On alternative mixed integer programming formulations and LP-based heuristics for lot-sizing with setup times

M Denizel[1]* and H Süral[2]

[1]*Sabancı University, Orhanlı, Tuzla/İstanbul, Turkey; and* [2]*Middle East Technical University, Ankara, Turkey*

We address the multi-item, capacitated lot-sizing problem (CLSP) encountered in environments where demand is dynamic and to be met on time. Items compete for a limited capacity resource, which requires a setup for each lot of items to be produced causing unproductive time but no direct costs. The problem belongs to a class of problems that are difficult to solve. Even the feasibility problem becomes combinatorial when setup times are considered. This difficulty in reaching optimality and the practical relevance of CLSP make it important to design and analyse heuristics to find good solutions that can be implemented in practice. We consider certain mixed integer programming formulations of the problem and develop heuristics including a curtailed branch and bound, for rounding the setup variables in the *LP* solution of the tighter formulations. We report our computational results for a class of instances taken from literature.

Keywords: capacitated lot-sizing; reformulation; valid inequalities; heuristics

Introduction

We consider a multi-item, dynamic, capacitated lot-sizing problem, which appears in several manufacturing environments both as a standalone problem and a subproblem in broader decision-making situations (see Karimi *et al*, 2003 for a recent review of lot-sizing problems). The problem is considered over T time periods where a single resource's capacity is to be allocated to N items according to their demands in each period. Each item requires a certain processing time on the resource, which is considered to be critical in the completion of production, possibly associated with a bottleneck operation. A lot to be produced for each item in a period requires a setup that generates some downtime for the resource. No direct costs due to setups are assumed. The setup times and also the processing times determine the capacity needs of items. When the available capacity is not sufficient to meet the demand of an item in a period, it has to be supplied from inventory carried from earlier periods at the expense of a unit inventory holding cost per period. Backlogging is not allowed. We also assume that production costs are stationary over time and can thus be ignored. The objective is to find a feasible schedule with minimum total inventory holding cost. The problem belongs to a class of problems that are difficult to solve. It can be shown that even the feasibility problem is NP-complete (Maes *et al*, 1991). Due to this nature, it deserves separate attention within the class of CLSPs. Its properties that differ from the other problems in this class need investigation.

The problem can typically be formulated as follows:

P

$$\text{Minimize} \quad z = \sum_{i=1}^{N}\sum_{t=1}^{T} h_i I_{it} \tag{P.1}$$

$$\text{subject to} \quad I_{i,t-1} + x_{it} - I_{it} = d_{it} \quad \text{each } i \text{ and } t \tag{P.2}$$

$$\sum_{i=1}^{N} (a_i x_{it} + s_i y_{it}) \leqslant C_t \quad \text{each } t \tag{P.3}$$

$$x_{it} \leqslant m_{it} y_{it} \quad \text{each } i \text{ and } t \tag{P.4}$$

$$I_{i0} = 0 \quad \text{each } i \tag{P.5}$$

$$x_{it} \geqslant 0, I_{it} \geqslant 0 \quad \text{each } i \text{ and } t \tag{P.6}$$

$$y_{it} = 0 \text{ or } 1 \quad \text{each } i \text{ and } t \tag{P.7}$$

The variables x_{it} and I_{it} denote the amount of item i produced in period t and the inventory level of item i at the end of period t, respectively. y_{it} is a binary variable indicating whether a setup time for item i in period t is incurred or not. The parameters, h_i, d_{it}, a_i, s_i and C_t are the cost of carrying one unit of item i in inventory from a period to the next, the demand for item i in period t, the processing time of item i, the setup time for item i, and the

Correspondence: M Denizel, Graduate School of Management, Sabancı University, Orhanlı 34956, Tuzla/İstanbul, Turkey.
E-mail: denizel@sabanciuniv.edu

available capacity in time units in period t, respectively. $m_{it} = \min\{\sum_{k=t}^{T} d_{ik}, (C_t - s_i)/a_i\}$ is an upper bound on production quantity and we assume that $s_i \leqslant C_t$ for each i and t.

In problem P, (P.1) minimizes the total inventory carrying costs. Equation (P.2) represents the inventory balance equations for each item and each period and together with the non-negativity of the inventory variables it ensures that demand is satisfied on time. Time capacity limitations on the total processing and setup times in each period are imposed by (P.3). Equation (P.4) makes sure that a setup is incurred for each production run and by (P.5) it is assumed that there exist no initial inventories on hand. Equations (P.6) and (P.7) guarantee the non-negativity of the production and the integrality of the setup variables, respectively.

Although the multi-item capacitated lot-sizing problem has generated considerable interest in the literature, incorporation of the setup times has not been considered much. Bahl and Zionts (1987), Karayel (1984), Trigeiro *et al* (1989), Diaby *et al* (1992), and Süral (1996) have addressed the problem with setup times. However, the problem P is different from those given in Bahl and Zionts (1987), Trigeiro *et al* (1989), Diaby *et al* (1992), such that Trigeiro *et al* (1989) include setup cost and the other two papers consider the overtime option. Karayel (1984) and Süral (1996) consider the same problem as P: Karayel (1984) develops a heuristic based on removing infeasibilities from a lot for lot schedule while, Süral (1996) proposes a Lagrangean relaxation and a period by period heuristic combined in a branch and bound algorithm. Alternative mixed integer programming formulations of the lot-sizing problem with setup times are studied by Stadtler (1996) for the multi-level environment with setup costs and overtime options.

Maes *et al* (1991) and Alfieri *et al* (2002) analyse the performance of LP-based rounding heuristics for the multi-level and the single-level lot-sizing problems without setup times, respectively. Despite encouraging results, there is no study in the literature, which extends this general approach to the problems with setup times. As a matter of fact, the NP-complete feasibility problem limits the use of similar methods for the problem when the setup times are not negligible. However, for the cases where the capacity is not a hard constraint and overtime or subcontracting is possible, it is worth exploring the performance of LP-based rounding heuristics for the problem. The approach described in this paper extends the LP-based rounding heuristics to the problem with setup times and no setup costs.

The purpose of this paper is two-fold. First is to analyse and compare the performances of different mixed integer programming (MIP) formulations of the problem and their linear relaxations. Exploring the possibilities for developing a quick and easy solution method based on the linear relaxation of a tight formulation is the second purpose. To our knowledge, this is the first attempt to compare the performance of three alternative tight formulations in solving CLSP using a general purpose MIP solver and heuristic methods.

In the first section, we review three alternative tight formulations of the problem. The next section provides an experimental analysis of formulations on a set of test problems taken from the literature. By using CPLEX, we first test the LP relaxations of the three MIPs with (a) primal simplex, (b) dual simplex, and (c) the barrier algorithm to determine the best LP solution method, as in Alfieri *et al* (2002), and then we solve the MIP formulations to optimality using the best method. In the third section, we describe several LP-based heuristics including a curtailed branch and bound procedure with rounding only in the very first step of the enumeration. We report all computational results on the performances of the heuristic approaches in the fourth section. The last section concludes our study.

MIP formulations

The model P presented in the first section is a standard formulation with O(NT) continuous and binary variables, and constraints. Although all solution methods (Diaby *et al*, 1992; Karayel, 1984; Süral, 1996; Trigeiro *et al*, 1989) that we mention have been developed based on this formulation, it is well known that it is a weak formulation of the CLSP where, the strength of a formulation is measured by the objective value of its LP relaxation. Here, we consider two alternative formulations from the literature (Alfieri *et al*, 2002; Eppen and Martin, 1987; Stadtler, 1996; Süral, 1996): the transportation problem formulation, TP, with O(NT^2) continuous and O(NT) binary variables, and O(NT^2) constraints and the shortest path formulation, SP, with O(NT^2) continuous and O(NT) binary variables, and O(NT) constraints. We, also, consider a third formulation, which we call the improved standard formulation, IS, obtained by an *a priori* addition of some valid inequalities developed by Barany *et al* (1984) for the uncapacitated lot-sizing problem.

Transportation problem formulation (TP)

Define z_{itr} as the quantity produced in period t to satisfy the demand of item i in period r, where $r \geqslant t$. Other variables are same as before. The current formulation makes use of the advantage that stems from the strongest formulation of the single item uncapacitated lot-sizing problem.

TP

$$\text{Minimize} \quad z = \sum_{i=1}^{N} \sum_{r=1}^{T} \sum_{t=1}^{r-1} (r-t) h_i z_{itr} \qquad \text{(TP.1)}$$

$$\text{subject to} \quad \sum_{t=1}^{r} z_{itr} = d_{ir} \quad \text{each } i \text{ and } r = 1, \ldots, T \qquad \text{(TP.2)}$$

$$\sum_{i=1}^{N}\left(\sum_{r=t}^{T} a_i z_{itr} + s_i y_{it}\right) \leqslant C_t \quad \text{each } t \tag{TP.3}$$

$$z_{itr} \leqslant d_{ir} y_{it} \quad \text{each } i \text{ and } t, r = t, \ldots, T \tag{TP.4}$$

$$\sum_{r=t}^{T} z_{itr} \leqslant m_{it} y_{it} \quad \text{each } i \text{ and } t \tag{TP.5}$$

$$z_{itr} \geqslant 0 \quad \text{each } i \text{ and } t, r = t, \ldots, T \tag{TP.6}$$

$$y_{it} = 0 \text{ or } 1 \quad \text{each } i \text{ and } t \tag{TP.7}$$

In this formulation, (TP.1) minimizes the total inventory carrying cost, (TP.2) assures that total production for item i in periods 1 through r is equal to the demand in period r, (TP.3) maintains that total production and setup times in period t do not exceed the available (time) capacity C_t, and (TP.4) incurs a setup for each production run. Equation (TP.5), which provides an aggregate bound on total production in a period, is actually redundant in TP but provides a valid inequality in its LP relaxation. Equations (TP.6) and (TP.7) impose non-negativity on the production variables and integrality on the setup variables, respectively.

Shortest path problem formulation (SP)

Define u_{itk} as the fraction of total demand for periods t through k of item i that is produced in period t. This formulation extends the reformulation of the single item uncapacitated lot-sizing problem as a shortest path problem, developed by Eppen and Martin (1987), to the capacitated case.

SP

$$\text{Minimize} \quad z = \sum_{i=1}^{N}\sum_{t=1}^{T}\sum_{k=t}^{T} H_{itk} u_{itk} \tag{SP.1}$$

subject to

$$\sum_{i=1}^{N}\left(\sum_{k=t}^{T} a_i D_{itk} u_{itk} + s_i y_{it}\right) \leqslant C_t \quad \text{for each } t \tag{SP.2}$$

$$-\sum_{k=1}^{t-1} u_{ik,t-1} + \sum_{k=t}^{T} u_{itk} = 0 \quad \text{for each } i \text{ and } t = 2, \ldots, T \tag{SP.3}$$

$$\sum_{t=1}^{T} u_{i1t} = 1 \quad \text{for each } i \tag{SP.4}$$

$$\sum_{k=t}^{T} u_{itk} \leqslant y_{it} \quad \text{for } t = 1 \text{ and } i \ni d_{i1} > 0 \text{ and each } i \text{ and } t \ni t \neq 1 \tag{SP.5}$$

$$\sum_{k=t}^{T} D_{itk} u_{itk} \leqslant m_{it} y_{it} \quad \text{for each } i \text{ and } t \tag{SP.6}$$

$$u_{itk} \geqslant 0 \quad \text{for each } i \text{ and } t, \text{ and } k = t, \ldots, T \tag{SP.7}$$

$$y_{it} = 0 \text{ or } 1 \quad \text{for each } i \text{ and } t \tag{SP.8}$$

where

$$D_{itk} = \sum_{j=t}^{k} d_{ij} \text{ and } H_{itk} = \sum_{j=t+1}^{k} h_i D_{ijk}$$

In this formulation (SP.1) minimizes the total inventory carrying cost, (SP.2) restricts the (time) capacity usage in each period, (SP.3) and (SP.4) define the path equations for each item, and (SP.5) incurs a setup for each production run. Note that for $t=1$ only the items with nonzero demands are considered to make sure that no unnecessary setups are incurred. Equation (SP.6), which provides an aggregate bound on total production in a period, is actually redundant in SP except for items with nonzero demands in $t=1$. For these items, it also guarantees that the required setup time is incurred if production is necessary to meet the demand of some future period. We however impose it for all i and t, since, it acts as a valid inequality in the LP relaxation of SP. Equations (SP.7) and (SP.8) impose non-negativity on the production variables and integrality on the setup variables, respectively.

In our computational studies the LP relaxations of the above two formulations always resulted with the same solution value. This is a result that may be worth further exploration.

The last alternative MIP formulation of the problem is as follows.

Improved standard formulation with Barany et al *(1984) cuts (IS)*

IS

$$\text{Minimize} \quad z = \sum_{i=1}^{N}\sum_{t=1}^{T} h_i I_{it} \tag{IS.1}$$

$$\text{subject to} \quad I_{i,t-1} + x_{it} - I_{it} = d_{it} \quad \text{each } i \text{ and } t \tag{IS.2}$$

$$\sum_{k=1}^{t-1} x_{ik} + d_{it} y_{it} \geqslant \sum_{k=1}^{t} d_{ik} \quad \text{each } i \text{ and } t = 2, \ldots, T \tag{IS.3}$$

$$\sum_{i=1}^{N} (a_i x_{it} + s_i y_{it}) \leqslant C_t \quad \text{each } t \tag{IS.4}$$

$$x_{it} \leqslant m_{it} y_{it} \quad \text{each } i \text{ and } t \tag{IS.5}$$

$$x_{it} \geqslant 0, I_{it} \geqslant 0 \quad \text{each } i \text{ and } t \qquad \text{(IS.6)}$$

$$y_{it} = 0 \text{ or } 1 \quad \text{each } i \text{ and } t \qquad \text{(IS.7)}$$

The above model is the same as the weak formulation P, except that a set of inequalities (IS.3), developed in Barany *et al* (1984) is added to tighten the formulation to some extent.

Computational analysis of MIP formulations and their LP relaxations

In our experimental analysis, all computations are carried out on test problems taken from Trigeiro *et al* (1989). The test problem set in this section includes 20 problems in total, with sizes $N=12{,}24$ and $T=15{,}30$. Setup times are item dependent. All problems assume per unit (time) capacity utilization for all items, that is, $a_i=1$ for all $i=1, \ldots, N$ (see Trigeiro *et al* (1989) for more details of the test instances). Since there are no setup costs in our case, a drawback of Trigeiro *et al*'s (1989) test instances regarding the large ratio of setup costs to setup times is eliminated. We coded all our procedures in C within the MS Visual C++ 6.0 environment in connection with the CPLEX 7.0 Callable Library, on an IBM PC with Intel Pentium III processor.

In the first part of our experiments, we made a brief comparison of CPU times needed by the three core LP algorithms in CPLEX namely, the primal, dual, and barrier algorithms, to solve the LP relaxations of MIP formulations, and compared the solution quality of these LP relaxations with respect to their objective function values. Results are given in Table 1.

Entries in Table 1 associated with each problem size are the average values of five problem instances. For each problem instance, the formulations TP and SP gave the same linear objective function value that is higher than that of both P and IS. DP(%) = 100(LB−L(P))/LB and DIS(%) = 100(LB−L(1S))/LB show the relative deviations of the lower bounds obtained by P and IS, where L(P) and L(IS) respectively denote the objective function values of their continuous solutions (LP relaxations) and LB refers to the best lower bound given by the objective function value of the continuous solutions of SP (or TP). Although the LP relaxations of P and IS are solved faster than the other two formulations by all the LP algorithms, the large deviations of their (continuous) solutions from those of SP and TP (98% for P and 14% for IS) make them less attractive. Alfieri *et al* (2002) report the relative deviation of the LP relaxation of P as about 60% on average for the lot-sizing problem with setup costs but no setup times, which may indicate the structural difference between the two problems. Between the two LP relaxations of SP and TP, SP is solved faster than TP on average in our experiments. This particular result was also observed by Alfieri *et al* (2002) in their experiments. It seems that the LP relaxation of SP formulation is slightly better than that of TP from a computational viewpoint. Among three LP algorithms, the dual algorithm is faster for all the formulations. Based on this, we did all our further experiments using the dual algorithm only.

In the second part of our experiments, we performed a comparison of the elapsed times needed to obtain an optimal solution to the MIP formulations controlling the features of the CPLEX MIP solver. To do so, we first ran all the test problems by tuning off the default features of CPLEX (ie, rounding-up heuristic, adding cutting planes) and then repeated the experiment this time with the default features on. In these experiments, we did not consider P any further due to the weakness of its LP solutions. The results given in Table 2 are for the solutions obtained by CPLEX with features off and a time limit of three hours.

In Table 2, $D(\%)$ is the relative integrality gap computed as $100(U(\cdot)-\text{LB})/\text{LB}$, where $U(\cdot)$ denotes the integer solution value of the formulation $(\cdot)$ and LB is the best lower bound as defined before. SP and TP are superior to IS in terms of finding both the highest number of best integer

Table 1 Deviation of the linear solutions of P and IS relative to that of SP (or TP) and CPU times by three core algorithms

	P	IS	CPU (s)								
			TP			SP			IS		
N × T	DP (%)	DIS (%)	B	P	D	B	P	D	B	P	D
12 × 15	97.15	9.76	0.59	0.31	0.21	0.42	0.24	0.11	0.12	0.06	0.07
12 × 30	99.34	17.97	3.90	2.68	1.57	1.55	2.07	0.75	0.48	0.27	0.21
24 × 15	96.72	13.38	1.33	0.94	0.71	0.88	0.76	0.46	0.31	0.21	0.14
24 × 30	99.68	13.29	8.63	7.02	3.55	5.06	6.01	1.93	1.26	0.92	0.54
Average	98.22	13.60	3.61	2.74	1.51	1.98	2.27	0.81	0.54	0.36	0.24

B, barrier; P, primal; D, dual algorithms.
DP (%): 100(LB−L(P))/LB where LB is the linear solution value of SP (or TP).
DIS (%): 100(LB−L(IS))/LB where LB is the linear solution value of SP (or TP).

solutions and the integer solutions with smaller $D(\%)$ values on average. The best integer solutions are found 15 and 14 times by SP and TP, respectively, while IS found only 7. On average, the relative errors are about 23% for SP and TP, and 26% for IS. Furthermore, SP and TP are less expensive compared to IS from a computational viewpoint.

In experiments using the default features of CPLEX, we increased the time limit by 1 hour. This made the results more comparable with the results given in Table 2 due to the extra time needed for the CPLEX features like the rounding-up heuristics and cutting plane generation. The results tabulated in Table 3 indicate that there is no significant difference among the three formulations in terms of the relative gap and the solution time. The average relative gap is almost the same (about 20%) for all formulations, but the best integer solutions are found 14 times by SP and IS and 10 times by TP. Default features with an extra hour have improved the average relative gap by 3% for SP and TP and by 6% for IS. These results all suggest that SP and TP perform better than IS for solving the lot-sizing problem with setup times using a general purpose MIP solver. Apparently, IS competes (and provides slightly better results in our case) with SP and TP within a general purpose MIP solver which adds cutting planes to tighten the solution space.

We should note that our computations proved the optimality of only seven best solutions out of 20 and that the relative deviation of the optimum from the lower bound (ie, the integrality gap) has been found as 14% on average. Stadtler (1996) has found the integrality gap as 7% on average for small-size problem instances in the multi-level setting and noted that higher gaps occur when there are tight capacity constraints. For the remaining 13 problems, the relative deviation of the best solution from the linear solution value is 23% on average for all formulations. One can, therefore, argue that there is about 9% error in the best-known solutions found by CPLEX for those 13 problems, and that the time limitation can be extended to obtain better solutions. As a matter of fact, in our preliminary experiments we tried spending more time than 3 or 4 h to solve the test problems optimally. However, even the larger solution times (eg, allowing more than 16 h of running time) did not help to solve these medium-sized problems optimally. This justifies the need for good heuristics to solve the lot-sizing problem with setup times in reasonable computation times.

Table 2 Deviation of the integer solution relative to the linear solution for the three formulations and CPU times by CPLEX with all features off*

	SP				*TP*				*IS*			
$N \times T$	*Best*	*D (%)*	*Node*	*CPU (min)*	*Best*	*D (%)*	*Node*	*CPU (min)*	*Best*	*D (%)*	*Node*	*CPU (min)*
12 × 15	5	17.91	58 661	5.98	5	17.91	90 165	17.47	5	17.91	750 802	39.07
12 × 30	4	29.28	454 607	181.18	1	29.79	248 098	181.61	—	37.21	1 532 316	180.14
24 × 15	4	5.57	528 998	147.39	5	5.32	323 948	153.32	—	6.69	1 606 673	180.14
24 × 30	2	38.85	256 913	181.07	3	40.38	175 948	180.87	2	42.58	794 599	180.32
Average		22.90	324 795	128.91		23.35	209 540	133.32		26.09	1 201 566	144.92

Best: the number of times that the best solution has been found.
D (%): $100(U(\cdot)-\mathrm{LB})/\mathrm{LB}$.
Node: the number of nodes explored.
*Time limit is 3 h IBM PC with Intel Pentium III.

Table 3 Deviation of the integer solution relative to the linear solution for SP and TP and CPU times by CPLEX with all features on*

	SP				*TP*				*IS*			
$N \times T$	*Best*	*D (%)*	*Node*	*CPU (min)*	*Best*	*D (%)*	*Node*	*CPU (min)*	*Best*	*D (%)*	*Node*	*CPU (min)*
12 × 15	5	17.91	30 341	4.16	5	17.91	37 595	9.38	5	17.91	69 238	7.82
12 × 30	2	24.32	336 669	241.15	1	24.97	171 778	241.92	2	24.21	444 086	240.08
24 × 15	5	5.20	516 985	181.01	3	5.23	310 591	203.11	5	5.20	677 483	184.45
24 × 30	2	32.91	178 103	240.93	1	33.46	111 432	241.33	2	32.24	316 510	240.12
Average		20.08	265 525	166.81		20.39	157 849	173.93		19.89	376 829	168.12

Best: the number of times that the best solution has been found.
D (%): $100(U(\cdot)-\mathrm{LB})/\mathrm{LB}$.
Node: the number of nodes explored.
*Time limit is 4 h on IBM PC with Intel Pentium III.

LP-based rounding heuristics and a curtailed branch and bound

LP-based rounding heuristics basically entail a simple enumeration process in which fractional solution values are rounded to integers in a sequence of iterative steps. Our experiments have revealed that a large portion of setup variables take on integer values in the first LP relaxation of MIP formulation. On average 79 (7)%, 77 (9)%, and 75 (6)% of integer variables of the test instances take the value of 1 (0) for SP, TP and IS, respectively, and this figure increases a little as the size of the problem instances increases. Hence, the LP-based rounding heuristic rules would be guiding us to round up about 14–20% of the setup variables.

Different strategies are applied for selecting the variables with fractional solution values in each step. This idea is pursued by Maes *et al* (1991) for the multi-level lot-sizing problems and Alfieri *et al* (2002) for the single-level lot-sizing problems without setup times. In this study, we extended the idea to the lot-sizing problems with setup times.

We mainly employ two policies: the first one is for deciding on the frequency with which we solve an LP (ie, to make an iteration), and the second one is for selecting which fractional setup variable(s) to round up to one in an iteration. We consider two different iteration frequencies.

One-by-one: Fix only one fractional setup variable at a time by picking the one with the highest fractional value. Fixing refers to rounding up to one.

Group-by-group: Define a round up threshold value k and fix only the set of fractional setup variables with values $\geqslant k$ at a time.

In all cases, we do not round down unless it is necessary. If we cannot find a variable to round up even when the solution is fractional, we round down to find a feasible solution. In an iteration, we consider the following three selection rules to define priorities for the set of eligible fractional setup variables to be rounded up.

No rules: It means that the above iteration frequency policies dominate the way we proceed. The variable with the largest fractional value will have the first priority to be rounded up. This rule is a general rule and applicable for any MIP. The following rules however make use of the problem structure to search for a better feasible solution.

Item based: Items are sorted in the order of nonincreasing inventory holding costs. We start with the first item i in the list and apply the above iteration policies for item i, and proceed with the next item in the list, and so on. Ties are broken with respect to the time periods (item produced in a later period is given priority) and item indices (smaller first).

Time based: We start with $t=T$ and apply the above iteration policies for time T to round up the fractional setup variables to one, and proceed with $t=t-1$, until $t=1$. Ties are broken with respect to the holding costs (item with a higher cost is given priority) and item indices (smaller first). Süral (1996) proposes a heuristic based on time decomposition of the same problem where single-period subproblems are successively solved starting with the last period T to minimize the cost of inventory carried from the previous period. The time-based rule resembles this approach in terms of the decomposition principle and the priority given to the late periods in the rounding process.

Main steps of a LP-based heuristic

1. Solve the LP relaxation.
2. Set y_{it} to 1 if $y_{it}=1$ in the LP solution.
3. Scan fractional y_{it}'s (>0.02), and determine a (a group of) candidate variable(s) according to the iteration frequency rule.
   - Among the candidate variables, pick y_{it}'s according the selection rule, and set those y_{it}'s to 1, and go to Step 1.
   - Else go to Step 4
4. Set $y_{it}=0$ for all y_{it} with $0\leqslant y_{it}\leqslant 0.02$.
5. If the solution is feasible halt the algorithm. Otherwise, initiate an iterative enumeration to resolve the infeasibility.

Curtailed branch and bound

The heuristic approaches explained above fix the fractional variables in the relaxed solution by applying some simple rules. This may lead to infeasible solutions to the problem. To overcome this, we implemented a partial branch and bound which implicitly enumerates all solutions before being trapped in an infeasible one. It works in the following way. In the root node, we solve the LP relaxation and fix all variables with positive integer solution values. Then we continue with a standard branch and bound procedure. This approach, called the curtailed branch and bound, is implemented in both Maes *et al* (1991) and Alfieri *et al* (2002) for the lot-sizing problem with negligible setup times.

When even the curtailed branch and bound fails to find a feasible solution, one has to make changes in the integer values initially fixed based on the LP solution. This, however, requires starting from scratch with a completely new strategy. Fortunately, in our experimental analysis this type of infeasibility had occurred only once indicating a good performance for our approach.

As a matter of fact, the curtailed branch and bound is the best strategy in the selection of variables with fractional values for rounding in an LP-based heuristic, because it implicitly enumerates all the solutions after the initial fixing at the first step of the heuristics. Therefore, in need of a good heuristic approach, it may be worthwhile to optimize the rounding rules even at the expense of some additional computational time.

Computational analysis

LP-based heuristics

We developed 12 variations of the LP-based rounding heuristics. Heuristic H1 applies no selection rules while heuristics H2 and H3 adopt the item-based and the time-based selection rules, respectively. Each heuristic was implemented three times each with a different threshold value ($k = 0.75$, 0.85, 0.95) under the group-by-group policy and once under the one-by-one policy, denoted by $k = \infty$. We halted the algorithm even if a feasible solution cannot be identified. All heuristics were tested in the same computing environment described before. We enlarged our test bed including 15 more problems with $N = 6$ and $T = 15$, 30. We were able to verify optimality in 14 of these 15 additional problems. We experimented with the LP relaxation of all three MIP formulations and solved the LP subproblems by the dual algorithm. Results are given in Tables 4–6.

In Tables 4–6, we report the average percent deviation of the heuristic solution values from the lower bounds (or the optimal solutions if known), the average CPU time (in seconds), and the average number of iterations (ie, the number of times an LP subproblem was solved). H3's performance is robust. The solutions H3 provides with different threshold values have the same solution quality and are the best among all heuristic variations and with all underlying MIP formulations. H1 and H2 have produced similar results, but 8 to 16% worse than H3 on average. This indicates the significance of incorporating structural properties of the problem into the rounding process.

Although H3 runs faster than H1, the fastest is H2. H3 found feasible solutions to the 28 problems out of the 35 test

Table 4 Relative deviation of the heuristic solution values for SP and CPU times

		H1			*H2*			*H3*		
k	$N \times T$	*DH (%)*	*CPU (s)*	*Iteration*	*DH (%)*	*CPU (s)*	*Iteration*	*DH (%)*	*CPU (s)*	*Iteration*
0.75	6 × 15	21.93	0.65	13.7	21.93	0.10	31.4	21.44	0.11	21.4
	6 × 30	70.68	2.87	17.0	68.61	0.53	19.4	47.21	0.70	37.8
	12 × 15	14.25	1.23	12.0	14.09	0.26	28.4	14.43	0.32	23.4
	Average*	35.62	1.58	14.24	34.88	0.30	26.41	27.69	0.38	27.53
	12 × 30	64.94	9.30	23.0	62.61	2.75	52.0	52.23	3.59	45.0
	24 × 15	12.66	3.12	12.8	13.94	1.01	29.6	12.86	1.27	23.6
	24 × 30	76.43	18.34	20.2	74.35	7.21	42.8	55.61	10.12	42.6
	Average	51.35	10.25	18.65	50.30	3.66	41.47	40.23	4.99	37.07
0.85	6 × 15	21.93	0.72	15.3	21.93	0.10	33.1	21.44	0.11	21.4
	6 × 30	68.61	3.51	20.8	68.61	0.56	22.6	47.21	0.75	37.8
	12 × 15	14.25	1.29	13.2	14.25	0.29	31.2	14.43	0.32	23.6
	Average*	34.93	1.84	16.43	34.93	0.32	28.98	27.69	0.40	27.59
	12 × 30	63.83	10.23	28.0	64.03	2.55	60.0	52.23	4.60	45.2
	24 × 15	12.14	3.43	14.0	13.87	1.17	32.8	12.86	1.27	24.2
	24 × 30	76.30	20.02	22.8	76.30	7.40	49.6	55.61	10.28	42.6
	Average	50.76	11.23	21.60	51.40	3.71	47.47	40.23	5.38	37.33
0.95	6 × 15	21.93	0.78	16.7	21.93	0.10	34.0	21.44	0.11	21.4
	6 × 30	68.61	3.82	23.4	68.61	0.57	24.4	47.21	0.78	37.8
	12 × 15	14.25	1.59	15.8	14.25	0.27	32.0	14.43	0.32	23.8
	Average*	34.93	2.06	18.64	34.93	0.31	30.13	27.69	0.40	27.66
	12 × 30	63.83	10.23	28.0	64.03	2.72	66.4	52.23	4.60	45.2
	24 × 15	13.78	3.89	17.0	13.78	1.18	34.8	12.86	1.27	24.4
	24 × 30	76.30	21.57	25.8	76.30	7.85	52.8	55.61	10.07	43.0
	Average	51.31	11.90	23.60	51.37	3.92	51.33	40.23	5.31	37.53
∞	6 × 15	21.93	0.79	17.0	21.93	0.10	34.3	21.44	0.11	21.4
	6 × 30	68.61	4.28	26.4	68.61	0.60	25.8	47.21	0.70	37.8
	12 × 15	14.25	1.60	16.4	14.25	0.27	32.4	14.43	0.33	23.8
	Average*	34.93	2.22	19.93	34.93	0.33	30.83	27.69	0.38	27.66
	12 × 30	55.95	9.84	26.8	64.03	3.20	67.2	52.23	3.60	45.8
	24 × 15	13.78	4.00	18.2	13.78	1.06	34.8	12.86	1.27	24.8
	24 × 30	76.30	23.29	27.2	76.30	8.23	53.2	55.61	10.04	43.2
	Average	48.68	12.37	24.07	51.37	4.16	51.73	40.23	4.97	37.93

DH (%): $100(H(\cdot)-\text{LB})/\text{LB}$ where $H(\cdot)$ denotes the solution value by the heuristic. The average results marked with '*' are computed using the optimal solution for LB, except for only one instance of 6 × 30 a lower bound is used.
Iteration: the number of times that the LP relaxation problem has been solved.

Table 5 Relative deviation of the heuristic solution values for TP and CPU times

		H1			*H2*			*H3*		
k	$N \times T$	*DH (%)*	*CPU (s)*	*Iteration*	*DH (%)*	*CPU (s)*	*Iteration*	*DH (%)*	*CPU (s)*	*Iteration*
0.75	6×15	21.20	0.62	13.6	21.20	0.15	31.5	20.83	0.18	21.4
	6×30	68.68	3.12	17.2	66.66	0.86	38.8	46.09	1.30	37.8
	12×15	13.63	1.20	12.0	12.95	0.39	28.4	13.90	0.49	23.6
	Average*	34.51	1.64	14.28	33.60	0.47	32.90	26.94	0.66	27.59
	12×30	61.82	10.60	23.8	63.12	4.12	53.6	51.79	6.23	45.4
	24×15	12.63	3.35	14.3	13.91	1.38	30.4	12.83	1.86	23.6
	24×30	76.00	17.93	20.2	73.96	8.87	42.8	55.37	10.80	42.8
	Average	50.15	10.63	19.42	50.33	4.79	42.27	40.00	6.30	37.27
0.85	6×15	21.20	0.70	15.1	21.20	0.15	33.5	20.83	0.18	21.4
	6×30	66.66	3.76	21.0	66.66	1.05	51.2	46.09	1.29	37.8
	12×15	13.82	1.40	13.4	13.82	0.43	32.8	13.90	0.49	23.8
	Average*	33.90	1.95	16.51	33.90	0.55	39.17	26.94	0.65	27.66
	12×30	64.34	11.58	28.4	64.54	4.81	67.6	51.79	6.19	45.6
	24×15	12.10	3.55	14.8	13.75	1.58	36.0	12.83	1.81	24.2
	24×30	75.91	20.08	22.6	71.49	9.74	52.0	55.37	11.38	42.8
	Average	50.78	11.74	21.92	49.93	5.38	51.87	40.00	6.46	37.53
0.95	6×15	21.20	0.77	16.4	21.20	0.16	33.5	20.83	0.17	21.4
	6×30	66.66	4.02	23.6	66.66	0.98	52.0	46.09	1.26	37.8
	12×15	13.82	1.50	15.8	13.82	0.42	32.8	13.90	0.48	24.0
	Average*	33.90	2.10	18.59	33.90	0.52	39.43	26.94	0.64	27.73
	12×30	55.58	10.21	23.8	64.54	4.73	69.6	51.79	5.72	46.0
	24×15	13.75	4.09	17.2	13.84	1.66	36.8	12.83	1.81	24.4
	24×30	71.49	22.09	25.2	71.49	9.67	54.0	55.37	11.54	43.2
	Average	46.94	12.13	22.07	49.96	5.35	53.47	40.00	6.36	37.87
∞	6×15	21.20	0.73	16.8	21.20	0.16	33.5	20.83	0.21	21.4
	6×30	66.66	4.06	25.6	66.66	0.98	51.2	46.09	1.34	37.8
	12×15	13.82	1.59	16.4	13.82	0.42	32.8	13.90	0.51	24.0
	Average*	33.90	2.13	19.58	33.90	0.52	39.17	26.94	0.68	27.73
	12×30	55.58	10.40	26.2	64.54	5.12	70.0	51.79	5.96	46.2
	24×15	12.09	4.28	18.8	13.84	1.55	36.8	12.83	1.85	24.8
	24×30	71.49	22.19	26.8	71.49	9.45	54.0	55.37	11.32	43.4
	Average	46.39	12.29	23.92	49.96	5.37	53.60	40.00	6.38	38.13

DH (%): $100(H(\cdot)-\mathrm{LB})/\mathrm{LB}$ where $H(\cdot)$ denotes the solution value by the heuristic. The average results marked with '*' are computed using the optimal solution for LB, except for only one instance of 6×30 a lower bound is used.
Iteration: the number of times that the LP relaxation problem has been solved.

problems for all variations and with all formulations. Despite the fact that the heuristics performed well in finding feasible solutions very quickly, they did not perform so well in finding near-optimal solutions. On average, the solutions found by H3 employing SP and TP (IS) were 27% (39%) worse than the optimal solution. This considerable difference in the deviations between SP (or TP) and IS highlights the importance of using tight formulations for developing quick and easy solution methods based on LP relaxations.

Alfieri *et al* (2002) have reported that the solutions found by an LP-based heuristic like H1 with $k = \infty$, 0.95 were almost optimal for the lot-sizing problems with setup costs but no setup times in their experiments. This again indicates the increased difficulty of the CLSP when setup times are considered.

Curtailed branch and bound

We performed our last experiment with the curtailed branch and bound heuristic to see the effect of optimizing the rounding process after the first LP solution. Since the other heuristics run quickly, we employed the curtailed branch and bound heuristic with time limits of 5, 15, and 30 min. Results, obtained by using the default features of CPLEX, are given in Table 7.

Table 7 shows the average percent deviation of the integer solution value from the lower bound (or the optimal solution if known), the average CPU time, the average number of nodes explored in the tree for the SP, TP, and IS formulations. Results are superior to those of the other

Table 6 Relative deviation of the heuristic solution values for IS and CPU times

		H1			H2			H3		
k	$N\times T$	*DH (%)*	*CPU (s)*	*Iteration*	*DH (%)*	*CPU (s)*	*Iteration*	*DH (%)*	*CPU (s)*	*Iteration*
0.75	6×15	26.23	0.33	18.0	28.25	0.06	41.1	24.75	0.06	24.1
	6×30	93.38	1.03	20.4	108.84	0.23	57.0	74.12	0.23	40.0
	12×15	23.25	0.81	23.8	28.23	0.15	46.4	18.49	0.15	31.2
	Average*	47.62	0.72	20.73	55.11	0.15	48.18	39.12	0.14	31.78
	12×30	111.45	2.62	30.3	96.70	0.77	71.0	91.42	0.99	53.6
	24×15	22.64	1.84	29.8	21.53	0.56	71.5	18.09	0.59	47.0
	24×30	125.08	6.28	38.6	125.10	2.55	79.6	86.14	2.96	62.0
	Average	86.39	3.58	32.87	81.11	1.29	74.03	65.22	1.51	54.20
0.85	6×15	26.23	0.33	18.9	28.25	0.06	41.1	24.75	0.06	24.1
	6×30	93.38	0.84	23.4	108.84	0.23	57.0	74.12	0.37	40.0
	12×15	23.25	0.73	24.4	28.23	0.14	46.4	18.49	0.20	31.2
	Average*	47.62	0.63	22.23	55.11	0.14	48.18	39.12	0.21	31.78
	12×30	111.17	2.45	33.0	96.70	0.81	71.0	91.42	0.93	53.6
	24×15	21.65	2.08	35.3	21.53	0.72	71.5	18.09	0.62	47.0
	24×30	125.18	5.85	34.2	125.10	2.57	79.6	86.14	3.13	62.0
	Average	86.00	3.46	34.15	81.11	1.37	74.03	65.22	1.56	54.20
0.95	6×15	24.61	0.35	19.8	28.25	0.07	41.1	24.75	0.05	24.1
	6×30	93.38	0.94	26.8	108.84	0.24	57.0	74.12	0.22	40.0
	12×15	23.25	0.80	25.2	28.23	0.16	46.4	18.49	0.15	31.2
	Average*	47.08	0.69	23.92	55.11	0.16	48.18	39.12	0.14	31.78
	12×30	111.17	2.71	36.0	96.70	0.80	71.0	91.42	0.89	53.6
	24×15	21.65	2.06	35.0	21.53	0.53	71.5	18.09	0.58	47.0
	24×30	125.19	6.28	36.8	125.10	2.66	79.6	86.14	2.71	62.0
	Average	86.00	3.68	35.93	81.11	1.33	74.03	65.22	1.39	54.20
∞	6×15	26.23	0.33	18.0	28.25	0.06	41.1	24.75	0.06	24.1
	6×30	93.38	1.03	20.4	108.84	0.22	57.0	74.12	0.22	40.0
	12×15	23.25	0.81	23.8	28.23	0.15	46.4	18.49	0.15	31.2
	Average*	47.62	0.72	20.73	55.11	0.14	48.18	39.12	0.14	31.78
	12×30	111.45	2.62	30.3	96.70	0.77	71.0	91.42	0.88	53.6
	24×15	22.64	1.84	29.8	21.53	0.53	71.5	18.09	0.58	47.0
	24×30	125.08	6.28	38.6	125.10	2.39	79.6	86.14	2.73	62.0
	Average	86.39	3.58	32.87	81.11	1.23	74.03	65.22	1.40	54.20

DH (%): $100(H(\cdot)-\mathrm{LB})/\mathrm{LB}$ where $H(\cdot)$ denotes the solution value by the heuristic. The average results marked with '*' are computed using the optimal solution for LB, except for only one instance of 6×30 a lower bound is used.
Iteration: the number of times that the LP relaxation problem has been solved.

LP-based heuristics for all formulations and comparable with the results given in Table 2 for SP and TP. While a standard branch and bound implementation with 3 h time limit provided an integrality gap of about 30% for $N=12$ and $T=30$, 5% for $N=24$ and $T=15$, and 40% for $N=24$ and $T=30$ on average for TP, the curtailed branch and bound resulted in average gaps of about 32% for $N=12$ and $T=30$, 8% for $N=24$ and $T=15$, and 38% for $N=24$ and $T=30$ in 15 min with the TP formulation. The curtailed branch and bound solutions deviate by only 8 and 7% from the optimal solutions for SP and TP, respectively. However, increasing time limit further did not improve the solution quality. The result that SP and TP outperform IS support our argument on the importance of using tighter formulations in devising efficient solution methods to solve the problem.

Tighter problem instances

Considering that feasibility is a difficult issue for the lot-sizing problem with setup times, the results we obtained indicate a very good performance of the LP-based heuristics. Since finding a feasible solution is NP-complete, this might raise questions regarding the tightness of the test problems in terms of resource capacities. To understand how tight the test problem instances are we expanded our test bed by modifying the test problems with $N=12$, 24 and $T=15{,}30$. In the first new set of 20 problems, we decreased the available resource capacity in each period by 10%. We also created two other new sets of 20 problems by increasing the original setup time for each item by 10 and 5 units separately, keeping the original resource capacities intact. This made a total of 60 new test problems.

Table 7 Relative deviation of the curtailed branch and bound solution values by SP, TP and IS and CPU times, using CPLEX with all features on

		Time limit = 5 min			*Time limit = 15 min*			*Time limit = 30 min*		
MIP	$N \times T$	*D (%)*	*Node*	*CPU (min)*	*D (%)*	*Node*	*CPU (min)*	*D (%)*	*Node*	*CPU (min)*
SP	6×15	7.71	393.4	0.06	†			†		
	6×30	11.48	4829.6	2.14						
	12×15	4.86	1765.6	0.55						
	Average*	8.02	2329.5	0.92						
	12×30	34.45	2433.6	5.02	31.85	9144.6	15.01	31.66	20 248.6	30.01
	24×15	7.59	4662.0	3.46	7.57	10 785.4	7.52	7.52	19 602.4	12.45
	24×30	39.43	537.2	4.39	39.41	2342.6	12.37	38.04	5340.2	24.37
	Average	27.16	2544.3	4.29	26.28	7424.2	11.63	25.74	15 063.7	22.28
TP	6×15	5.87	288.6	0.07	†					
	6×30	11.58	1190.0	1.36	10.74	5321.6	4.70	†		
	12×15	4.76	1707.6	1.39	†					
	Average*	7.40	1062.1	0.94	7.12	2439.3	2.06			
	12×30	34.48	1286.4	5.03	31.83	4133.2	15.03	31.26	9695.4	30.03
	24×15	7.57	2992.8	4.40	7.54	6800.2	9.37	7.52	9843.2	15.83
	24×30	39.03	388.8	4.21	38.20	1192.2	12.25	38.10	3601.6	24.21
	Average	27.03	1556.0	4.55	25.86	4041.87	12.22	25.63	7713.4	23.36
IS	6×15	7.03	155.2	0.01	†			†		
	6×30	24.68	4218.4	0.77						
	12×15	6.33	1150.4	0.02						
	Average*	12.68	1841.3	0.33						
	12×30	52.83	8496.4	5.00	51.58	23 418.6	12.46	51.05	39 986.6	21.50
	24×15	12.33	8515.8	3.73	12.33	12 602.4	5.84	12.33	18 604.6	8.83
	24×30	61.32	1502.0	3.62	59.43	3806.4	9.62	59.37	7749.2	18.62
	Average	42.16	6171.4	4.12	41.11	13 275.8	9.31	40.91	22 113.5	16.32

D (%): $100(U(\cdot)-\text{LB})/\text{LB}$. The average results marked with '*' are computed using the optimal solution for LB, except for only one instance of 6×30 a lower bound is used.
Node: the number of nodes explored.
†The algorithm has terminated before the 5-min limit.

We used only the LP relaxation of SP to solve the new problems.

When we reduced the original capacity in each period by 10%, even the LP relaxations of seven of the 20 problems became infeasible. For another two problems, CPLEX could not find a feasible integer solution in 1 h. For the remaining 11 feasible instances, the curtailed branch and bound heuristic produced an average gap of about 8% in 30 min for eight instances whereas H3 provided a solution for only three instances.

When we increased the original setup time for each item by 10 units, CPLEX confirmed the infeasibility of two out of 20 instances. The curtailed branch and bound heuristic found a feasible solution, in 5 min, to 14 out of 18 instances with an average gap of about 9%. In the case of 5 units increase in setup times, one problem became infeasible and the curtailed branch and bound heuristic found a feasible solution to 19 instances with about 12% average gap. Considering that the average gaps are the deviations of the integer solution values from the lower bounds, the performance of H3 was also good for these new problem instances. H3 provided a feasible solution, in less than 3 s, to 12 instances with about 15% average gap for the problems with 10 units increase in setup times, and to 15 instances with about 18% average gap for those with 5 units increase.

These results we obtained for the tighter problem instances indicate that original problems are not loose in terms of the resource capacity.

Conclusion and further research

The multi-item capacitated lot-sizing problem with setup times that we address in this research is known to be very challenging from a computational viewpoint. Inclusion of setup times makes even the feasibility problem NP-complete. The alternative formulations we considered led to considerably different solution values when their linear relaxations were solved. This is not observed when setup times are not included or when capacity can be relaxed by overtime decisions. This suggests that it is even more important, in this case, to start with the right formulation in any attempt to develop an efficient solution procedure for the problem or to find a reliable solution using a commercial MIP solver. Our computational results suggest that, in an LP-based approach, the use of the SP and TP formulations within a general purpose MIP solver employing the dual algorithm would be relatively efficient. Besides, as an alternative to SP and TP, the use of the IS formulation can be advised within a general purpose MIP solver that generates cutting planes, again with the dual algorithm.

Among the LP-based heuristics that we developed, H3, which makes use of the problem structure and prioritizes selection with respect to larger time indices turned out to be the best and the most robust. In general, our heuristics were successful in finding a feasible integer solution to the problem in negligible computation times for the test problems we solved, however, the optimality gap turned out to be rather large. One important observation was that in the first LP solution about 75–79% of variables take on positive integer values. After that all heuristic efforts are for fixing the remaining 20–25%. The curtailed branch and bound algorithm was considered to fix them in an optimal way. This approach improved the quality of the LP-based heuristic solutions significantly by 13–26%, with very reasonable computation times. In general, the underlying formulations SP and TP performed better than IS for our LP-based heuristic approaches. Based on these results, we propose that LP-based heuristics should be considered in practical settings and further explored. Furthermore, as our analysis suggests more research should be done for investigating different MIP formulations, since they lead to significantly different results when their relaxations are considered or when they are solved by different MIP solvers, especially in this case.

References

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

Bahl HC and Zionts S (1987). Multi-item scheduling by Benders' decomposition. *J Opl Res Soc* **38**: 1141–1148.

Barany I, Van Roy TJ and Wolsey LA (1984). Uncapacitated lot sizing: the convex hull of solutions. *Math Program Stud* **22**: 32–43.

Diaby M, Bahl HC, Karwan MH and Zionts S (1992). Capacitated lot-sizing and scheduling by Lagrangean Relaxation. *Eur J Opl Res* **59**: 444–458.

Eppen GD and Martin RK (1987). Solving multi-item capacitated lot-sizing problems using variable redefinition. *Opns Res* **35**: 832–848.

Karayel MN (1984). *Dual-based heuristics for capacity constrained production scheduling*. PhD thesis, University of California, Berkeley.

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

Maes J, McClain JO and Van Wassenhove LN (1991). Multilevel capacitated lot-sizing complexity and LP-based heuristics. *Eur J Opl Res* **53**: 131–148.

Stadtler H (1996). Mixed integer programming model formulations for dynamic multi-item multi-level capacitated lot-sizing. *Eur J Opl Res* **94**: 561–581.

Süral H (1996). *Multi-item lot-sizing with setup times*. PhD thesis, Middle East Technical University, Ankara.

Trigeiro WW, Thomas LJ and McClain JO (1989). Capacitated lot-sizing with setup times. *Mngt Sci* **35**(3): 353–366.