unit of item i in period t ($i \in \mathcal{N}$; $t = 1,\ldots,T$).
h_{it} : Inventory holding cost for one unit of item i in period t ($i \in \mathcal{N}$; $t = 1,\ldots,T$).

Decision variables (output):

x_{it} : Quantity of item i to be produced in period t.
I_{it} : Quantity of inventory of item i to be carried from period t to period $t+1$.
$y_{it} = \begin{cases} 1 & \text{if item } i \text{ is to be produced in period } t\ (x_{it} > 0), \\ 0 & \text{otherwise.} \end{cases}$

Throughout this paper it will be assumed that all demand values are expressed in terms of the capacity units, i.e. $a_i = 1$ for all $i \in \mathcal{N}$.

It has been shown by Florian et al. [7] that CLSP($\mathcal{N}$) is NP-hard. The problem has been of continuing interest and there exist many approaches, exact or heuristic, for the CLSP($\mathcal{N}$). Initially, Manne [15] and later Dzielinski and Gomory [5] and Lasdon and Terjung [13] have developed set partitioning and column generation based procedures for solving the linear relaxation of the CLSP($\mathcal{N}$). Gelders et al. [8] suggest a branch and bound algorithm in which lower bounds are obtained by the Lagrangian relaxation of the capacity constraints, (3). Later Barany et al.[1] and Eppen and Martin [6] have considered solving the problem directly by general mixed integer programming codes. Barany et al. [1] make use of the extreme point properties of the uncapacitated problem to add a set of cuts to the model (1)–(6). Eppen and Martin use variable redefinition techniques in order to develop a model which results in tighter LP-bounds. However all of the above discussed exact procedures fail to solve problems that have realistic sizes. Therefore most research on CLSP($\mathcal{N}$) has concentrated on developing heuristic procedures.

The heuristic approaches reported in the literature can be classified into two groups, as mathematical programming based and common sense approaches. The heuristics suggested by Thizy and Van Wassenhove [16], Trigeiro [17], and Cattrysse et al. [2] belong to the first class. The first two heuristics are Lagrangian relaxation based procedures. The procedures suggested in Cattrysse et al. [2], use the set partitioning formulation of Manne [15] for the problem and generate candidate plans for the items by several well-known heuristics. The feasible schedules are obtained by rounding off the LP -relaxation of the set partitioning problem.

Most common sense heuristics use of a period-by-period approach, in which CLSP($\mathcal{N}$) is solved on a period by period basis. In each period, lot sizes for all items are determined on the basis of a cost savings criterion. In a given period, future demand of items are scheduled to be produced in that period until no further cost savings are possible or until all the capacity at that period is exhausted. Some of the heuristics reported in the literature which use this approach are the ones due to Dixon and Silver [3], Lambrecht and Vanderveken [12], Günther [9], and Dogramaci et al. [4].

In this paper we suggest a different approach for generating solutions to CLSP($\mathcal{N}$). Our approach is called 'item-by-item' approach. In Section 2 this approach is outlined and an algorithm which uses this

approach is presented. In Section 3, our algorithm is compared to other well known heuristics in the literature. Conclusions and suggestions for further research are presented in Section 4.

2. Item-by-item approach

The item-by-item approach generates solutions to the CLSP($\mathcal{N}$) iteratively. In each iteration, a set of items among the items which are not yet scheduled is selected and production schedules over the planning horizon for this set of items are determined. Then available capacity in each period are updated with respect to the production schedules of these items. The procedure terminates when all items are scheduled. An outline of the item-by-item approach is given below.

Define:

k: Iteration counter.
$\mathcal{W}$: Set of items which are not scheduled yet.
P_t: Available capacity that can be used for items in $\mathcal{W}$.

Item-by-item approach

Step 1. (Initialization) Set:
$k := 0$, $\mathcal{W} := \mathcal{N}$, $Z := 0$, and $P_t := C_t$ for all $t = 1, \ldots, T$. Go to Step 2.

Step 2. (Candidate item set determination). Set:
$k := k + 1$ and determine a set of candidate items to be scheduled, $\mathcal{S}_k$, such that $\mathcal{S}_k \subset \mathcal{W}$ and $|\mathcal{S}_k| \geq 1$ based on a specified 'selection rule'. For each period t, determine upper bounds on the total production, $U_t(\mathcal{S}_k)$, and inventory carrying, $Q_t(\mathcal{S}_k)$. Go to Step 3.

Step 3. (Scheduling the candidate set). Solve the following CLSP($\mathcal{S}_k$):

CLSP($\mathcal{S}_k$)

$$Z_k = \text{Min} \sum_{i \in \mathcal{S}_k} \sum_{t=1}^{T} (A_{it} y_{it} + h_{it} I_{it} + c_{it} x_{it}) \tag{7}$$

subject to
$$I_{i,t-1} + x_{it} - I_{it} = d_{it}, \quad i \in \mathcal{S}_k, \ t = 1, \ldots, T, \tag{8}$$
$$\sum_{i \in \mathcal{S}_k} x_{it} \leq U_t(\mathcal{S}_k), \quad t = 1, \ldots, T, \tag{9}$$
$$\sum_{i \in \mathcal{S}_k} I_{it} \leq Q_t(\mathcal{S}_k), \quad t = 1, \ldots, T, \tag{10}$$
$$x_{it} \leq P_t y_{it}, \quad i \in \mathcal{S}_k, \ t = 1, \ldots, T, \tag{11}$$
$$I_{it}, x_{it} \geq 0, \quad i \in \mathcal{S}_k, \ t = 1, \ldots, T, \tag{12}$$
$$y_{it} \in \{0, 1\}, \quad i \in \mathcal{S}_k, \ t = 1, \ldots, T. \tag{13}$$

Let the solution be Z_k and $(\tilde{x}_{it}, \tilde{I}_{it}, \tilde{y}_{it})$ for $i \in \mathcal{S}_k$, $t = 1, \ldots, T$.

Step 4. (Update). Let:

$$P_t := P_t - \sum_{i \in \mathcal{S}_k} \tilde{x}_{it}, \quad t = 1, \ldots, T,$$

$$Z := Z + Z_k \quad \text{and} \quad \mathcal{W} := \mathcal{W} - \mathcal{S}_k.$$

If $\mathcal{W} \neq \emptyset$, then go to Step 2, else stop with the solution Z and $(\tilde{x}_{it}, \tilde{I}_{it}, \tilde{y}_{it})$ for $i \in \mathcal{N}$, $t = 1, \ldots, T$.

In order to implement the above outlined approach we specify Steps 2 and 3 further in detail.

a) Candidate Item Set Determination: In Step 2, on the basis of a selection criterion, a set of items among the items which are not yet scheduled must be selected. The overall efficiency of the approach

will be sensitive to the criterion with which these items are selected. A selection criterion must specify the sequence and the number of items selected in each iteration. If in each iteration one item is selected then a single item capacitated lot sizing problem must be solved in Step 3. On the other hand, if initially all in items are selected then the original CLSP($\mathcal{N}$) should be solved. Selecting few items in each iteration may reduce computational effort to the expense of high cost solutions, however, if more items are selected then the computational effort will increase.

b) Determination of Upper Bounds on Production and Inventories: Production and inventory holding for the items in any candidate set, $\mathcal{S}_k$, should be restricted in order to ensure feasible plans for the remaining items. In other words, the schedules for the items in $\mathcal{S}_k$ determined in Step 3 must satisfy the following inequalities:

$$P_t - \sum_{i\in\mathcal{S}_k} x_{it} \geq 0 \text{ and } 0 \leq \sum_{i\in\mathcal{S}_k}\sum_{r=1}^{t} x_{ir} + \sum_{i\in\mathcal{W}}\sum_{r=1}^{t} d_{ir} \leq \sum_{r=1}^{t} P_r, \quad t=1,\ldots,T. \tag{14}$$

Inequalities in (14) state that the production quantities of the items in set $\mathcal{S}_k$ must be determined such that in each period the remaining capacity for the items in $\mathcal{W}-\mathcal{S}_k$ must be nonnegative and similarly the cumulative demand for those items in $\mathcal{W}-\mathcal{S}_k$ must not exceed the remaining cumulative capacity. These are the necessary feasibility conditions for CLSP($\mathcal{W}-\mathcal{S}_k$). Therefore in CLSP($\mathcal{S}_k$) in each period the total production as well as the total cumulative production must be bounded in order to ensure obtaining feasible schedules for the remaining items. Note that restricting cumulative production is same as imposing upper bounds on the inventory levels since cumulative production of an item can be expressed as inventory level plus the cumulative demand of that item. Therefore, total production and inventory carried for the items in set $\mathcal{S}_k$ are restricted in each period as in (9) and (10) respectively. A procedure for determining $U_t(\mathcal{S}_k)$ and $Q_t(\mathcal{S}_k)$ is developed by utilizing similar concepts to the ones suggested in Dogramaci et al. [4] which can be derived from the neccessary feasiblity conditions for the CLSP. For given sets $\mathcal{S}_k$ and $\mathcal{W}$, upper bounds on total inventory and production for each period can be found as follows:

(i) *Upper bound on the total inventory for the items in* $\mathcal{W}$, $Q_t(\mathcal{W})$: We first determine an upper bound on the total inventory level of all the items not yet scheduled. Let

$$Q_t(\mathcal{W}) = \text{minimum}\left\{\sum_{r=t+1}^{T} D_r,\ \sum_{r=1}^{t} K_r,\ D_{t+1} + Q_{t+1}(\mathcal{W})\right\}, \quad t=T-1,\ldots,1, \tag{15}$$

where

$$Q_T(\mathcal{W}) = 0, \qquad D_t = \sum_{i\in\mathcal{W}} d_{it}, \qquad K_t = P_t - D_t \quad \text{for } t = T-1,\ldots,1.$$

Maximum level of inventories at any given period is equal to the minimum of the total cumulative demand until the end of the horizon, cumulative slack capacity up to that period, and the sum of the total demand and cumulative slack capacity of the next period.

(ii) *Upper bound on total inventory for the items in* $\mathcal{S}_k$, $Q_t(\mathcal{S}_k)$: Set $Q_T(\mathcal{S}_k) = 0$ and

$$Q_t(\mathcal{S}_k) = \text{minimum}\left\{Q_t(\mathcal{W}),\ \sum_{i\in\mathcal{S}_k}\sum_{r=t+1}^{T} d_{ir},\ Q_{t+1}(\mathcal{S}_k) + \sum_{i\in\mathcal{S}_k} d_{i,t+1}\right\}, \quad t=T-1,\ldots,1. \tag{16}$$

Total amount of inventory that can be carried to the next period for the items in set $\mathcal{S}_k$ is equal to the minimum of the upper bound on the total inventory level of all items, the cumulative demand of these items until the end of the horizon, and the sum of the upper bound on inventories and the total demand in the next period.

(iii) *Upper bound on total production*, $U_t(\mathcal{S}_k)$:

$$U_t(\mathcal{S}_k) = \text{minimum}\left\{Q_t(\mathcal{S}_k) + \sum_{i\in\mathcal{S}_k} d_{it},\ P_t\right\}, \quad t=1,\ldots,T. \tag{17}$$

Total capacity that can be allocated to the items in set $\mathscr{S}_k$ at a period t must not be larger than the sum of the total demand of that period and the upper bound on the total inventory level of these items that can be carried from that period. If this value is larger than the available capacity at that period, then the capacity that can be used for these items is set to the total available capacity.

c) Solving CLSP($\mathscr{S}_k$): In Step 3, in order to schedule the items in set $\mathscr{S}_k$ a capacitated dynamic lot sizing problem with bounds on the inventory levels, CLSP($\mathscr{S}_k$), is solved. This problem, may be solved by using either exact or heuristic procedures. Obviously, heuristic solutions may result in higher total cost for the overall problem.

The item-by-item approach discussed above is implemented with a specific candidate item selection strategy. With this strategy, in each iteration a single item from the set of items which are not yet scheduled is selected. Several different criteria are tested in order to determine the order in which the items are to be selected and these are discussed in Kökten [11]. Among all those criteria tested the one which selects the item that has the largest estimated total cost per unit demand among the items not yet scheduled resulted in lower total cost for CLSP($\mathscr{N}$). For each item i the average cost per unit (V_i) is estimated by using the well-known Economic Order Quantity (EOQ) concept where uniform demand is assumed. The average cost per unit for an item is computed by dividing the EOQ cost per period to mean demand per period of that item and an expression for V_i is given in (18). Then the item that has the largest average cost per unit is selected for scheduling. The selection rule assumes time independent cost coefficients, i.e., $c_{it} = c_i$, $h_{it} = h_i$, and $A_{it} = A_i$. This procedure is called the 1-ITEM Algorithm and is outlined below.

1-ITEM Algorithm

Step 1. (Initialization) Let:

$k := 0$, $\mathscr{W} := \mathscr{N}$, $Z := 0$, $P_t := C_t$, for all $t = 1, \ldots, T$. For each item $i = 1, \ldots, n$ compute the coefficient V_i as follows:

$$V_i = \sqrt{2A_i h_i \bar{d}_i}/\bar{d}_i, \tag{18}$$

where $\bar{d}_i = (1/T)\Sigma_{t=1}^{T} d_{it}$.

Go to Step 2.

Step 2. (Candidate item determination). Set:

$k := k + 1$,

$j = \text{argmax}_{i \in \mathscr{W}}\{V_i\}$ and compute $Q_t(j)$ and $U_t(j)$ using (16) and (17) respectively. Go to Step 3.

Step 3. (Scheduling item j). Solve CLSP($\mathscr{S}_k$) as defined in (7)–(13) with $\mathscr{S}_k = \{j\}$ and let the solution be Z_j and $(\tilde{x}_{jt}, \tilde{I}_{jt}, \tilde{y}_{jt})$, $t = 1, \ldots, T$.

Step 4. (Update). Set:

$P_t := P_t - \tilde{x}_{jt}$, $t = 1, \ldots, T$;

$Z := Z + Z_j$ and $\mathscr{W} := \mathscr{W} - \{j\}$.

If $\mathscr{W} \neq \emptyset$, then go to Step 2, else stop with the solution Z and $(\tilde{x}_{jt}, \tilde{I}_{jt}, \tilde{y}_{jt})$ for $j \in \mathscr{N}$, $t = 1, \ldots, T$.

During the initial computational tests with the algorithm, it was observed that usually the items that are scheduled at the initial iterations use almost all of the capacity at certain specific periods. Therefore high cost solutions are obtained for the remaining items. In order to avoid this drawback, a capacity adjustment factor (λ_j) is used not to allocate all the available capacities to the initially selected items. This adjustment procedure is as follows:

First $Q_t(j)$ and $U_t(j)$ values are computed using (16) and (17) respectively. Then, for each period t, the minimum inventory level that must be carried to the next period ($L_t(j)$) is computed. This is a lower bound on the inventory level that ensures obtaining a feasible schedule for item j and computed as in (19).

Let $L_T(j) = 0$ and

$$L_t(j) = \text{maximum}\{0,\ d_{jt+1} + L_{t+1}(j) - U_{t+1}(j)\}, \quad t = T-1, \ldots, 1. \tag{19}$$

Then for any $t = 1, \ldots, T$, if $L_t(j) \le \lambda_j Q_t(\mathscr{W}) \le Q_t(j)$, then $Q_t(j)$ is set to $\lambda_j Q_t(\mathscr{W})$ and $U_t(j)$ values are computed with the new values of $Q_t(j)$ using (17) again.

We have experimented with several different schemes for determining the capacity adjustment factor. The scheme which computes λ_j as a function of capacity tightness ratio, set-up to holding cost ratios, and Time Between Orders (TBO_i) has provided lowest cost solutions. In the algorithm for each item j a different capacity adjustment factor is used and it is computed as follows:

$$\lambda_j = \alpha(1+\rho) f_j / F, \tag{20}$$

$$\alpha = \frac{\sum_{t=1}^{T} C_t}{\sum_{i=1}^{n}\sum_{t=1}^{T} d_{it}}, \qquad \rho = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{A_i}{h_i} - \frac{1}{n}\sum_{i=1}^{n}\frac{A_i}{h_i}\right|,$$

where

$$f_i = (\mathrm{TBO}_i - 1)\left[\frac{\bar{d}_i \mathrm{TBO}_i(\mathrm{TBO}_i - 1)}{2}\right], \qquad F = \sum_{i \in \mathscr{W}} f_i, \quad \textit{and} \quad \mathrm{TBO}_i = \max\left\{1, \sqrt{\frac{2A_i}{h_i \bar{d}_i}}\right\}.$$

Here α is the mean capacity tightness factor, ρ is the mean deviation of set-up to holding cost ratio, and TBO_i is the time between orders of item i in terms of EOQ. The values f_i indicate the capacity requirements of a specific item in terms of its TBO_i and mean demand. The term in the brackets in the equation for f_i is the estimate for total inventory per cycle for item i and this is multiplied by the number of periods in which inventory carried in the cycle. Larger portions of the available capacitites will be allocated to the items that have larger TBO_i and/or larger mean demand. Notice that all the parameters except F should be computed once at the initialization step.

In Step 3 of the 1-ITEM algorithm the single item bounded lot sizing problem, (CLSP(j)) is solved by the Two Stage Algorithm (TSA) of Kırca [10]. This procedure implements dynamic programming and makes use of the extreme point properties of the problem to reduce the state space. The algorithm is modified to handle the upper bounds on the inventory levels in (9).

3. Computational results

The computational performance of 1-ITEM is compared with the performances of four well-known heuristics in the literature over a total of 120 test problems. First two heuristics are the 'common sense' heuristics that use 'period-by-period' approach and these are the ones suggested in Lambrecht and Vanderveken [12] (LV) and Dixon and Silver [3] (DS). The third heuristic (ABCX) is an extension of the ABC heuristic of Maes and Van Wassenhovel [14]. This heuristic again uses the 'period-by-period' approach and it is a collection of 360 fast capacitated lot sizing algorithms. The last one (HEUR4) is a set partitioning and column generation based heuristic which is reported in Cattrysse et al. [2]. Although HEUR4 requires extensive computation time it is one of the most cost efficient procedures reported yet in the literature.

The test problems are the ones that are reported in [2] and in which there are three problem sets each differing in the number of items and periods, (n, T), as (50, 8), (20, 20), and (8, 50) respectively. The problems in each set differ with respect to three factors as capacity utilization (U), capacity requirement per unit of item, i.e. a_i, (C), and demand variation (S). For each factor there are two levels. These levels are low (L) and high (H) for capacity utilization and demand variation, and constant (C) and varying (V) for unit capacity requirements. In each combination of the two different levels of the three factors there are 5 different randomly generated problems and therefore in total there are 40 problems in each problem set.

The solutions of the ABCX and HEUR4 for the test problems were supplied by the authors of [2] and therefore here we used their results. In the problem set with (n, T) = (8, 50) for four problems results

Table 1
Summary of the computational experiments

Average % deviation from the best solution [a]	$(n, T) = (50, 8)$					$(n, T) = (20, 20)$					$(n, T) = (8, 50)$ [b]				
	LV	DS	ABCX	HEUR4	1-ITEM	LV	DS	ABCX	HEUR4	1-ITEM	LV	DS	ABCX	HEUR4	1-ITEM
LU-CC-LS	2.90	2.85	3.60	0.00	0.96	6.32	6.50	3.51	0.04	0.22	4.80	5.33	2.34	2.28	0.00
LU-CC-HS	2.37	2.49	4.22	0.00	1.81	4.55	4.41	3.11	0.18	0.30	3.45	4.37	3.14	2.60	0.11
LU-VC-LS	6.22	5.26	5.62	0.00	2.00	9.04	7.50	5.21	0.11	0.48	5.18	8.04	1.81	1.54	0.19
LU-VC-HS	2.99	3.38	4.17	0.00	2.02	3.30	2.48	1.70	0.13	0.17	2.86	4.62	1.86	1.74	0.00
LU average	3.62	3.49	4.40	0.00	1.70	5.80	5.22	3.38	0.11	0.29	4.07	5.59	2.29	2.04	0.07
LU max	14.98	12.95	8.32	0.00	4.08	22.00	25.98	12.95	0.80	1.31	12.52	21.13	5.41	5.20	0.95
HU-CC-LS	1.02	0.15	1.79	0.13	0.02	11.61	16.95	8.68	3.18	0.24	6.47	8.84	7.38	5.13	0.00
HU-CC-HS	0.92	0.60	2.02	0.23	1.47	7.13	6.81	6.47	1.98	0.74	12.78	10.82	15.06	9.66	0.00
HU-VC-LS	2.90	0.36	1.81	0.08	0.70	17.04	14.05	6.01	2.40	0.00	15.71	17.56	12.58	8.89	0.52
HU-VC-HS	1.58	0.12	2.04	0.09	1.07	5.83	3.86	5.74	0.73	0.41	12.63	9.32	9.03	7.63	0.00
HU average	1.60	0.31	1.92	0.13	0.82	10.40	10.42	6.73	2.07	0.35	11.90	11.64	11.01	7.83	0.13
HU max	5.69	1.88	3.74	0.76	2.63	25.81	29.94	15.88	8.27	2.98	28.73	29.55	26.49	17.09	2.60
Total average	2.61	1.91	3.16	0.07	1.26	8.10	7.82	5.05	1.09	0.33	7.99	8.61	6.63	4.94	0.11
LS average	3.26	2.16	3.21	0.05	0.92	11.00	11.25	5.85	1.43	0.24	8.04	9.94	6.03	4.46	0.18
HS average	1.96	1.65	3.11	0.08	1.59	5.20	4.39	4.25	0.75	0.41	7.93	7.28	7.27	5.41	0.03
CC average	1.80	1.52	2.91	0.09	1.06	7.40	8.67	5.44	1.35	0.38	6.87	7.34	6.98	4.92	0.03
VC average	3.42	2.28	3.41	0.04	1.45	8.80	6.97	4.66	0.84	0.26	9.09	9.89	6.32	4.95	0.18
Number of problems best	0	5	0	30	5	0	0	0	19	21	0	1	1	2	37
CPU Seconds [c]															
LU average	1.62	2.04	86.00	384.00	2.55	1.68	2.10	87.00	770.00	3.31	26.12	4.28	46.00	2974.00	5.42
HU average	1.76	0.88	78.00	362.00	2.36	3.85	2.04	89.00	1934.00	2.97	21.03	3.32	152.00	4735.00	5.01
Total average	1.69	1.46	82.00	373.00	2.46	2.02	2.07	88.00	1352.00	3.14	23.58	3.80	94.00	3854.00	5.22

[a] % deviation from the best = $100*(Z_i / Z^b - 1)$, where Z_i the cost of the respective heuristic, $Z^b = \text{Min}_{i=1,\ldots,5}(Z_i)$.

[b] In this set of problems results of four problems for HEUR4 were not available. For these four problems the results of HEUR2 given in [2] are used.

[c] For LV, DS, and 1ITEM on a Commodore AT with 80286/80287 processors running at 16MHz. For ABCX and HEUR4 are as reported in [2], on an Olivetti M24 with 8086/8087 processors running at 8MHz.

with HEUR4 were not available. For these four problems the results of HEUR2 which is another column generation based heuristic reported in [2] were used. The solutions for LV, DS, and 1-ITEM were obtained on a Commodore AT with 80286/80287 processors running at 16MHz. The codes for LV and DS were in FORTRAN, and 1-ITEM was coded in PASCAL.

In Table 1 the results of the computational experiments with LV, DS, ABCX, HEUR4, and 1-ITEM are summarized for each problem set. The average percentage deviation of the costs found with each algorithm from the smallest is reported in the first part of the table. In the second part the total number of times the respective algorithms provided the best solution are given. Note that the sum of the numbers here may exceed the total number of problems tested because in some problems two or more heuristics may provide the best solution. Finally the average computation times in CPU seconds are reported.

The results of the computational experiments, in general, show that 1-ITEM outperforms other heuristics and HEUR4 is the next best performing heuristic. 1-ITEM was the most efficient algorithm in all problem sets except the one that has many items and few periods, i.e., $(n, T) = (50, 8)$. The best performance of 1-ITEM was observed in the problem set with 8 items and 50 periods, i.e. few items and many periods. In this set of problems on the average 1-ITEM resulted in 4% better solutions than HEUR4. Furthermore, in this set, out of a total of 40 problems tested, in 37 of those it provided the best cost solutions which corresponds to 92.5% of all problems. The other advantage of 1-ITEM was that it's performance is highly consistent. In all the 120 problems tested 1-ITEM has never resulted in a deviation more than 5% of the best solution.

When the computation times are compared, on the average, 1-ITEM required slightly more CPU seconds than DS and was comparable with LV. The CPU seconds provided for ABCX and HEUR4 are not directly comparable to the ones for LV, DS, and 1-ITEM due to the differences of the processors used in running those respective algorithms. However, considering the differences of the speeds of those two different processors used, we may conclude that 1-ITEM would require considerably less CPU time than ABCX and HEUR4 if all those were run on a same processor. With respect to the cost and computation time efficiency of all those heuristics we tested here we can safely conclude that with 1-ITEM schedules with lower costs for CLSP($\mathcal{N}$) may be obtained with significantly less computation time compared to the next best performing heuristic.

The only case where 1-ITEM did not outperform other heuristics was for the problems with many items and few periods, i.e., $(n, T) = (50, 8)$. In this type of problems many items compete for production in some specific periods. In 1-ITEM, the items that are scheduled in the initial iterations usually consume almost all the capacity in those periods. As a result those items that are scheduled last either cannot be produced in those specific periods or they result in lot-for-lot schedules thus high cost plans are obtained for those items.

Table 2
Comparison of 1-ITEM with "period-by-period" heuristics

Average % deviation from the best solution [a]	$(n, T) = (50, 8)$				$(n, T) = (20, 20)$				$(n, T) = (8, 50)$ [b]			
	LV	DS	ABCX	1-ITEM	LV	DS	ABCX	1-ITEM	LV	DS	ABCX	1-ITEM
LU	2.05	1.92	2.82	0.15	5.50	4.92	3.08	0.00	4.07	5.59	2.29	0.07
HU	1.36	0.07	1.68	0.58	10.10	10.13	6.44	0.07	11.76	11.49	10.87	0.00
LS	2.43	1.33	2.38	0.11	10.75	11.00	5.61	0.00	7.90	9.80	5.88	0.05
HS	0.99	0.67	2.12	0.62	4.86	4.05	3.91	0.07	7.93	7.28	7.27	0.03
CC	1.02	0.74	2.12	0.29	7.08	8.35	5.13	0.07	6.87	7.34	6.98	0.03
VC	2.39	1.25	2.37	0.43	8.53	6.70	4.39	0.00	8.96	9.74	6.18	0.05
Total average	1.71	1.00	2.25	0.36	7.81	7.53	4.26	0.04	7.92	8.54	6.58	0.04
Max. deviation	12.75	10.76	6.58	1.97	22.00	25.98	12.95	1.41	28.73	29.55	26.49	0.95
Number of problems best	3	13	1	25	0	0	1	39	0	1	1	38

[a] As described in Table 1.

In Table 2 we compare the performance of 1-ITEM with the performances of the 'period-by-period' heuristics, namely LV, DS, and ABCX. The superiority of the 'item-by-item' approach over the 'period-by-period' heuristics is more clearly seen in Table 2. Out of 120 problems, in 102 cases 1-ITEM provided the best solution, which corresponds to 85% of all problems. In this case even for the problem set with many items and few periods 1-ITEM outperforms the other 'period-by-period' heuristics.

4. Conclusion

In this paper we have suggested a new heuristic approach, called item-by-item, for solving the multi-item dynamic lot sizing problem. We have demonstrated that than an algorithm which uses this approach is more efficient than some other well-known procedures suggested in the literature. The algorithm is highly superior for the problems that have few items and many periods where these are considered to be hard problems in the literature.

Further research on this approach may concentrate on developing more efficient strategies in implementing this approach for solving CLSP($\mathscr{N}$). Different item selection and capacity adjustment rules may be developed in order to improve the performance of the 1-ITEM algorithm especially for the many items few periods problems. By using several different item selection strategies different algorithms that implement the item-by-item approach can be developed. One such procedure may be to select more than one item in each iteration and the resulting subproblems can be solved by an appropriate heuristic available in the literature. The insight we have gained in this research suggests that the item-by-item approach with a dynamic item selection strategy may result in a better performance. This strategy may be to implement single item selection strategy for some number of iterations, until k items remain in set $\mathscr{W}$. Then a fast and efficient heuristic can be implemented to solve the remaining k-item dynamic lot sizing problem. A good value for k may be around two or three. The performance of the item-by-item approach may be improved significantly, especially for the problems that have many items and few periods, by the use of stronger bounds on inventory and production level in (9) and (10).

Acknowledgements

We would like to thank Marc Salomon, from Erasmus University, for providing us the test data and the codes of LV and DS. We also extend our appreciation to the authors of [2], namely Dirk Cattrysse, Johan Maes, and Luk N. van Wassenhove, for making available to us the detailed results of their computational experiments.

References

[1] Barany, I., Van Roy, T.J., and Wolsey, L.A., "Strong formulations for multi-item capacitated lotsizing", *Management Science* 30/10 (1984) 1255–1261.

[2] Cattrysse, D., Maes, J., and Van Wassenhove, L.N., "Set partitioning and column generation heuristics for capacitated lotsizing", *European Journal of Operational Research* 46 (1990) 38–47.

[3] Dixon, P.S., and Silver, E.A., "A heuristic solution procedure for the multi-item single level, limited capacity, lotsizing problem", *Journal of Operations Management* 2/1 (1981) 23–39.

[4] Dogramaci, A., Panayiotopoulos, J.C., and Adam, N.R., "The dynamic lot sizing problem for multiple items under limited capacity", *AIIE Transactions* 13/4 (1981) 294–303.

[5] Dzielinski, B.P. and Gomory, R.E., "Optimal programming of lot sizes, inventory and labor allocations", *Management Science* 11/9 (1965) 874–890.

[6] Eppen, G.D., and Martin, R.K., "Solving multi-item capacitated lot-sizing problems using variable redefinition", *Operations Research* 35/6 (1987) 832–835.

[7] Florian, M., Lenstra, J.K, and Rinnooy Kan, A.H.G., "Deterministic production planning: Algorithms and complexity", *Management Science* 26/7 (1980) 669–679.

[8] Gelders, L.F., Maes, J., and Van Wassenhove, L.N., "A branch and bound algorithm for the multi-item single level capacitated dynamic lotsizing problem", in: S. Axsaster et al. (eds.), *Multi-Stage Production Planning and Inventory Control*, Lecture Notes in Economics and Mathematical Systems 266, Springer-Verlag, Berlin, 1986, 92–108.

[9] Günther, H.O., "Planning lot sizes and capacity requirements in a single stage production system", *European Journal of Operational Research* 31/2 (1987) 223–231.

[10] Kırca, Ö., "An efficient algorithm for the capacitated single item dynamic lot size problem", *European Journal of Operational Research* 45/1 (1990) 15–34.

[11] Kökten, M., "Efficient procedures for multi-item lot sizing problems based on tight formulations", Unpublished M.S. Thesis, Middle East Technical University, Ankara, 1990.

[12] Lambrecht, M.R., and Vanderveken, H., "Heuristic procedure for the single operation, multi-item loading problem", *AIIE Transactions* 11/4 (1979) 319–326.

[13] Lasdon, L.S., and Terjung, R.C., "An efficient algorithm for multi-item scheduling", *Operations Research* 19/4 (1971) 946–969.

[14] Maes, J., and Van Wassenhove, L.N., "A simple heuristic for the multi-item single level capacitated lotsizing problem", *Operations Research Letters* 4/6 (1986) 265–273.

[15] Manne, A.S., "Programming of economic lot-sizes", *Management Science* 4/2 (1958) 115–135.

[16] Thizy, J.M., and Van Wassenhove, L.N., "Lagrangian relaxation for the multi-item capacitated lot-sizing problem: A heuristic implementation", *IIE Transactions* 17/4 (1985) 308–313.

[17] Trigeiro, W.W., "A dual cost heuristic for the capacitated lot sizing problem", *IIE Transactions* 19/1 (1987) 67–72.