"Solving multi-item lot-sizing problems with an MIP solver using classification and reformulation"

Wolsey, Laurence

Abstract
Based on research on the polyhedral structure of lot-sizing models over the last twenty years, we claim that there is a nontrivial fraction of practical lot-sizing problems that can now be solved by nonspecialists just by taking an appropriate a priori reformulation of the problem, and then feeding the resulting formulation into a commercial mixed integer programming solver. This claim uses the fact that many multi-item problems decompose naturally into a set of single-item problems with linking constraints, and that there is now a large body of knowledge about single-item problems. To put this knowledge to use, we propose a classification of lot-sizing problems (in large part single-item), and then indicate in a set of Tables what is known about a particular problem class, and how useful it might be. Specifically we indicate for each class i) whether a tight extended formulation is known, and its size, ii) whether one or more families of valid inequalities are known defining the con...


Solving Multi-Item Lot-Sizing Problems with an MIP Solver using Classification and Reformulation

Laurence A. Wolsey

April 4, 2002

Abstract

Based on research on the polyhedral structure of lot-sizing models over the last twenty years, we claim that there is a nontrivial fraction of practical lot-sizing problems that can now be solved by nonspecialists just by taking an appropriate a priori reformulation of the problem, and then feeding the resulting formulation into a commercial mixed integer programming solver.

This claim uses the fact that many multi-item problems decompose naturally into a set of single-item problems with linking constraints, and that there is now a large body of knowledge about single-item problems. To put this knowledge to use, we propose a classification of lot-sizing problems (in large part single-item), and then indicate in a set of Tables what is known about a particular problem class, and how useful it might be. Specifically we indicate for each class i) whether a tight extended formulation is known, and its size, ii) whether one or more families of valid inequalities are known defining the convex hull of solutions, and the complexity of the corresponding separation algorithms, and iii) the complexity of the corresponding optimization algorithms (which would be useful if a column generation or Lagrangian relaxation approach was envisaged).

1CORE and INMA, Université Catholique de Louvain, Belgium

This paper presents research results of the Belgian Program on Interuniversity Poles of Attraction initiated by the Belgian State, Prime Minister’s Office, Science Policy Programming. The scientific responsibility is assumed by the authors.

This research was also supported by the European Commission GROWTH Programme, Research Project LISCOS, Large Scale Integrated Supply Chain Optimization Software Based on Branch-and-Cut and Constraint Programming Methods, Contract No. GRDI-1999-10056.
Three distinct multi-item lot-sizing instances are then presented to demonstrate the approach, and comparative computational results are presented. Finally we also use the classification to point out what appear to be some of the important open questions and challenges.

Keywords: Lot-sizing, Production Planning, Classification, Convex Hull, Extended Formulation, Mixed Integer Programming
1 Introduction

Production planning problems involving lot-sizing have been an area of active research since the seminal paper of Wagner-Whitin [56] in 1958. Work on the polyhedral structure of the uncapacitated problem started with Barany et al. [5] and on extended formulation with Bilde and Krarup [22] and Eppen and Martin [15]. Since then there has been a considerable amount of research extending these results for the single item problem to incorporate other important features such as backlogging, start-ups, constant and varying capacities, etc. See Pochet and Wolsey [40] for a survey, and Pochet [35] and Wolsey [58] for two recent tutorials. On the other hand although almost all practical problems are multi-item, and also often multi-machine and multi-level, the polyhedral results concerning such models are limited. See [12, 21, 30] for some notable exceptions. As a result the approach of choice in solving such problems has been implicitly or explicitly some form of decomposition, namely the development of solution methods, such as Lagrangian relaxation, column generation or branch-and-cut, that explicitly use algorithms for optimization, or for separation of single item problems.

In two recent papers we have described ways to formulate certain constraints that arise in practical lot-sizing models and thereby improve solution times [7], and presented a special purpose modelling and branch-and-cut system BC-PROD designed for lot-sizing problems [6]. Here we would like to suggest that, based on the research cited above and the progress of commercial MIP systems, certain multi-item lot-sizing problems can now be solved just using standard reformulations and an off-the shelf MIP solver. To achieve this we present a simple classification of single-item lot-sizing problems, and then indicate in the form of Tables our present knowledge about such problems. This knowledge consists of extended formulations, families of valid inequalities that provide or approximate the convex hull of solutions, and separation algorithms allowing one to use the valid inequalities as cutting planes, along with their complexity. This is the knowledge typically needed when solving the problems directly as MIPs using branch-and-cut, the approach favoured here. For those interested in developing column generation or Lagrangian relaxation approaches, the Tables also indicate the complexity of optimization and give references. We then indicate a few of the characteristics of multi-item problems
for which useful modelling results are available, and finally we show by three examples how the classification and the corresponding reformulations can be used to obtain guaranteed high quality solutions using nothing but a basic MIP system. Earlier classification schemes can be found in [8] and [23]. The former is mostly concerned with the varying capacity single-item problem, and in distinguishing which special cost structures lead to polynomial variants, and the latter considers very general batching and scheduling problems.

The outline of the paper is as follows. In Section 2 we present a brief description of three multi-item problems. Just from these descriptions, we obtain a first verbal classification as an indication of what needs to be classified formally. In Section 3 we present the single-item classification that we have found useful. In Section 4 we present Tables indicating the status of the most important problems concerning
i) families of valid inequalities, whether they describe the convex hull, and the complexity of the separation problem for these families of inequalities
ii) the existence of tight, or “good” extended formulations giving the convex hull exactly or approximately
iii) the complexity of optimization.

In Section 5 we extend the classification to some aspects of multi-item problems and discuss briefly the important results available. In Section 6 we show how the Classification and Tables of Sections 3 and 4 can be used to obtain effective formulations in practice, giving computational results for the three multi-item problems presented earlier. Finally in Section 7 we indicate several important open problems.

2 Three Multi-Item Problems

Here we take the description of three multi-item lot-sizing problems, and use the description to derive a verbal classification of each problem, suggesting what will be the important points in the formal classification presented later. In Section 6 we will translate these verbal classifications into our formal scheme, and use this to reformulate and solve one or more instances of each problem.
**Problem 1**  
This is a bottling line problem with a 30 day planning horizon. There are four products. The line is available 16 hours per day, and only one product can be produced per day. There are storage, set-up and start-up costs, which are all constant over time. The minimum production per day is 7 hours.

Classification.  
i) Multi-item constraints and costs. At most one item can be produced per period.  
ii) Individual item constraints and costs. When produced, each item is produced for between 7 and 16 hours, so both the upper bound and the lower bounds on production per period are time invariant. Also the unit production and storage costs are time invariant, and there are start-up costs.

**Problem 2**  
This is a lot-sizing problem with ten items with sequence-dependent changeover costs and storage costs studied by Fleischman [18]. Production is at full capacity, and at most one item is produced per period.

Classification.  
i) Multi-item constraints and costs. At most one item can be produced per period, and there are sequence dependent set-up costs.  
ii) Individual item constraints and costs. Production is all or nothing with constant capacities. There are no unit production costs, and storage costs are nonnegative and constant over time.

**Problem 3**  
This is a general model for multilevel problems with assembly product structure proposed in [42], involving product families consisting of one or more items, where each family can in turn have a fixed cost, a set-up time or a resource constraint associated with it. Instances of this problem come from the construction of bottle racks and the production of animal feed. Instances of this problem have been tackled earlier with the
special purpose systems bc-prod [6] and bc-opt [13].

Classification.
i) Multilevel structure. Assembly type product structure.
ii) Multi-item constraints and costs. Many items can be produced in each period, and the capacity constraints limiting production in each period involve both production levels and set-up times for families.
iii) Individual item constraints and costs. There are no individual capacity constraints, but there are storage costs and implicit fixed costs through the families.

3 Single-Item Classification

We start by defining the basic lot-sizing problem (LS). There is a time horizon of \( n \) periods, and in each period there is a demand to be satisfied \( d_t \), and a production limit \( C_t \). There is a per unit production cost \( p_t \), a fixed set-up cost \( f_t \) if production takes place in \( t \) for \( t = 1, \ldots, n \), and a cost \( h'_t \) per unit of stock at the end of period \( t \) for \( t = 0, \ldots, n \). Note that in principle a variable amount of initial stock is allowed.

3.1 The Basic Classification

There are three fields \( PROB - CAP - VAR \). We use \([x, y, z]\) to denote exactly one element from the set \([x, y, x]\), and \([x, y, z]^*\) to denote any subset of \([x, y, x]\). Fields that are empty are dropped.

In the first field \( PROB \), there is a choice of four problem versions \([LS, WW, DLSI, DLS]\)

\( LS \): (Lot-Sizing) This is the general problem defined above.

\( WW \): (Wagner-Whitin) This is problem \( LS \), except that the variable production and storage costs satisfy \( h_t = h'_t + p_t - p_{t+1} \geq 0 \) for \( t = 0, \ldots, n - 1 \).

\( DLSI \): (Discrete Lot-Sizing with Variable Initial Stock) This is problem \( LS \) with the
restriction that there is either no production or production at full capacity $C_t$ in each period $t$.

**DLS**: (Discrete Lot-Sizing) This is problem $DLSI$ without an initial stock variable.

The second field $CAP$ concerns the production limits or capacities $[C, CC, U]$.

**PROB – C**: (Capacitated) Here the capacities $C_t$ vary over time.

**PROB – CC**: (Constant Capacity) This is the case where $C_t = C$, a constant, for all $t$.

**PROB – U**: (Uncapacitated) This is the case when there is no limit on the amount produced in each period, i.e. $C_t$ exceeds the sum of all present and future demand.

Before presenting the third parameter involving the many possible extensions, we now present mixed integer programming formulations of the four basic variants with varying capacities $PROB – C$.

### 3.2 Formulations

The standard formulation of $LS$ as a mixed integer program involves the variables

$$x_t$$ the amount produced in period $t$ for $t = 1, \ldots, n$,

$$s_t$$ the stock at the end of period $t$ for $t = 0, \ldots, n$, and

$$y_t = 1$$ if the machine is set-up to produce in period $t$, and $y_t = 0$ otherwise.

We also use the notation $d_{kt} \equiv \sum_{u=k}^{t} d_u$ throughout.
LS – C now has the formulation

\[
\begin{align*}
\min & \sum_{t=1}^{n} p_t x_t + \sum_{t=0}^{n} h_t' s_t + \sum_{t=1}^{n} f_t y_t \\
\text{s.t.} \quad & x_{t-1} + x_t = d_t + s_t \quad \text{for } t = 1, \ldots, n \\
& x_t \leq C_t y_t \quad \text{for } t = 1, \ldots, n \\
& x \in \mathbb{R}^n_+, s \in \mathbb{R}^{n+1}_+, y \in \{0, 1\}^n.
\end{align*}
\]

WW – C can be formulated just in the space of the \(s, y\) variables.

\[
\begin{align*}
\min & \sum_{t=0}^{n} h_t s_t + \sum_{t=1}^{n} f_t y_t \\
\text{s.t.} \quad & s_{k-1} + \sum_{u=k}^{t} C_u y_u \geq d_{kt} \quad \text{for } 1 \leq k \leq t \leq n \\
& s \in \mathbb{R}^{n+1}_+, y \in \{0, 1\}^n.
\end{align*}
\]

To derive this formulation, one first uses (2) to eliminate \(x_t\) from the objective function (1). To within a constant, the resulting objective function is

\[
\sum_{t=0}^{n} (h'_t + p_t - p_{t+1}) s_t + \sum_{t=1}^{n} f_t y_t = \sum_{t=0}^{n} h_t s_t + \sum_{t=1}^{n} f_t y_t.
\]

Then as \(h_t \geq 0\) for all \(t\), it follows that once the set-up periods are fixed, the stocks will be as low as possible compatible with satisfying the demand. Thus

\[
s_{k-1} = \max(0, \max_{t=k, \ldots, n} [d_{kt} - \sum_{u=k}^{t} C_u y_u]),
\]

see [39]. It follows that the proposed formulation is valid, though its \((s, y)\) feasible region is not the same as that of \(LS – C\). Specifically \((s, y)\) is feasible in (6)-(7) if and only if there exists \((x, s', y)\) feasible in (2)-(4) with \(s' \leq s\).

DLSI – C can be formulated by adding \(x_t = C_t y_t\) in the formulation of \(LS – C\). However after elimination of the variables \(s_t = \sum_{u=1}^{t} x_u - d_{1t} \geq 0\) and \(x_t = C_t y_t\), we obtain an equivalent formulation just in the space of the \(s_0\) and the \(y\) variables.

\[
\begin{align*}
\min & \ h_0 s_0 + \sum_{t=1}^{n} f'_t y_t \\
\text{s.t.} \quad & s_0 + \sum_{u=1}^{t} C_u y_u \geq d_{1t} \quad \text{for } 1 \leq t \leq n \\
& s_0 \in \mathbb{R}_+^1, y \in \{0, 1\}^n.
\end{align*}
\]
DLS – C can be formulated just in the space of the $y$ variables.

\[
\begin{align*}
\min & \sum_{t=1}^{n} f'_t y_t \\
\sum_{u=1}^{t'} C_u y_u & \geq d_{1t} \text{ for all } 1 \leq t \leq n \\
y & \in \{0, 1\}^n.
\end{align*}
\]

Without introducing a new problem class, we say that DLS has Wagner-Whitin costs if $f'_t \geq f'_{t+1}$ for all $t$.

3.3 Complexity

Observation 1. All eight constant or uncapacitated instances $PROB - [CC, U]$ are polynomially solvable. The dynamic programming algorithm of Florian and Klein [19] solves $LS - CC$ and the other seven problems can all be seen as special cases.

Observation 2. All four varying capacity instances $PROB - C$ are NP-hard. All four problems are polynomially reducible to the 0-1 knapsack problem, see [8].

The above imply that we can only reasonably hope to have complete convex hull descriptions, and/or tight reformulations when $CAP$ is selected from $[U, CC]$.

We now consider the relationships between the different problems.

**Notation.** We let $X_{PROB - CAP}$ denote the feasible region of $PROB - CAP$ as formulated in Section 2.2 in the corresponding space of variables.

\[\text{proj}_w(Y)\] denotes the projection of the solution set $Y$ onto the space of variables $w$.

\[X_{DLS - C}^k = \{(s, y) \in R_{+}^{n+1} \times [0, 1]^n : s_{k-1} + \sum_{u=k}^{t'} C_u y_u \geq d_{kt} \text{ for } t = k, \ldots, n\}.\]

The following proposition states more formally the links between the different formulations introduced in the previous subsection.

**Proposition 1** i) $\text{proj}_{s,y} X_{LS - C} \subseteq X_{WW - C}^{W}$
ii) \( \text{proj}_{s_0, y} X^{WW-C} = X^{DLSI-C} \)

iii) \( X^{WW-C} = \bigcap_{k=1}^{n} X^{DLSI-C}_k \) with \( X^{DLSI-C}_1 = X^{DLSI-C} \)

iv) \( X^{LS-C} \subseteq X^{LS-CC} \subseteq X^{LS-U} \) if we take \( \max_t C_t \) as the constant capacity.

On the other hand it is clear that in the \((x, s, y)\) space, DLSI is a restriction of LS.

**Corollary.** Every valid inequality for \( WW-CAP \) in \((s, y)\) variables is valid for \( LS-CAP \), and every valid inequality for \( DLSI-CAP \) in \((s_0, y)\) variables is valid for \( WW-CAP \). Also every valid inequality for \( PROB-U \) is valid for \( PROB-[C, CC] \).

### 3.4 Extensions

The third field \( \text{VAR} \) concerns extensions/variants \([B, SC, ST, LB, SL, SS]^*\) to one of the twelve problems \( PROB-CAP \) considered so far.

**B:** (Backlogging) Demand must be satisfied, but the items can be produced later than requested. The cumulated shortfall \( \max\{0, d_t - s_0 - \sum_{j=1}^t x_j\} \) in satisfaction of the demand in period \( t \) is charged at a cost of \( b_t \) per unit.

**SC:** (Start-Up Costs) If a sequence of set-ups starts in period \( t \), a start-up cost \( g_t \) is incurred.

**ST:** (Start-Up Times) If a sequence of set-ups starts in period \( t \), the capacity \( C_t \) is reduced by an amount \( ST_t \). (\( ST(C) \)) for constant start-up times.

**LB:** (Minimum Production Levels) If production takes place in period \( t \), a minimum amount \( LB_t \) must be produced. (\( LB(C) \)) denotes constant lower bounds.

**SL:** (Sales) In addition to the demand \( d_t \) that must be satisfied in each period, an additional amount up to \( u_t \) can be sold at a unit price of \( c_t \).
SS: (Safety Stocks) There is a lower bound $S_t$ on the stock level at the end of period $t$.

Now we have the three fields that describe a single item lot-sizing problem

$$[LS, WW, DLSI, DLS] - [C, CC, U] - [B, SC, ST(C), SL, LB, LB(C), SS]^*$$

where one entry is required from each of the first two fields, and any number of entries from the third.

**Example 1**

i) $WW-U-\emptyset$ (or just $WW-U$) denotes the uncapacitated Wagner-Whitin problem.

ii) $DLSI-CC-\{B-ST\}$ denotes the constant capacity discrete lot-sizing problem with initial stock variable, backlogging and start-up times.

Again we observe that the variants are still polynomially solvable in versions $PROB-[CC,U]-VAR$ provided that the start-up times or lower bounds, if any, are constant (versions $ST(C), LB(C)$).

### 4 Knowledge about $PROB-CAP-VAR$

In this section we catalogue our state of knowledge about the most important polynomially solvable variants. Specifically we present three tables for $PROB-[U,CC]$, $PROB-[U,CC]-B$ and $PROB-[U,CC]-SC$ respectively. We also indicate the relatively few results known for more complicated variants.

For each problem $PROB-CAP-VAR$ we present a Table with three parts. The first part FORMULATION deals with extended formulations whose projection is the convex hull of $X^{PROB-CAP-VAR}$. First some indication of the name of the reformulation (if any) is given, along with the number of constraints and variables in the formulation, and then references. The second part VALID INEQUALITIES and SEPARATION gives the family of valid inequalities describing the convex hull, the complexity of their separation,
and references. The third OPTIMIZATION gives the complexity of the best known algorithm, and references. An asterisk * indicates that the family of inequalities only gives a partial description of the convex hull of solutions. A triple asterisk indicates that we do not know of any result specific to the particular problem class.

4.1 $PROB - [U, CC]$

Table 1 contains results for $PROB - [U, CC]$. The cases $[DLSI, DLS] - U$ have been left blank as the results and algorithms are trivial.

<table>
<thead>
<tr>
<th>FORMULATION</th>
<th>LS</th>
<th>WW</th>
<th>DLSI</th>
<th>DLS</th>
</tr>
</thead>
<tbody>
<tr>
<td>$U$</td>
<td>$SP \ O(n) \times O(n^2)$</td>
<td>$WW \ O(n^2) \times O(n)$</td>
<td>$-$</td>
<td>$-$</td>
</tr>
<tr>
<td></td>
<td>$FL \ O(n^2) \times O(n^2)$</td>
<td>$39$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$CC$</td>
<td>$O(n^3) \times O(n^3)$</td>
<td>$O(n^2) \times O(n^2)$</td>
<td>$O(n) \times O(n)$</td>
<td>$O(n) \times O(n)$</td>
</tr>
<tr>
<td></td>
<td>$53$</td>
<td>$39$</td>
<td>$32, 39$</td>
<td>Folklore</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SEPARATION</th>
<th>LS</th>
<th>WW</th>
<th>DLSI</th>
<th>DLS</th>
</tr>
</thead>
<tbody>
<tr>
<td>$U$</td>
<td>$(l, S)$</td>
<td>$(l, S)(WW)$</td>
<td>$-$</td>
<td>$-$</td>
</tr>
<tr>
<td></td>
<td>$O(n \log n)$</td>
<td>$O(n)$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$[5]$</td>
<td>$39$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$CC$</td>
<td>$klSI^*$</td>
<td>$klSI(WW)$</td>
<td>Mixing</td>
<td>Gomory</td>
</tr>
<tr>
<td></td>
<td>$-$</td>
<td>$O(n^2 \log n)$</td>
<td>$O(n \log n)$</td>
<td>Folklore</td>
</tr>
<tr>
<td></td>
<td>$[38]$</td>
<td>$39$</td>
<td>$20, 32, 39$</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>OPTIMIZATION</th>
<th>LS</th>
<th>WW</th>
<th>DLSI</th>
<th>DLS</th>
</tr>
</thead>
<tbody>
<tr>
<td>$U$</td>
<td>$O(n \log n)$</td>
<td>$O(n)$</td>
<td>$-$</td>
<td>$-$</td>
</tr>
<tr>
<td></td>
<td>$[1, 16, 55]$</td>
<td>$[1, 16, 55]$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$CC$</td>
<td>$O(n^3)$</td>
<td>$O(n^2 \log n)$</td>
<td>$O(n \log n)$</td>
<td>$O(n \log n)$</td>
</tr>
<tr>
<td></td>
<td>$[19, 50]$</td>
<td>$[54]$</td>
<td>$[54]$</td>
<td>$[54]$</td>
</tr>
</tbody>
</table>

Table 1: Models $PROB - [U, CC]$
Remarks concerning Table 1.

FL denotes the facility location reformulation from [22].

SP denotes the shortest path reformulation from [15].

(l, S) denotes the (l, S)-inequalities derived in [5].

(l, S)(WW) denotes the subclass of (l, S)-inequalities needed for Wagner-Whitin costs in [39].

klSI denotes the klSI-inequalities derived in [38]. A heuristic separation algorithm can be devised for this class based on that for the subclass klSI(WW).

klSI(WW) denotes a restricted subclass of klSI-inequalities, see [39].

Here mixing denotes essentially the klSI(WW)-inequalities, see [20].

Gomory indicates that Gomory fractional cuts give a tight $O(n) \times O(n)$ formulation for DLS − CC. The basic algorithm for LS − CC, due to Florian and Klein [19], was an $O(n^4)$ algorithm based on a shortest path over regeneration intervals. This algorithm extends easily to LS − CC − B and also LS − CC − SC. For LS − CC Van Hoesel and Wagelmans [50] show how the costs of the regeneration intervals can be calculated more efficiently, leading to an $O(n^3)$ implementation.

Varying Capacities: Valid Inequalities and Separation In [34] it is shown how flow cover inequalities [36] can be used to derive a class of valid inequalities for LS − C. Recently a dynamic knapsack model has been studied [25, 26, 28] leading to new families of valid inequalities for DLSI − C, WW − C and LS − C, as well as a separation heuristic. A fully polynomial approximation scheme is given in [51].

We now consider what results are known for the most important variants, in particular those with backlogging and start-up costs respectively.

4.2 Backlogging PROB − [U, CC] − B

The basic formulation for LS − C − B has as additional data $b_t'$ the per unit cost of backlogging demand in period t. Its formulation requires the introduction of new variables $r_t$ is the amount backlogged at the end of period $t$ for $t = 1, \ldots, n$. 13
It is assumed throughout that $r_0$ is undefined, or equivalently that $r_0 = 0$.

$LSCB$ now has the formulation

\[
\begin{align*}
\min & \sum_{t=0}^n h'_t s_t + \sum_{t=1}^n b'_t r_t + \sum_{t=1}^n p_t x_t + \sum_{t=1}^n f_t y_t \\
& s_{t-1} - r_{t-1} + x_t = d_t + s_t - r_t \quad \text{for } t = 1, \ldots, n \\
& x_t \leq C_t y_t \quad \text{for } t = 1, \ldots, n \\
& x, r \in R^n, s \in R^{n+1}, y \in \{0, 1\}^n.
\end{align*}
\]  

(WW $-$ $C$ $-$ $B$). With backlogging, the costs are said to be Wagner-Whitin if both $h_{t-1} = p_{t-1} + h'_{t-1} - p_t \geq 0$ and $b_t = p_{t+1} + b'_t - p_t \geq 0$ for all $t$. However it is not known if there is a simple formulation similar to that of WW $-$ $C$ involving just the $s, r, y$ variables.

$DLSI - C - B$ has the formulation in the $(s, r, y)$ space

\[
\begin{align*}
& s_0 + \sum_{u=1}^t C_u y_u = d_{1t} + s_t - r_t \quad \text{for } t = 1, \ldots, n \\
& s \in R^{n+1}, r \in R^n, y \in [0, 1]^n.
\end{align*}
\]

Now the variables $r_1, \ldots, r_n$ (or alternatively $s_1, \ldots, s_n$) can be eliminated, giving the feasible region

\[
\begin{align*}
& s_0 + r_t + \sum_{u=1}^t C_u y_u \geq d_{1t} \quad \text{for } t = 1, \ldots, n \\
& s_0 \in R^1, r \in R^n, y \in [0, 1]^n.
\end{align*}
\]

$DLS - C - B$ is obtained from $DLSI - C - B$ by setting $s_0 = 0$.

The results for $PROB - [U, CC] - B$ are given in Table 2.

Remarks concerning Table 2.

$SP$ and $FL$ are again shortest path and facility location like formulations.
RI indicates a formulation based on regeneration intervals. Ext(l, S) indicates a large family of inequalities including the Cycle inequalities (giving conv(X^WW−U−B)), which are in turn a generalization of the (l, S) inequalities. A simple separation heuristic involves adding backlog variables to (l, S) inequalities so as to make them feasible for LS−U−B.

Cycle inequalities can be separated by finding a negative cost cycle in an appropriate graph.

In similar fashion Ext(klSI) is the family of klSI inequalities extended to be valid for LS−CC−B.

FC denotes flow-cover inequalities, RC reduced capacity inequalities, GMIX denotes mixing inequalities made feasible by the addition of appropriate backlog variables, and MIR denotes mixed integer rounding inequalities.
4.3 Start-Up Costs (SC)

The basic formulation for $LS - C - SC$ has as additional data the start-up costs $g_t$ for $t = 1, \ldots, n$. It requires the introduction of new variables $z_t$:

\[ z_t = \begin{cases} 1 & \text{if there is a start-up in period } t, \text{ i.e. there is a set-up in period } t, \text{ but there was not in period } t - 1, \text{ and } z_t = 0 \text{ otherwise.} \\ \end{cases} \]

The resulting formulation is

\[
\begin{align*}
\min & \sum_{t=1}^{n} p_t x_t + \sum_{t=0}^{n} h_t s_t + \sum_{t=1}^{n} f_t y_t + \sum_{t=1}^{n} g_t z_t \\
& s_{t-1} + x_t = d_t + s_t \text{ for } t = 1, \ldots, n \\
& x_t \leq C_t y_t \text{ for } t = 1, \ldots, n \\
& z_t \geq y_t - y_{t-1} \text{ for } t = 1, \ldots, n \\
& z_t \leq y_t \text{ for } t = 1, \ldots, n \\
& z_t \leq 1 - y_{t-1} \text{ for } t = 1, \ldots, n \\
& x \in \mathbb{R}^n_+, s \in \mathbb{R}^{n+1}_+, y, z \in \{0, 1\}^n.
\end{align*}
\]

where we assume that $y_0$, the state of the machine at time 0, is given as data.

The formulations of $[WW, DLSI, DLS] - C - SC$ are obtained by just adding the constraints (21)-(23) and $z \in \{0, 1\}^n$ to the earlier formulations given in Section 2.

The results for $PROB - [U, CC] - SC$ are given in Table 3.

Remarks concerning Table 3. Eppen and Martin [15] provided a first shortest path formulation for $LS - U - SC$ with $O(n^3)$ variables.

Again for $LS - U - SC$, Rardin and Wolsey [41] showed that the separation problem for $(l, R, S)$ inequalities can be solved by a single max flow calculation in a graph with
Table 3: Model PROB – CAP – SC with Start-Ups

<table>
<thead>
<tr>
<th>FORMULATION</th>
<th>LS</th>
<th>WW</th>
<th>DLSI</th>
<th>DLS</th>
</tr>
</thead>
<tbody>
<tr>
<td>U</td>
<td>$SP(SC) \times O(n^2)$</td>
<td>$O(n^2) \times O(n) \times O(n^2)$</td>
<td>$O(n^2) \times O(n)$</td>
<td>$–$</td>
</tr>
<tr>
<td></td>
<td>$FL(SC) \times O(n^3)$</td>
<td>$–$</td>
<td>$–$</td>
<td>$–$</td>
</tr>
<tr>
<td>CC</td>
<td>$O(n^2) \times O(n^3)$</td>
<td>$O(n^2) \times O(n^2)$</td>
<td>$O(n^2) \times O(n^2)$</td>
<td>$[49]$</td>
</tr>
<tr>
<td>SEPARATION</td>
<td>$U$</td>
<td>$O(n)$</td>
<td>$–$</td>
<td>$–$</td>
</tr>
<tr>
<td></td>
<td>$(l, R, S)$</td>
<td>$O(n)$</td>
<td>$–$</td>
<td>$–$</td>
</tr>
<tr>
<td></td>
<td>$[52, 57]$</td>
<td>$[1, 16, 55]$</td>
<td>$[1, 16, 55]$</td>
<td>$[17]$</td>
</tr>
<tr>
<td></td>
<td>$CC$</td>
<td>$O(n^3)$</td>
<td>$O(n^2)$</td>
<td>$O(n^2)$</td>
</tr>
<tr>
<td></td>
<td>$[11]$</td>
<td>$[19]$</td>
<td>$[17]$</td>
<td>$[46]$</td>
</tr>
</tbody>
</table>

O($n^3$) nodes.

For WW – U – SC the $(l, S)(SC)$ inequalities are a simple modification of the $(l, S)(WW)$ inequalities to include start-up variables.

In [11], $O(n^2)$ separation algorithms are given for the classes of left and right submodular inequalities that are valid for LS – C – SC with varying capacities. Also an $O(n^3)$ separation algorithm is given for the family of left $klsi$ inequalities valid for LS – CC – SC.

In [44], polynomial separation algorithms are given for several classes of hole/bucket inequalities for DLS – CC – SC.

Formulations for DLSI – CC – ST can be obtained by viewing the set $X_{DLSI-CC-SC}$.
as the union of \( n + 1 \) sets of the form \( X^{DLS-CC-SC} \) depending on the possible values taken by the initial stock variable \( s_0 \).

4.4 Other Variants

We indicate a series of results concerning either formulations or families of valid inequalities that can be useful.

- \( WW-U-\{B, SC\} \). In [2], an \( O(n^2) \times O(n) \) reformulation is presented generalizing those for \( WW-U-B \) and \( WW-U-SC \).
- \( LS-U-\{SS, SL\} \). In [27], a family of valid inequalities describing the convex hull are presented, as well as tight extended formulations in certain special cases.
- \( LS-CC-SC \). In [11], several families of valid inequalities are presented as well as efficient separation algorithms.
- \( LS-U-LB \) In [12], models are studied that provide relaxations of both \( LS-U-LB \), and also of single period relaxations of multi-item models.
- \( LS-CC-ST(C) \) For the optimization problem a dynamic programming algorithm is presented in [43].

5 Classification of Multi-Item/Machine/Level Problems

Here we present a minimal extension of the classification scheme to deal with a limited class of multi-item and/or multi-machine problems. We assume that there are several items and one or more machines.

Machines \( \{ NK = \#, [IM, VM], [LT]^*, [SB1, SB2, BB], [SET, ST, SQT, SQC]^* \} \)

The first subfields are simple.
\( NK \) is the number of machines.
\( LT \) indicates that there are lead times.

The next subfield gives information about the time periods.
If a machine produces more than one item, there are typically joint capacity constraints across items. When periods are short so that only one or two items are produced by the machine in a period, one talks of small time buckets. When more than two set-ups are permitted per period, there are big time buckets.

The following subfield gives information about the time buckets. 
$SB_1, SB_2$ indicate a small bucket model in which either at most one or at most two set-ups are permitted per period respectively. $SB_1$ is often referred to as a model with mode constraints.
$BB$ denotes a big bucket model with at least one joint capacity constraint imposing a limit $L^k_t$ on the amount of capacity available in each period. $a_{ik}$ denotes the capacity consumption rate per unit of item $i$.

The last subfield gives information about the capacity utilization.
$SET$ indicates that there are also set-up times $t_{ijk}$ that reduce the capacity available.
$ST$ indicates that there are start-up times $e_{ijk}$.
$SQT$ indicates that there are sequence dependent changeover times $q_{tijk}$.
$SQC$ indicates that there are sequence dependent changeover costs $q_{tijk}$ whether it is a big or small bucket model.

**Multi-Level Production** \{\text{NL} = \#, [G, A, S]\}.

The production structure classification is simple
\text{NL} denotes the number of levels, with $\rho_{tijk}$ the number of units of item $i$ needed to produce one item of $j$ on machine $k$ in period $t$ for each item $j \in S(i)$, the set of successors of $i$.
$G$ denotes a general product structure
$A$ denotes assembly structure
$S$ denotes in series assembly structure, i.e. linear.

Finally to complete this very partial classification, we may wish to add $NT = n$ the number of time periods, and $NI$ the number of items.
5.1 MIP formulation

Introducing additional suffices \( i \) or \( j \) for items, and \( k \) for machines, we also require new variables \( u^{ijk}_{it} \) to model sequence dependent changeovers. Most of the problems covered by the above classification can now be represented by the MIP:

\[
\begin{align*}
\min & \sum_{i,k,t} \text{Cost}(x^{ik}_{it}, y^{ik}_{it}, s^{i}_{it}, r^{i}_{it}, z^{ik}_{it}) + \sum_{i,j,k,t} q^{ijk}_{it} u^{ijk}_{it} \\
& s^{i}_{t-1} - r^{i}_{t-1} + \sum_{k} x^{ik}_{it} = d^{i}_{t} + \sum_{j \in S(i)} \rho^{jik} x^{jk}_{it} + s^{i}_{t} - r^{i}_{t} \\
& \sum_{i}(a^{ik} x^{ik}_{it} + b^{ik} y^{ik}_{it} + c^{ik} z^{ik}_{it} + \sum_{j \neq i} q^{tijk} u^{ijk}_{it}) \leq L_{t} \tag{25}
\end{align*}
\]

Constraints modelling start – ups \( \tag{27} \)

Constraints modelling sequence – dependence, etc \( \tag{28} \)

\[
\ldots
\]

We note that in \( SB1 \) models, \( a^{ik} \) and \( c^{ik} \) and \( q^{tijk} \) are zero, and the inequality (26) reduces to

\[
\sum_{i} y^{ik}_{it} \leq 1 \text{ for all } k, t. \tag{29}
\]

One possible model for \( SB2 \) has the constraints

\[
\begin{align*}
\sum_{i} y^{ik}_{it} & \leq 2 \\
\sum_{i}(y^{ik}_{it} - z^{ik}_{it}) & \leq 1.
\end{align*}
\]

The latter constraint says that there is only one set-up per period that is not a start-up.

5.2 Known Results for Multi-Item Problems

We present a few basic results on polynomial solvability, reformulation, and valid inequalities. In all the special cases below, there is a single machine (\( NK=1 \)).

- Multi-Level Uncapacitated Lot-Sizing in Series. \{\( NL > 1, S \}\{LS – U \} is polynomially solvable by dynamic programming. \[60\]}
• Multilevel-Level Lot-Sizing. \{NL > 1, G\}\{LS – CC – \{VAR\}\}. Using an echelon stock reformulation [9] leads to a formulation with a single-item lot-sizing problem for each item.

• Multi-Item Single Mode Constant Capacity Discrete Lot-Sizing. \{SB\}\{DLS – CC\} reduces to a network flow problem. This is part of the folklore, see for example [32].

• Multi-Item Single Mode Constant Capacity Discrete Lot-Sizing with Backlogging. \{SB\}\{DLS – CC – B\}. The convex hull of solutions is obtained using the convex hull formulation for NI = 1 plus the mode constraints (29), see [32].

• Big Bucket Problems with Set-Up Times. \{BB, SET\}\{LS – C\}. Valid inequalities have been proposed by Miller et al. [30, 31].

• \{[BB, SB\{1, SB\{2\}, [SQT, SQC]\}*\} Formulations for sequence-dependent changeovers for small buckets and big buckets can be found in [7, 10, 21, 57].

6 Three Problems: Reformulation by Classification

Here we show how to profit from the classification of Sections 3 and 4 to obtain a good formulation. We then demonstrate the approach on three problem instances. In each case we first classify the instance. Then we use the Tables to derive a strong reformulation of the instance that is then fed into a standard MIP solver. Results obtained are compared either with those provided by alternative formulations, or with those obtained earlier using one or more special purpose systems.

6.1 Use of the Classification

As an illustration of how to use the classification, we consider a multi-item single level single machine problem. Suppose that the problem is single mode with backlogging and constant capacities, namely \{NK = 1, SB\}\{LS – CC – B\}.

**Step 1.** Check to see if the costs are Wagner-Whitin, as this property is unaffected by mode constraints. We assume that the answer is positive.
Step 2. Check $WW - CC - B$ in Table 2. An approximate reformulation is proposed, but $O(n^3) \times O(n^3)$ appears too large.

Step 3. We can move upwards or towards the right in the Table 2 to find a relaxation. Moving upwards from $CC$ to $U$, the relaxation $WW - U - B$ is obtained for which a tight $O(n^2) \times O(n)$ reformulation is indicated in Table 2.

Step 4. Moving right from $WW$ to $DLSI$, we obtain the relaxations $DLSI_k - CC - B$ for which a good $O(n^2) \times O(n^2)$ reformulation is again known for each $k$. However this leads to an $O(n^3) \times O(n^3)$ formulation, which is again rejected as being too big.

Step 5. Decide to use the reformulation of Step 3 which has $NI \times O(n^2)$ constraints and $NI \times O(n)$ variables, and is of reasonable size.

A similar approach has been taken in tackling the three instances treated below, starting from the verbal classification dervied in Section 2.

6.2 Problem 1: Bottling

i) Multi-item constraints and costs. At most one item can be produced per period.

ii) Individual item constraints and costs. When produced, each item is produced for between 7 and 16 hours, so both the upper bound and the lower bounds on production per period are time invariant. Also the unit production and storage costs are time invariant, and there are start-up costs.

From this, the problem can be classified as $\{NK = 1, SB\} \{WW - CC - \{SC, LB\}\}$ with formulation

22
\[ \min \sum_{i,t}(p_i^tx_i^t + h_i^ty_i^t + f_i^ty_i^t + g_i^tz_i^t) \]

\[ s_{i-1}^i + x_i^i = d_i^i + s_i^i \quad \forall i, t \]  

\[ x_i^t \leq C_i^ty_i^t \quad \forall i, t \]

\[ x_i^t \geq L_i^ty_i^t \quad \forall i, t \]

\[ \sum_t y_i^t \leq 1 \quad \forall t \]

\[ z_i^t \geq y_i^t - y_{i-1}^t \quad \forall i, t \]

\[ z_i^t \leq y_i^t \quad \forall i, t \]

\[ x, s \geq 0, y, z \in \{0, 1\}. \]

In Table 3 we see that the reformulation of \( WW - CC - \{SC, LB\} \) is blank. However there is an \( O(n^2) \times O(n) \) reformulation of \( WW - U - SC \). Also in Table 1 we see that there is an \( O(n^2) \times O(n^2) \) reformulation of \( WW - CC \).

The reformulation for \( WW - U - SC \) is obtained by just adding the \( O(n^2) \) inequalities

\[ s_{t-1}^t \geq \sum_{j=t}^l d_j(1 - y_t - z_{t+1} - \ldots - z_l) \quad \forall t, l \text{ with } t \leq l. \]

The reformulation for \( WW - CC \) for each item is

\[ s_{k-1}^t \geq C \sum_{i=k}^n f_i^k \delta_i^k + C \mu_k \quad \forall k \]

\[ \sum_{u=k}^t y_u \geq \sum_{\tau \in \{0\} \cup \{k, n\}} \left[ \frac{d_k^\tau}{C} - f_k^\tau \right] \delta_k^\tau - \mu_k \quad \forall k, t, k \leq t \]

\[ \sum_{\tau \in \{0\} \cup \{k, n\}} \delta_k^\tau = 1 \quad \forall k \]

\[ \mu_k \geq 0, \delta_k^t \geq 0, \text{ for } t \in \{0\} \cup \{k, n\} \forall k \]

\[ 0 \leq y_t \leq 1 \quad \text{for } t = 1, \ldots, n \]

where \( f_0^k = 0, f_k^\tau = \frac{d_k^\tau}{C} - \left[ \frac{d_k^\tau}{C} \right] \) and \( [k, t] \) denotes the interval \( \{k, k+1, \ldots, t\} \). The additional variables \( \delta_k^t \) indicate that \( s_{k-1}^t = C f_k^t \text{ (modulo } C) \).

In Table 4 we present computational results showing the effects of the reformulations. Instance cl-1a is the original formulation (30)-(37). Instance cl-1b is with the addition of the inequalities (38) for \( WW - U - SC \). Instance cl-1c has in addition the reformulation

23
(39)-(43) of WW – CC for each item. The nine columns represent the instance, the number of rows, columns and 0-1 variables, followed by the initial LP value, the value XLP after the system has automatically added cuts, IP the optimal value, the total number of seconds required to prove optimality, and finally the number of nodes in the branch-and-cut tree. All runs were carried out with the default version of the XPRESS MIP optimizer [59] version 12.50 running on a 500Mhz Pentium III under Windows NT.

<table>
<thead>
<tr>
<th>instance</th>
<th>m</th>
<th>n</th>
<th>int</th>
<th>LP</th>
<th>XLP</th>
<th>IP</th>
<th>secs</th>
<th>nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>cl-1a</td>
<td>511</td>
<td>720</td>
<td>120</td>
<td>1509.1</td>
<td>3549.6</td>
<td>4414.2</td>
<td>5000</td>
<td>3.8 ×10^5</td>
</tr>
<tr>
<td>cl-1b</td>
<td>2354</td>
<td>720</td>
<td>120</td>
<td>3800.6</td>
<td>4305.1</td>
<td>4404.5</td>
<td>82</td>
<td>3826</td>
</tr>
<tr>
<td>cl-1c</td>
<td>4454</td>
<td>2824</td>
<td>120</td>
<td>4309.9</td>
<td>4310.5</td>
<td>4404.5</td>
<td>82</td>
<td>175</td>
</tr>
</tbody>
</table>

Table 4: Results for Problem 1

An asterisk * indicates that the run was terminated before optimality was proved. For formulation cl1a the best lower bound on termination was 4251.2 leaving a gap of 3.7%.

### 6.3 Problem Instance 2: Discrete Lot-Sizing and Sequence Dependent Changover Costs

i) Multi-item constraints and costs. At most one item can be produced per period, and there are sequence dependent set-up costs.

ii) Individual item constraints and costs. Production is all or nothing with constant capacities. There are no unit production costs, and storage costs are nonnegative and constant over time.

The problem can be classified as \{NK = 1, SB1, SQC\}{DLS – CC}.

As observed in [18], there is no backlogging, so demands can be normalized with \(d_t \in \{0,1\}\). A basic formulation is then
\[ \begin{align*}
\min & \sum_{i,t} h_i s_i^t + \sum_{i,j,t} q_{ij} u_{ij}^t \\
& s_{i-1}^t + x_i^t = d_i^t + s_i^t \quad \forall \ i, t \\
& x_i^t \leq y_i^t \quad \forall \ i, t \\
& \sum_i y_i^t = 1 \quad \forall \ t \\
& u_{ij}^t \geq y_{i-1}^t + y_j^t - 1 \quad \forall \ i, j, t \\
x, y \in \{0, 1\}, s, u \geq 0.
\end{align*} \]

**Observation 1** The reformulation of changeover variables [21, 57] indicated in Section 5.2 leads to the constraints

\[ \begin{align*}
\sum_i u_{ij}^t &= y_j^t \quad \forall j, t \\
\sum_j u_{ij}^t &= y_i^t - 1 \quad \forall i, t \\
\sum_i y_i^0 &= 1 \\
u_{ij}^t &\geq 0 \quad \forall i, j, t
\end{align*} \]

representing the flow of a single unit passing from item set-up to item set-up over time. Here the set-up variable \( y_i^t \) is the flow through node \((i, t)\) and \( u_{ij}^t \) is the flow from node \((i, t-1)\) to node \((j, t)\) indicating a switch from a set-up of item \(i\) in period \(t-1\) to a set-up of item \(j\) in \(t\).

**Observation 2:** Inclusion of start-up variables. When there are changeover variables, there are implicitly start-up variables for which we know tighter formulations. Thus we introduce the equations

\[ z_i^t = \sum_{i \neq j} u_{ij}^t \]

to define the start-up variables. Switch-off variables \( w_i^t \) can be defined similarly. This means that it is possible to use results for the single item model \( DLS - CC - SC \).

**Observation 3:** Reformulation of \( DLS - CC - SC \)
From Table 3, we see that there is a tight $O(n^2) \times O(n)$ reformulation under the assumption of Wagner-Whitin costs. This consists of the inequalities

$$s^i_{t-1} + \sum_{u=t}^{t+p-1} y^i_u + \sum_{u=t+1}^{t+p-1} (d_{ul} - (t + p - u))z_u + \sum_{u=t+p}^l d_{ul}z_u \geq p$$

for all $t, l$ such that $d_l = 1, l \geq t$, where we suppose that $d_{t_1} = \ldots = d_{t_p} = 1$ with $t < t_1 < \ldots < t_p = l$ and $d_r = 0$ in intervening periods in $\{t, \ldots, l\}$.

In Table 5 we present computational results showing the effects of the reformulations. Instance cl2-NTa is the initial formulation, instance cl2-NTb is the formulation with reformulation from Observations 1, and instance cl2-NTc also includes the reformulation of $DLS - CC - SC(WW)$ from Observations 2 and 3. Instances with $NT = 35$ and $NT = 60$ periods were solved. Table 5 has the same structure as Table 4.

<table>
<thead>
<tr>
<th>instance</th>
<th>m</th>
<th>n</th>
<th>int</th>
<th>LP</th>
<th>XLP</th>
<th>IP</th>
<th>secs</th>
<th>nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>cl2-35a</td>
<td>3797</td>
<td>4110</td>
<td>350</td>
<td>27.2</td>
<td>34.7</td>
<td>2056</td>
<td>1800*</td>
<td>51500*</td>
</tr>
<tr>
<td>cl2-35b</td>
<td>2062</td>
<td>5130</td>
<td>690</td>
<td>180.9</td>
<td>531.6</td>
<td>1599</td>
<td>1800*</td>
<td>8000*</td>
</tr>
<tr>
<td>cl2-35c</td>
<td>2599</td>
<td>5130</td>
<td>690</td>
<td>1361.5</td>
<td>1361.5</td>
<td>1387</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cl2-60c</td>
<td>4817</td>
<td>8880</td>
<td>1190</td>
<td>1453.6</td>
<td>1454.0</td>
<td>1560</td>
<td>17579</td>
<td>8117</td>
</tr>
</tbody>
</table>

Table 5: Results for Problem 2

Note that cl2-35a and cl2-35b are unsolved after 1800 seconds. The best lower bounds obtained are 240.9 and 804.3 respectively.

6.4 Problem 3: Multi-Level Assembly

i) This is a multilevel problem with assembly type product structure.

ii) Multi-item constraints and costs. Many items can be produced in each period, and the capacity constraints limiting production in each period involve both production levels and set-up times for families.

iii) Individual item constraints and costs. There are no individual capacity constraints, but there are storage costs and implicit fixed costs through the families.

This gives the classification $\{NL > 1, A\} \{NK > 1, BB, ST(Family)\} \{LS - U\}$. 26
We now present the initial formulation from [42], except for the replacement of the stock variables $s_i^t$ by echelon stock variables $e_i^t$, where $s_i^t = e_i^t - e_i^{\sigma(i)}$ and $\sigma(i)$ is the unique successor if any of item $i$. This gives

\[
\begin{align*}
\min & \sum_{i,t} \bar{h}_i^t e_i^t + \sum_{f,t} c^f \eta^f_t \\
 e_{i-1}^t + x_i^t &= d_i^{\sigma(i)} + e_i^t \quad \text{for all } i, t \\
 e_i^t &\geq e_i^{\sigma(i)} \quad \text{for all } i, t \\
x_i^t &\leq M y_i^t \quad \text{for all } i, t \\
y_i^t &\leq \eta^f_t \quad \text{for all } i, f, t \text{ with } i \in F(f) \\
\sum_{i \in F(f)} a^f_i x_i^t + \sum_{g \in V(f)} b_{gf} \eta^g_t &\leq C_{f}^t \eta^f_t \quad \text{for all } f, t \\
y_i^t, \eta^f_t &\in \{0, 1\}, x_i^t, s_i^t \geq 0 \quad \text{for all } i, f, t
\end{align*}
\]

where $q(i)$ is the final product containing item $i$, $h_i^t = h_i^t - \sum j \in P(i) h_j^t$ where $P(i)$ is the set of immediate predecessors of item $i$, $\eta^f_t$ is the set-up variable for family $f$ in period $t$, $F(f)$ is the set of items in family $f$ and $V(f)$ is a set of families appearing in the budget constraint of family $f$.

This model can also be reformulated by eliminating the $y_i^t$ variables giving

\[
x_i^t \leq M \eta^f_t \quad \text{for all } i, f, t \text{ with } i \in F(f),
\]

in place of the constraints (47)-(48).

As observed in Section 4.2, the echelon stock formulation is such that the constraints (45)-(47) give a model of the form $LS - U$. Rather than use an $O(n) \times O(n^2)$ reformulation of $LS - U$ involving many new variables, we have used the reformulation $WW - U$, see Table 1. In addition to avoid adding too many constraints, we have added only a subset of the $(l, S)(WW)$ inequalities

\[
e_{i-1}^t + \sum_{u=t}^l d_{u}^{\sigma(i)} y_u \geq d_{lt}^{\sigma(i)} \quad \text{for all } t, l, l - t \leq PAR
\]

where $PAR$ is an integer. We denote the resulting formulation by cl3-NT-#c, where $\# \in \{1, 2\}$ is the number of the instance.

27
In the model with the $y^i_t$ variables eliminated, we can do something similar, adding the constraints

$$e^i_{t-1} + \sum_{u=t}^{l} d^{o(i)}_{ul} \eta^{f(i)}_u \geq d^{o(i)}_{tl}$$

for all $t, l, l-t \leq PAR$, where $f(i)$ is any family containing item $i$. Clearly these inequalities are only unique when each item belongs to just one family. We denote the resulting formulations by cl3-NT-#b.

In Table 6 we present results for the four instances tackled in [7]. In all cases NT=16. The two 78 item instances have each item belonging to a single family, so for these we have used the more compact formulation cl3-78-#b. These two instances were run with PAR=4.

The 80 item instances were run with the larger formulation cl3-80-#c, and with PAR=8.

The columns of Table 6 contain the same information as in Tables 4 and 5, except that the last column has been replaced by the % Gap on termination, where $GAP = \frac{BIP - BLB}{BIP} \times 100$ with BLB the value of the best lower bound.

<table>
<thead>
<tr>
<th>Instance</th>
<th>r</th>
<th>c</th>
<th>int</th>
<th>LP</th>
<th>XLP</th>
<th>BIP</th>
<th>Secs</th>
<th>BLB</th>
<th>Gap %</th>
</tr>
</thead>
<tbody>
<tr>
<td>cl3-78-1b</td>
<td>7607</td>
<td>2688</td>
<td>192</td>
<td>10777.0</td>
<td>10839.9</td>
<td>11592.0</td>
<td>450</td>
<td>10934.0</td>
<td>5.7</td>
</tr>
<tr>
<td>cl3-78-2b</td>
<td>7618</td>
<td>2688</td>
<td>192</td>
<td>10464.8</td>
<td>10511.1</td>
<td>10926.0</td>
<td>450</td>
<td>10550.9</td>
<td>3.4</td>
</tr>
<tr>
<td>cl3-80-1c</td>
<td>13725</td>
<td>4128</td>
<td>288</td>
<td>21376.9</td>
<td>21551.7</td>
<td>25160.3</td>
<td>900</td>
<td>21869.3</td>
<td>13.1</td>
</tr>
<tr>
<td>cl3-80-2c</td>
<td>13700</td>
<td>4128</td>
<td>288</td>
<td>21951.6</td>
<td>22152.5</td>
<td>26377.4</td>
<td>900</td>
<td>22417.3</td>
<td>15.0</td>
</tr>
</tbody>
</table>

Table 6: Results for Problem 3

The best results obtained in [7] were gaps of 8.1,4.9,% running bc-opt on the two 78 item instances with the echelon stock formulation (44)-(50), but with (47) replaced by (51), and gaps of 13.5,13.8 % running bc-prod on the two 80 item instances using the original formulation without echelon stock variables. There all four instances were run for 1800 secs on a 350 Mhz Pentium running under Windows NT.
7 Conclusions

The three examples treated in the last section suggest that certain practical lot-sizing problems can now be effectively tackled with nothing but appropriate tight a priori reformulations and a commercial mixed integer programming system. Another such example can be found in [32].

The classification scheme for single item problems introduced and detailed in Sections 2 and 3 show that there are still a number of open questions whose solution would allow us to tackle an even larger range of lot-sizing problems. Here we list a few that we believe are the most important or challenging.

i) $DLSI - CC - B$. Find a compact tight reformulation, and establish whether the $O(n^2) \times O(n^2)$ formulation from [32] is tight. This question is also of importance for $WW - CC - B$.

ii) $DLSI - CC - SC$ and $DLS - CC - \{B, SC\}$. Find compact formulations and/or strong valid inequalities.

iii) $LS - CC - SS$. Find formulations and valid inequalities.

iv) $PROB - C$. Find fast and effective separation heuristics for the dynamic knapsack inequalities proposed in [26].

v) $NK > 1, NI = 1$. Study the multi-machine single-item problem. Do the dynamic knapsack inequalities suffice computationally? For problems with two machines, do the recent two variable knapsack results of Agra and Constantino [3] provide useful inequalities?

There are also obviously a wealth of questions when one turns to multi-item problems. Some important ones are:
vi) \(\{SB1\} - \{WW - U\}\). For the simplest possible single mode problem, find valid inequalities involving multiple items.

vii) \(\{BB - ST\} - \{LS - CAP\}\). Find valid inequalities to deal with start-up times in big bucket models, extending the results of [30, 31].

viii) \(\{BB - [SQC, SQT]^*\} - \{PROB - CC\}\). Find valid inequalities for big bucket models with sequence dependent costs and/or times.

It is also perhaps worth pointing out that there is to our knowledge still no complete convex hull description, or compact convex hull reformulation for the basic uncapacitated lot-sizing in series problem \(\{NL > 1, S\} - \{LS - U\}\).

The approach advocated here also raises algorithmic questions, such as finding ways to combine valid inequalities and tight reformulations, finding approximate, but more compact, reformulations that are tight for many instances, or using the reformulations with LP to solve the separation problems. Given that some reformulations provide very good bounds, but are too large to be effective during enumeration, one could also perhaps imagine working simultaneously with more than one formulation. Finally there is the largely untouched question of whether the classification and reformulations can be used to develop effective primal heuristics.

References


[16] A. Federgrun and M. Tsur, "A simple forward algorithm to solve general dynamic lot-size models with \( n \) periods in \( O(n \log n) \) or \( O(n) \) time, Management Science 37, 909-925, 1991.


[33] A. Miller and L.A. Wolsey, Tight formulations for some simple MIPs and Convex Objective IPs, CORE, Université Catholique de Louvain, Belgium September 2001


[58] L.A. Wolsey, Integer Programming for Production Planning and Scheduling, Chorin Winter School, November 1999
