UNIVERSITE PARIS DAUPHINE

Laboratoire d'Analyse et Modélisation de Systèmes pour l'Aide à la Décision
CNRS UMR 7024

CAHIER DU LAMSADE

202

Janvier 2003

An alternative to safety stock policies for multi-level rolling schedule MRP problems

N. P. Dellaert, J. Jeunet

CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE

An alternative to safety stock policies for multi-level rolling schedule MRP problems

N. Dellaert*and J. Jeunet†

November 16, 2002

Abstract

We study the impact of positive lead times on the multi-level lot-sizing problem in a rolling schedule environment. We show how stockout situations may arise even in a context of deterministic demand. We therefore develop a procedure to avoid such stockouts and we compare its performance through a simulation study to a safety stock strategy. Simulation results show the superiority of the proposed procedure.

1 Introduction

Research on lot-sizing decisions in MRP systems has essentially focused on the development of single-level and multi-level heuristics for solving problems with deterministic demands over finite horizons. However, this static framework ignores the common practice of using a rolling schedule. This approach consists in applying a lot-sizing rule over a limited number of future periods, the forecast window, for which demand is known either deterministically or probabilistically. Usually, only the first lot size is implemented and the horizon rolls forward the next decision period. New demands are then revealed, the lot-sizing technique is re-applied and the decision of the first period is again enacted. This process is repeated until the final period of the problem horizon is reached.

Optimal methods for single and multi-level problems with a fixed horizon do not necessarily provide an optimal solution in a rolling-schedule environment. On the contrary, several single-level studies have shown that it might be worth using computationally simple heuristics especially when the forecast window is short. Blackburn and Millen (1980) show that the Wagner-Whitin algorithm (1958) can be outperformed by the Silver-Meal heuristic (1972), notably when the number of known future demand is limited. De Bodt, Van Wassenhove and Gelders (1982), while analyzing the effect of forecast errors within the forecast window on cost performance of several single-level models, show that it might be worth using the 'simplistic' Economic Order Quantity. Aucamp (1985) provides a comparative study of the performance of several lot-sizing rules, also used in combination with a Look ahead/Look back strategy which consists in either increasing or decreasing the lot sizes generated by any

*Department of Operations Management, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands. E-mail: n.p.dellaert@tm.tue.nl

†CNRS, LAMSADE, Université Paris IX Dauphine, Place du Maréchal de Lattre de Tassigny, 75775 Paris Cedex. E-mail: jeunet@lamsade.dauphine.fr

rule until no cost improvement can be found. The author experimentally observes the poor performance of the Wagner-Whitin algorithm and the consistently high performance of Least Total Cost and Silver-Meal. Bookbinder and Hn'g (1986) show that a modified version of the Silver-Meal heuristic (to deal with sharply decreasing demand patterns) and a heuristic by Bookbinder and Tan (which combines the Silver-Meal and the Least Unit Cost criteria) yield the best results in most situations. However, the Wagner-Whitin algorithm is the best method for large forecast windows and any demand pattern other than constant demand.

Similar conclusions have been drawn based upon multi-level studies under rolling schedule conditions. Blackburn and Millen (1982) evaluate the cost performance of the Silver-Meal heuristic and the Wagner Whitin algorithm, also used in combination with several cost modifications designed to account for interdependencies among stages in assembly structures. Simulation results indicate that in many situations, the Silver-Meal heuristic outperforms the Wagner-Whitin algorithm. Gupta, Keung and Gupta (1992) show that the Silver-Meal heuristic provides lower costs than the Wagner-Whitin algorithm in most cases. In a recent study Simpson (1999) finds that the Silver-Meal heuristic when combined to one of the cost modifications of Blackburn and Millen (1982) provides the lowest cost schedules under short forecast windows. For larger windows however, the Wagner-Whitin algorithm yields better results. This conclusion was already drawn in earlier single-level studies (Blackburn and Millen, 1980, Bookbinder and Hn'g, 1986).

Past results make clear that forecast window length impacts cost performance, mostly because length dramatically affects the first optimal lot size, a phenomenon called 'the horizon effect'. An obvious strategy to overcome this difficulty is to lengthen the window W so as to stabilize the first lot size. In practice, future demand data or accurate forecast are available only for a limited number of future periods. The idea is therefore to use available demand data through W and forecast demand beyond W. This strategy is only useful when combined to the Wagner-Whitin algorithm as myopic methods ignore additional information and provide identical schedules, be the horizon extended or not. Implementing such a strategy raises the question of when to stop extending the horizon.

Horizon extension approaches have been widely implemented in single-level problems. Carlson, Beckman and Kropp (1982) investigate the impact of extending the horizon on the cost performance of the Wagner-Whitin algorithm and conclude that in some situations the more information the better. Kropp, Carlson and Beckman (1983) propose and test 4 stopping conditions. They conclude that simpler stopping rules perform better. More recently, Russel and Urban (1993) show that, with horizon extension, the Wagner-Whitin algorithm beats the Silver-Meal heuristic for moderate to large window values. For small W-values horizon extension does not help the Wagner-Whitin algorithm to seek improvements over the Silver-Meal heuristic. Refinement of the horizon extension principle has been brought by Stadtler (2000) who proposes a cost modification to be used with the Wagner-Whitin algorithm so as to account for the fictitious aspect of demand data beyond the forecast window. The idea is to assign to a lot size a cost which is proportional to the periods the order covers falling within the forecast window. Stadtler expects this modification to favour more orders in later periods. The resultant look-beyond model yields the best overall results when compared to the Silver-Meal technique and the heuristic of Groff (1979).

Horizon extension is naturally discontinued when a planning horizon is found since the first optimal lot size stabilizes in that case. But the common situation may rarely offers this opportunity. Chand (1982) has therefore designed a simple decision rule to select the first lot size whenever a planning horizon is not found. The rule consists in choosing the first lot size with a minimum cost per period. The set of first lot sizes is provided by solving optimally all t-period problems, with $t = 1, \ldots, W$. The computational study indicates that the modified algorithm exhibits a better behaviour than the Wagner-Whitin algorithm and the Silver-Meal heuristic. Later on, Chand (1983) has provided an adaptation of his algorithm to serial systems. Chand is therefore the first author to design a rolling procedure for multi-stage problems even if it is restricted to the very specific serial case.

Despite this abundant relevant literature on the rolling-schedule problem, there is a paucity of research examining multi-level instances with general product structures and positive lead times. However, common industrial settings often involve product structures with numerous common parts. Furthermore, the zero lead time assumption is hardly realistic and more restrictive than it seems at first sight. Indeed, implementing lot sizing decisions on a rolling basis becomes much more complex in this context (i.e. positive lead time and general structures) as lot-sizing methods may provide infeasible schedules due to a lack of component availability.

In this paper we develop a procedure to cope with stockout situations that arise when positive lead times are introduced. This procedure is compared to a strategy that consists in introducing safety stocks at all levels in the product structure. We show through simulation experiments the superiority of our procedure over the safety stock strategy.

The next section is dedicated to a detailed description of the multi-level lot-sizing problem under rolling schedule conditions. We provide an example of a stockout resulting from the introduction of positive lead times. Section 3 briefly presents the safety stock strategy that may be implemented to avoid such stockout situations. The alternative solution method, namely our repair procedure, is then described. Section 4 is dedicated to the experimental framework designed to compare the cost effectiveness of several lot-sizing techniques when combined to either our repair procedure or the safety stock strategy. Section 5 analyses the simulation results. We conclude in Section 6 with the main findings of our study and limitations.

2 The multi-level lot-sizing problem under rolling schedule conditions

2.1 Statement of the problem

It is common to represent product structures as directed acyclic graphs (see Bookbinder and Koch, 1990, for example). In such a graph each node corresponds to an item and each edge (i, j) between node i and node j indicates that item i is directly required to assemble item j. Node i (equivalently item i) is fully defined by $\Gamma^{-1}(i)$ and $\Gamma(i)$, its sets of immediate predecessors and successors. The set of ancestors –immediate or non-immediate predecessors– of item i is denoted by $\hat{\Gamma}^{-1}(i)$. Product structures may be categorized in terms of their complexity index, $\mathcal{C}$, as defined by Kimms (1997). Recall products are numbered in topological order by the integers $1, \ldots, P$ and let $P(k)$ be the number of products at level k, with $k = 0, \ldots, K$ ($K+1$

is the depth of the structure). The total number of items obviously equals $\Sigma_{k=0}^{K} P(k)$, which by definition is also P. The most complex—in the sense of having the largest number of product interdependencies—structure is obtained when each item enters the composition of all the items located at higher levels in the product structure. By contrast, the simplest structure obtains when each item enters the composition of exactly one item belonging to a higher level. Kimms (1997) defines the complexity of a product structure as

$$\mathcal{C} = \frac{A - A_{\min}}{A_{\max} - A_{\min}} \tag{1}$$

where $A = \Sigma_{i=1}^{P} |\Gamma(i)|$ is the actual number of arcs in the structure. There is of course a minimal number of arcs such as the product structure is connected, which we denote $A_{\min}$ and is equal to $P - P(0)$. Conversely there is a maximum number of arcs denoted $A_{\max}$ that the graph can contain, and which is written as $A_{\max} = \Sigma_{k=0}^{K-1}\{P(k) \cdot \Sigma_{j=k+1}^{K} P(j)\}$. Structures for which the number of arcs equals the minimum number of arcs ($A = A_{\min}$) are necessarily assembly structures with a zero $\mathcal{C}$-value, whereas structures such as $A = A_{\max}$ satisfy $\mathcal{C} = 1$. The $\mathcal{C}$-index is therefore bounded from below and above, whereas the traditional index of Collier (1981) is not.

Let L_i be the cumulative lead time of item i and l_i its own lead time (time required to produce or assemble item i). We have

$$L_i = \max_{j \in \Gamma^{-1}(i)} (L_j + l_j), \tag{2}$$

with $L_i = 0$ for all i with no predecessor. As decision periods are the periods in which we order and not the delivery periods, individual lead times l_i are not included in the definition of L_i.

Let W be the forecast window length and T, the final period of the planning horizon, with $W \leq T$. Period T is usually infinite, as a rolling problem has no true end. But in the simulation experiments, period T must be set to a finite value. For each decision period t, lot-sizing rules are applied on time interval $R = \{t, \ldots, \min(t + W - 1, T)\}$, with $t = 1, \ldots, T$ and only the first lot size (that of period t) is executed. Within interval R, demand for end items is either known with certainty or probabilistically. In the latter case, demand forecast is subject to error (note the word 'forecast' is not suitable when there is no error). The interval on which lot-sizing rules are applied may be extended to E periods, to increase their efficiency. Interval R becomes $R = \{t, \ldots, t+W-1, \ldots, \min(t+E-1, T)\}$. Demand in periods $t+W, \ldots, t+E$ may be generated with the same pattern as the one utilized in periods $t+1, \ldots, t+W-1$. This demand is always fictitious.

Gross requirements $d_{i,s}$ for item i in period $s \in R$ correspond to the demand that must be satisfied in period $s + l_i$. Gross requirements are either forecasted or known with certainty for end items within the forecast window whereas they result from planned and firm orders at deeper levels for components.

Let $x_{i,t}$ be a firm order for item i in period t. Quantity $x_{i,t}$ is the number of units of item i which is ordered in period t to fill the net requirements for item i in period $t + l_i$.

The pipeline inventory $Z_{i,s}$ is defined as

$$Z_{i,s} = \max(0, Z_{i,s-1} + x_{i,s} - d_{i,s}), \tag{3}$$

where $Z_{i,0}$ denotes the actual amount of inventory at the end of period 0.

Net requirements $b_{i,s}$ result from the following definition/computation

$$b_{i,s} = \max(0, d_{i,s} - Z_{i,s-1} - x_{i,s}). \tag{4}$$

Once net requirements have been computed in time interval R, any lot-sizing rule is applied on these net requirements so as to generate planned orders $p_{i,s}$ representing quantities that will possibly be launched in period s and becoming available in period $s + l_i$. The first planned order, $p_{i,t}$ (recall that t is the first period of time interval R) is then transformed into a firm order $x_{i,t}$. In other words, we set $x_{i,t} = p_{i,t}$ with $x_{i,t}$ corresponding to a quantity that is really ordered in the current period t. Planned orders and firm orders are used to compute the planned gross requirements for components as follows

$$d_{i,s} = \sum_{j \in \Gamma(i)} c_{i,j} \cdot X_{j,s+l_i} \text{ with } X_{j,s+l_i} = \begin{cases} x_{j,s+l_i} \text{if } s + l_i < t \\ p_{j,s+l_i} \text{ otherwise} \end{cases}, \tag{5}$$

where $c_{i,j}$ denotes the production ratio (number of units of i to produce one unit of j).

At the end of period t, planned gross requirements become actual requirements, as they do not depend on planned orders at higher levels anymore but only on firm orders. Let $D_{i,t}$ be the actual gross requirements to be satisfied in period t, we have

$$D_{i,t} = \sum_{j \in \Gamma(i)} c_{i,j} \cdot x_{j,t}. \tag{6}$$

Note that $D_{i,t+l_i} = d_{i,t}$ for all t.

A firm order $x_{i,t}$ becoming available in period $t + l_i$ generates a scheduled receipt $r_{i,t+l_i}$, in that period. Put another way, we have $r_{i,t} = x_{i,t-l_i}$.

At the end of the current period t of the rolling horizon, we are able to compute the actual inventory $I_{i,t}$ (as opposed to the pipeline inventory) for all items. We have

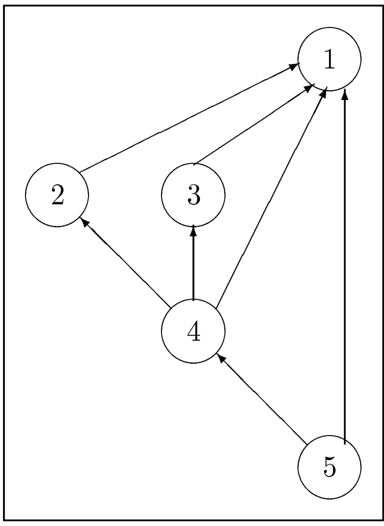
$$I_{i,t} = I_{i,t-1} + r_{i,t} - D_{i,t}. \tag{7}$$

The initial value of the pipeline inventory is equal to the initial inventory. We thus have $Z_{i,0} = I_{i,0}$, $\forall i$.

2.2 The occurrence of a stockout

This subsection presents a situation of stockout due to positive lead times in a multi-level lot-sizing problem with a rolling horizon. Let us consider the example in Figure 1.

The product structure involves 5 items and the complexity index equals $\mathcal{C} = (7-4)/(9-4) = 0.6$. We chose $W = 4$, which means that we apply any lot-sizing rules on the net requirements for 4 periods of deterministic demand. Lead times are only positive for items 3, 4 and 5 with $l_3 = l_4 = l_5 = 1$. We set $I_{i,0} = 0$, $\forall i = 1, \ldots, 5$.



$P = 5 \quad C = 0.6 \quad N = 3 \quad T = 12 \quad W = 4$

Item i	1	2	3	4	5
S_i	8	35	20	32	44
h_i	0.885	0.120	0.127	0.118	0.001
l_i	0	0	1	1	1
L_i	3	2	2	1	0

Figure 1: Data of an example.

We have $Z_{i,0} = I_{i,0}$, $\forall i$. Scheduled receipts for items 3, 4 and 5 take the following values in period 1: $r_{3,1} = 56$, $r_{4,1} = 354$ and $r_{5,1} = 177$. Due to the lead times, scheduled receipts for these items result from ordering decisions prior to the first period of the rolling horizon.

Table 1 exhibits the rolling procedure for the first three decision periods. In the current period $t = 1$, we know the demand for end item 1 in periods 1 to 4, $\{d_{1,1}, \ldots, d_{1,4}\}$, and we must implement launching decisions for all items in period 1. At the beginning of period 1, no ordering decision has been made, so we have $x_{i,1} = 0$, $\forall i$. We first compute the pipeline inventories and the net requirements for item 1, using formulae (3) and (4). In period $s = 1$, we have $Z_{1,1} = \max(0, 0+0-56) = 0$ and $b_{1,1} = \max(0, 56 - 0 - 0) = 56$. Following the same logic, we obtain $Z_{1,s}$ and $b_{1,s}$ for $s = 2, \ldots, 4$. We can now apply any lot sizing rule on the stream of net requirements for the end item. In this example, we have applied the Wagner-Whitin algorithm in a sequential fashion. The algorithm proposes the lot for lot solution for item 1 as $p_{1,s} = b_{1,s}$, $\forall s = 1, \ldots, 4$. We can now compute the planned gross requirements for item 2, using formula (5). We have $d_{2,s} = p_{1,s}$, $\forall s = 1, \ldots, 4$. We then compute the pipeline inventories and the net requirements. The Wagner-Whitin algorithm leads to a unique lot in period 1, with $p_{2,1} = b_{2,1} + \cdots + b_{2,4} = 177$. Planned gross requirements for item 3 result from the following computation: $d_{3,1} = p_{1,2}$, $d_{3,2} = p_{1,3}$ etc. Note that $d_{3,4} = 0$ since $p_{1,5}$ is still set to zero. The Wagner-Whitin again proposes a unique lot in the first period: $p_{3,1} = b_{3,1} + \cdots + b_{3,4} = 121$. Item 4 has the three first items as successors. We have $d_{4,1} = p_{1,2} + p_{2,2} + p_{3,2} = 36 + 0 + 0 = 36$. We follow the same reasoning to compute the next gross requirements, the pipeline inventories and the net requirements. The Wagner-Whitin algorithm leads to a single order in period 1. The same holds for item 5.

At the end of period 1, we set $x_{i,1} = p_{i,1}$ for all items. This means that we actually launch in period 1 the production of 56 units of item 1, 177 units of item 2, etc. For the first two items for which the lead time is zero, these ordering decisions instantaneously produce scheduled receipts in period 1: $r_{1,1} = x_{1,1} = 56$ and $r_{2,1} =$

$x_{2,1} = 177$. We can also compute the actual gross requirements for all items in period 1 and the final inventories, using formulae (6) and (7). We have $D_{1,1} = d_{1,1} = 56$; $D_{2,1} = x_{1,1} = 56$; $D_{3,1} = x_{1,1} = 56$; $D_{4,1} = x_{1,1} + x_{2,1} + x_{3,1} = 56 + 177 + 121 = 354$; $D_{5,1} = x_{1,1} + x_{4,1} = 56 + 121 = 177$. Inventories at the end of period 1 take the following values: $I_{1,1} = 0+56-56 = 0$; $I_{2,1} = 0+177-56 = 121$; $I_{3,1} = 0+56-56 = 0$; $I_{4,1} = 0 + 354 - 354 = 0$; $I_{5,1} = 0 + 177 - 177 = 0$. All these values ($D_{i,t}$, $x_{i,t}$, $r_{i,t}$ and $I_{i,t}$) characterize the state of the system in period t. They are used for cost calculation purposes but are not necessary to make ordering decisions.

In the next decision period ($t = 2$), a new demand is revealed for the end item (that of period 5, with $d_{1,5} = 34$). To compute the net requirements and the pipeline inventories in periods 2 to 5, we need to recompute the pipeline inventory in period 1 and the planned gross requirements in period 1. Indeed, to compute $b_{i,s}$, we need $Z_{i,s-1}$ which depends on $d_{i,s-1}$ (see formulae (4) and (3)). The planned demand $d_{i,s-1}$ may have changed, since some planned orders have been transformed into firm orders as the horizon rolls forward. For the end item, we still have $d_{1,1} = 56$. Inventory $Z_{1,1} = \max(0, 0 + 56 - 56) = 0$. We can now compute $b_{1,s}$ and $Z_{1,s}$ for $s = 2, \ldots, 5$. Applying the Wagner-Whitin algorithm on $\{b_{1,2}, \ldots b_{1,5}\}$ leads to the lot for lot solution. For item 2, we have $d_{2,1} = x_{1,1} = 56$ (according to (5)). Inventory $Z_{2,1} = \max(0, 0 + 177 - 56) = 121$. Once $d_{2,s}$, $b_{2,s}$ and $Z_{2,s}$ have been computed, we apply WW on net requirements which leads to an order $p_{2,5} = 34$, to cover the new net requirement. For item 3, we also have a single net requirement in period 4 which is covered by an order $p_{3,4} = b_{3,4} = 34$. For item 4 we have $d_{4,1} = p_{1,2} + p_{2,2} + p_{3,2} = 36 + 0 + 0 = 36$. Inventory $Z_{4,1} = \max(0, 0 + 121 - 36) = 85$. Applying the same reasoning, we obtain an order of 102 units in period 3 to cover the net requirements of periods 3 and 4. Note that in the first decision period ($t = 1$), a single order of item 4 in period 1 was sufficient to cover the requirements of periods 1 to 3. Now the demand has changed, the former planned order in period 1 is not large enough. A similar remark can be made for item 5: a positive planned order in period 2 is now necessary to cover the new net requirements.

At the end of period 2, we set $x_{i,2} = p_{i,2}$ for all items. We have $D_{1,2} = 36$, $D_{2,2} = x_{1,2} = 36$, $D_{3,2} = x_{1,2} = 36$, $D_{4,2} = x_{1,2} + x_{2,2} + x_{3,2} = 36 + 0 + 0 = 36$ and $D_{5,2} = x_{1,2} + x_{4,2} = 36 + 0 = 36$. Scheduled receipts take the following values: $r_{1,2} = x_{1,2} = 36$, $r_{2,2} = x_{2,2} = 0$, $r_{3,2} = x_{3,1} = 121$, $r_{4,2} = x_{4,1} = 121$ and $r_{5,2} = x_{5,1} = 121$. We finally compute the ending inventories: $I_{1,2} = 0$, $I_{2,2} = \cdots = I_{5,2} = 85$.

In period $t = 3$, demand in period 6 is known for the end item ($d_{1,6} = 36$). As before, we recompute $d_{1,2}$ and $Z_{1,2}$ to determine the net requirements within the forecast window. The Wagner-Whitin algorithm suggests an order for item 1 in period 6 to cover the new demand. For item 2, the planned order in period 5 is now increased to cover the new net requirement of period 6 (in period $t = 2$, we had $p_{2,5} = 34$ and now, in period $t = 3$, we have $p_{2,5} = 70$). For item 3, we need to recompute $d_{3,1}$, as this demand depends on $x_{1,2}$ which has only been determined in the previous period $t = 2$. We have $d_{3,1} = x_{1,2} = 36$. Inventory $Z_{3,1} = \max(0, 0 + 121 - 36) = 85$. In the next period, $d_{3,2} = p_{1,3} = 39$ and $Z_{3,2} = \max(0, 85 + 0 - 39) = 46$. Computing the net requirements within the window and applying WW lead to a single order in period 4 to cover the two positive requirements of item 3. A similar reasoning is applied to item 4. A large planned order is proposed in period 3 with $p_{4,3} = 210$ instead of 102 units which was the planned order computed in the previous period $t = 2$. For item 5, we have $d_{5,1} = x_{1,2} + x_{4,2} = 36 + 0 = 36$ and $Z_{5,1} = \max(0, 0 + 121 - 36) = 85$.

In the next period, $d_{5,2} = p_{1,3} + p_{4,3} = 39 + 210 = 249$. We can immediately observe that this demand cannot be satisfied as the available quantity of item 2 equals $Z_{5,1} + x_{5,2} = 85 + 136 = 221$ which is inferior to $d_{5,2} = 249$. Of course, it is already too late to modify $x_{5,2}$ which has been launched in period 2 that now belongs to the past. Setting $x_{1,3} = p_{1,3}$ and $x_{4,3} = p_{4,3}$ would lead to a stockout of item 5 in period 3, as $I_{5,3} = I_{5,2} + r_{5,3} - D_{5,3} = 85 + 136 - 249 = -28$. Planned orders computed for items 1 to 4 in period $t = 3$ are therefore infeasible since their implementation would lead to a stockout for item 5 in period 3.

	$t=1$				$t=2$					$t=3$					
	1	**2**	**3**	**4**	1	**2**	**3**	**4**	**5**	1	2	**3**	**4**	**5**	**6**
$d_{1,s}$	56	36	39	46	56	36	39	46	34	56	36	39	46	34	36
$b_{1,s}$	56	36	39	46		36	39	46	34			39	46	34	36
$p_{1,s}$	56	36	39	46		36	39	46	34			39	46	34	36
$x_{1,s}$					56					56	36				
$Z_{1,s}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$d_{2,s}$	56	36	39	46	56	36	39	46	34	56	36	39	46	34	36
$b_{2,s}$	56	36	39	46		0	0	0	34			0	0	34	36
$p_{2,s}$	177	0	0	0		0	0	0	34			0	0	70	0
$x_{2,s}$					177					177	0				
$Z_{2,s}$	0	0	0	0	121	85	46	0	0	121	85	46	0	0	0
$d_{3,s}$	36	39	46	0	36	39	46	34	0	36	39	46	34	36	0
$b_{3,s}$	36	39	46	0		0	0	34	0			0	34	36	0
$p_{3,s}$	121	0	0	0		0	0	34	0			0	70	0	0
$x_{3,s}$					121					121	0				
$Z_{3,s}$	0	0	0	0	85	46	0	0	0	85	46	0	0	0	0
$d_{4,s}$	36	39	46	0	36	39	80	68	0	36	39	116	104	36	0
$b_{4,s}$	36	39	46	0		0	34	68	0			70	104	36	0
$p_{4,s}$	121	0	0	0		0	102	0	0			210	0	0	0
$x_{4,s}$					121					121	0				
$Z_{4,s}$	0	0	0	0	85	46	0	0	0	85	46	0	0	0	0
$d_{5,s}$	36	39	46	0	36	141	46	34	0	36	249	46	34	36	0
$b_{5,s}$	36	39	46	0		56	46	34	0						
$p_{5,s}$	121	0	0	0		136	0	0	0						
$x_{5,s}$					121					121	136				
$Z_{5,s}$	0	0	0	0	85	0	0	0	0	85					

$d_{i,s}$: gross requirements for item i in period s, $b_{i,s}$: net requirements,
$p_{i,s}$: planned orders, $Z_{i,s}$: pipeline inventory, $\Gamma(i)$: set of successors.
$\Gamma(1) = \emptyset$, $\Gamma(2) = \{1\}$, $\Gamma(3) = \{1\}$, $\Gamma(4) = \{1, 2, 3\}$, $\Gamma(5) = \{1, 4\}$.
l_i: lead time. $l_1 = 0$, $l_2 = 0$, $l_3 = 1$, $l_4 = 1$, $l_5 = 1$.

Table 1: A situation of stockout.

We are now able to provide a formal definition of the feasibility of planned orders. Planned orders $\{p_{i,t}, \ldots, p_{i,t+W-1}\}$ for all items in period t are feasible if and only if

$$Z_{i,k-1} + x_{i,k} \geq d_{i,k} \ \forall\, i | \, \Gamma(i) \neq \emptyset \text{ and } \forall k \in \{t - l_i, \ldots, t-1\}. \tag{8}$$

In such a case, all planned orders $\{p_{i,t}\}_i$ in period t can be transformed into firm orders $\{x_{i,t}\}_i$. On the contrary, if the previous inequality is not satisfied for one item i in one period k, some planned orders for some successors of item i should be modified as they trigger a too large demand for item i which can not be satisfied on the basis of previous orders and stocks.

To check inequality (8), we only need to recompute the gross requirements and the pipeline inventories in time interval $\{t-l_i, \ldots, t-1\}$, using formulae (5) and (3).

The question is now how to get a feasible stream of orders when we face a stockout? Past literature usually leaves aside the question of positive lead times when demand is known with certainty as positive lead times are supposedly 'without any effect' in a deterministic framework. The next section presents two methods to avoid stockouts in this environment.

3 Avoiding stockout situations

The first subsection reviews the safety stock approach which is traditionally used in a context of stochastic demand. Subsection 2 is dedicated to a detailed description of our repair procedure.

3.1 Safety stock strategies

Safety stock policies are widely employed in a stochastic context. De Bodt et al. (1982) pointed out the importance of carefully designing a safety stock policy to reach a good service level. But the question of the location and size of these safety stocks is awkward and remains widely open.

In several studies examining the performance of lot-sizing techniques under demand uncertainty, safety stocks are computed recursively so as to reach a 100% service level for cost comparisons purposes (see for instance Bookbinder and Heath, 1988). Safety stocks are introduced at all levels in the product structure and whenever a stockout is about to occur the safety stock value is increased so as to avoid lost demands. To illustrate, let us consider our example in figure 1. We apply the WW algorithm in a sequential fashion with a forecast window of 4 periods and we stop the rolling horizon at period $T = 8$. Demand for the end item equals $\{56, 36, 39, 46, 34, 36, 5, 57\}$. If we do not introduce any safety stock, we obtain the following situation for component 5 (Table 2).

Period t	1	2	3	4	5	6	7	8
Actual gross requirements $D_{5,t}$	177	36	249	46	189	41	0	57
Scheduled receipts $r_{5,t}$	177	121	136	116	0	98	0	0
Ending inventory $I_{5,t}$	0	85	-28	70	-119	57	57	0

Table 2: Situation for component 5 without safety stock; end of run 1.

As before (see Table 1), a stockout of 28 units appears in period 3. Another stockout of 119 units is recorded in period 5. To set the value of the safety stock for item 5 (SS_5), we simply take the maximum stockout along the horizon. We have $SS_5 = 119$. Table 3 provides the values of gross requirements ($d_{i,s}$), net requirements ($b_{i,s}$), planned orders ($p_{i,s}$) and pipeline inventories ($Z_{i,s}$) for item $i = 5$ in the first three periods of the rolling horizon when the safety stock value for this item is set to 119 units (we also have $SS_1 = SS_2 = SS_3 = 0$ and $SS_4 = 31$).

Period	**1**	**2**	**3**	**4**
Gross requirements $d_{5,s}$	36	39	46	0
Net requirements $b_{5,s}$	155	39	46	0
Planned orders $p_{5,s}$	240	0	0	0
Lot size $x_{5,s}$	0	0	0	0
Pipeline Inventory $Z_{5,s}$	0	0	0	0

Period	1	**2**	**3**	**4**	**5**
Gross requirements $d_{5,s}$	36	141	46	34	0
Net requirements $b_{5,s}$		56	46	34	0
Planned orders $p_{5,s}$		136	0	0	0
Lot size $x_{5,s}$	240				
Pipeline Inventory $Z_{5,s}$	204	0	0	0	0

Period	1	2	**3**	**4**	**5**	**6**
Gross requirements $d_{5,s}$	36	249	46	34	36	0
Net equirements $b_{5,s}$			74	34	36	0
Planned orders $p_{5,s}$			144	0	0	0
Lot size $x_{5,s}$	240	136				
Pipeline inventory $Z_{5,s}$	204	91	0	0	0	0

Table 3: Rolling procedure for item 5 with $SS_5 = 119$; run 2.

Computation of net requirements now includes the safety stock value. Formula (4) simply becomes

$$b_{i,s} = \max(0, d_{i,s} + SS_i - Z_{i,s-1} - x_{i,s}). \tag{9}$$

In period $t = 1$, we have $b_{5,1} = \max(0, 36 + 119 - 0 - 0) = 155$. In the next periods, net requirements equal gross requirements, as we need to order the safety stock value only once within the forecast window. Table 4 provides the same data as Table 2 at the end of run 2.

Period t	1	2	3	4	5	6	7	8
Actual gross requirements $D_{5,t}$	208	36	249	46	220	41	0	57
Scheduled receipts $r_{5,t}$	208	240	136	144	0	217	0	0
Ending inventory $I_{5,t}$	0	204	91	189	-31	145	145	88

Table 4: Situation for component 5 with $SS_5 = 119$; end of run 2.

Another stockout of 31 units is recorded in period 5 and originates in the safety stock required by item 4 at the beginning of run 2 ($SS_4 = 31$). Safety stock for item 5 ($SS_5 = 119$) did not take into account the need for a safety stock of 31 units for item 4. Safety stock values must be updated: we add to the previous safety stock value the maximum stockout recorded along the horizon. In our case, we set

$SS_5 := SS_5 + 31 = 150$. Another run of the rolling procedure produces the final results for item 5 (see Table 5).

Period t	1	2	3	4	5	6	7	8
Actual gross requirements $D_{5,t}$	208	36	249	46	220	41	0	57
Scheduled receipts $r_{5,t}$	208	271	136	144	0	248	0	0
Ending inventory $I_{5,t}$	0	235	122	220	0	207	207	150

Table 5: Situation for component 5 with $SS_5 = 150$; end of run 3.

Stockouts no longer appear. It should be noted that scheduled receipts in period 1 have switched from 177 units to 208 units to cover the first gross requirements. This change has no incidence on cost results as scheduled receipts in period 1 are just meant to cover exactly the first requirement; they do not incur neither a set up cost (they do not result from ordering decisions made within the current horizon but prior to the first period of the horizon), nor carrying costs (they never exceed requirements). The way initial scheduled receipts are set should just be seen as a programming trick.

3.2 A repair procedure

A more sophisticated solution to avoid shortages would be to track down the items whose lot sizes are responsible for a stockout of a given item. In our example this would amount to consider the successors of item 5 and then to detect the lot sizes to be decreased. Item 5 has two successors: item 1 and item 4. As the lot for lot solution is proposed for item 1, none of these lots may be decreased. Item 1 is therefore not responsible for the stockout of item 5 in period 2. The lot size of 210 units of item 4 in period 3 is the only candidate to depletion. To obtain a feasible solution in period 3, it suffices to decrease the lot size of item 4 in period 3 from the demand of 36 units in period 6 and to place an order of 36 units in period 6. In this example, the way to proceed is straightforward but more ambiguous situations may be encountered. For instance, if both items 1 and 4 were candidates to lot sizes depletion, the question of how to select these items and how to decrease the corresponding orders would be open. Of course in assembly systems the question of how to choose a candidate to depletion does not arise. But in highly complex structures relevancy of such a question rises. One way to get around this question is to transform the rolling schedule problem into a 'capacitated' problem. This provides a natural repair procedure to obtain a feasible schedule from an infeasible one.

3.2.1 Keynote and illustration of the procedure

To illustrate, let us consider our example in Table 1. Observe first that in period $t = 2$, a planned order of 102 units of item 4 in period 3 triggers (together with an order of 39 units of item 1 in period 3) a gross requirement of 141 units of item 5 in period 2. This triggers in turn a net requirement for item 5 which may be covered by a feasible order in period 2. If, in period $t = 3$, we force the order of item 4 in period 3 not to exceed 102 units, we can obtain a feasible schedule for all items in period 3. But this condition is not sufficient. Gross requirements for items 3 and 4 in period 2 should not exceed 39 units. No condition is required on gross requirements for item 2 as its lead time is zero. Only its gross requirements in the current period

t may change so it is never too late to order a proper amount. For the end item, gross requirements always keep the same value from one period to the next (in this deterministic framework) so no stockout can be expected. One way to guarantee the satisfaction of such conditions on gross requirements is to force, in the current period t, all planned orders within the cumulative lead time not to exceed their previous values (computed in period $t-1$). This writes

$$\sum_{k=t}^{l} p_{i,k}^{t} \leq \sum_{k=t}^{l} p_{i,k}^{t-1} \text{ for all } l = t, \ldots, t + L_i - 1. \tag{10}$$

To illustrate, consider our example in Table 1. At the end of period 2 we record the values of planned orders within the cumulative lead time. We record for item 1 $\{p_{1,3}^2, p_{1,4}^2, p_{1,5}^2\} = \{39, 46, 34\}$, for item 2 $\{p_{2,3}^2, p_{2,4}^2\} = \{0, 0\}$, for item 3 $\{p_{3,3}^2, p_{3,4}^2\} = \{0, 34\}$ and for item 4 $\{p_{2,4}^2\} = \{102\}$. In period 3, we first apply WW without any restriction on the value of planned orders. As we face a stockout, we re-apply WW together with conditions (10) on all items so as to obtain a feasible schedule. For item 1, we first set $p_{1,3}^3 = 39$ which verifies inequality $p_{1,3}^3 \leq p_{1,3}^2$. We then evaluate two alternatives: 'ordering in period 3 the demand for periods 3 and 4' versus 'ordering in period 4 the demand for period 4 and adopting the optimal policy for the previous period'. Option 1 leads to $p_{1,3}^3 = 85$, $p_{1,4}^3 = 0$ but these orders must verify $p_{1,3}^3 \leq p_{1,3}^2$ and $p_{1,3}^3 + p_{1,4}^3 \leq p_{1,3}^2 + p_{1,4}^2$. As they only satisfy the second condition, they are abandoned and the second option ($p_{1,3}^3 = 39, p_{1,4}^3 = 46$), which verifies both inequalities, is adopted as the optimal solution for the second period. Three options are then evaluated: $p_{1,3}^3 = 119$, $p_{1,4} = 0$, $p_{1,5} = 0$ or $p_{1,3}^3 = 39$, $p_{1,4} = 80$, $p_{1,5} = 0$ or $p_{1,3}^3 = 39$, $p_{1,4} = 46$, $p_{1,5} = 34$. The cheapest option that verifies $p_{1,3}^3 \leq p_{1,3}^2$ and $p_{1,3}^3 + p_{1,4}^3 \leq p_{1,3}^2 + p_{1,4}^2$ and $p_{1,3}^3 + p_{1,4}^3 + p_{1,5}^3 \leq p_{1,3}^2 + p_{1,4}^2 + p_{1,5}^2$ is selected as the optimal solution for a horizon of three periods. Only the second schedule obviously fits these criteria and is selected as the optimal plan. In the next period we evaluate the following alternatives: 'ordering a single lot in period 3', 'ordering $b_{1,4} + b_{1,5} + b_{1,6}$ in period 4 and adopting the optimal policy for period 3', 'ordering $b_{1,5} + b_{1,6}$ in period 5 and adopting the optimal policy for periods 3 and 4', 'ordering $b_{1,6}$ in period 6 and adopting the optimal policy for periods 3 to 5'. The last option is selected as the optimal plan. If we had $W = 5$ we would have an additional demand in period 7 and as the optimal previous date of order is period 6 which is outside the cumulative lead time (i.e. the interval in which capacities are considered) a 'regular' WW would have been applied from period 6 while keeping the optimal plan from periods 3 to 5. This limits the number of undesirable plans to be evaluated and so the computational requirements of the algorithm. In the following, we use a simpler notation for capacities; we set $C_{i,s} = p_{i,s}^{t-1}$. Table 6 provides the feasible schedules for all items in period 3, resulting from the application of the 'capacitated' WW.

Now we have laid the basis of our repair procedure we shall present the full procedure in a more formal way in the next paragraph.

3.2.2 Formal presentation

Table 7, while giving the initial value of the variables, provides a useful summary of the notation employed in the paper.

Table 8 lists the operations to be performed at the first iteration for all items. The horizon is limited to the forecast window W within which we compute the gross

				$t=3$			
	Period	1	2	3	4	5	6
$\Gamma(1)=\emptyset$	$d_{1,s}$	56	36	39	46	34	36
$l_1=0$	$b_{1,s}$			39	46	34	36
	$p_{1,s}$			39	46	34	36
	$C_{1,s}$			**39**	**46**	**34**	
	$x_{1,s}$	56	36				
	$Z_{1,s}$	0	0	0	0	0	0
$\Gamma(2)=\{1\}$	$d_{2,s}$	56	36	39	46	34	36
$l_2=0$	$b_{2,s}$			0	0	34	36
	$p_{2,s}$			0	0	70	0
	$C_{2,s}$			**0**	**0**		
	$x_{2,s}$	177	0				
	$Z_{2,s}$	121	85	46	0	0	0
$\Gamma(3)=\{1\}$	$d_{3,s}$	36	39	46	34	36	0
$l_3=1$	$b_{3,s}$			0	34	36	0
	$p_{3,s}$			0	34	36	0
	$C_{3,s}$			**0**	**34**		
	$x_{3,s}$	121	0				
	$Z_{3,s}$	85	46	0	0	0	0
$\Gamma(4)=\{1,2,3\}$	$d_{4,s}$	36	39	80	140	36	0
$l_4=1$	$b_{4,s}$			34	140	36	0
	$p_{4,s}$			34	176	0	0
	$C_{4,s}$			**102**			
	$x_{4,s}$	121	0				
	$Z_{4,s}$	85	46	0	0	0	0
$\Gamma(5)=\{1,4\}$	$d_{5,s}$	36	73	222	34	36	0
$l_4=1$	$b_{5,s}$			74	34	36	0
	$p_{5,s}$			144	0	0	0
	$x_{5,s}$	121	136				
	$Z_{5,s}$	85	148	0	0	0	0

Table 6: Application of the repair procedure ('capacitated' WW).

for $i=1,\ldots,P$	
for $s=1,\ldots,T$	
$d_{i,s}=0$	Gross requirements that must be satisfied in period $s+l_i$
$x_{i,s}=0$	Firm order launched in period s
$Z_{i,s}=0$	Pipeline inventory at the end of period s
$b_{i,s}=0$	Net requirements that must be satisfied in period $s+l_i$
$p_{i,s}=0$	Planned order that will possibly be released in period s
$I_{i,s}=0$	'Actual' ending inventory in period s
$r_{i,s}=0$	Scheduled receipts in period s resulting from orders really launched in period $s-l_i$

Table 7: Initialization of variables and summary of notation.

requirements, then the pipeline inventory and the net requirements. Once net requirements have been calculated, we apply any lot-sizing rule to determine planned orders. Only the first planned order is transformed into a firm order that triggers scheduled receipts l_i period(s) later. We then record the capacities that will possibly be used in case the solution at the next iteration is infeasible. In time interval $\{1, \ldots, l_i\}$ scheduled receipts must be considered as data for they result from lot-sizing decision that took place before period 1. Nonetheless we set them to a value that make feasible any solution. By doing so, all production possibilities may take place within the cumulative lead time for each item. To smooth away the periods in which no production may happen, some authors (e.g. Wemmerlöv and Whybark, 1984) simply choose to record no statistics for a given number of periods called a start-up period.

---

for $i = 1, \ldots, P$

- for $s = 1, \ldots, W$
  Compute $d_{i,s}$ using equation (5)
  Compute $Z_{i,s}$ using equation (3)
  Compute $b_{i,s}$ using equation (4)
- Apply any rule on $\{b_{i,s}\}_{s=1,\ldots,W}$
  (we thus obtain $\{p_{i,s}\}_{s=1,\ldots,W}$)
- set $x_{i,1} = p_{i,1}$
  $r_{i,1+l_i} = x_{i,1}$
- for $s = 2, \ldots, 1 + L_i$
  set $C_{i,s} = p_{i,s}$

Initial values for scheduled receipts
for $i = 1, \ldots, P$
for $s = 1, \ldots, l_i$
$$r_{i,s} = \sum_{j \in \Gamma(i)} p_{j,s}$$

Table 8: First iteration ($t = 1$).

Table 9 presents the procedure at any iteration $t > 1$ (left side of the table). In each period t we compute the gross requirements of period $t - l_i - 1$. In this past period, gross requirements no longer result from planned orders but from firm orders since we have $d_{i,t-l_i-1} = \sum_{j \in \Gamma(i)} c_{i,j} \cdot x_{j,t-1}$ and $x_{j,t-1}$ for each item j has been determined in the previous period.[1] In time interval $\{t - l_i, \ldots, t - 1\}$ demand is recomputed as it results from planned orders in periods t to $t + l_i - 1$ but in time interval $\{t - l_i, \ldots, t - 1\}$, orders have already been launched so demand should not exceed the material availability. In other words, we check inequality (8). When this inequality is not verified, we implement the repair procedure exhibited in the right

[1] Note that instead of period $t - l_i - 1$ we should speak about period $\max(1, t - l_i - 1)$. A similar remark may be done for period $t + W - 1$, it should be period $\min(t + W - 1, T)$.

side of Table 9. We shall comment this sub-procedure later. When the feasibility condition is satisfied we determine the net requirements within the window and we apply any lot-sizing rule. If all items have passed the feasibility test, we implement the first plan order by setting $x_{i,t} = p_{i,t}$, for all items and we set the scheduled receipts to the appropriate value. We finally record the capacities.

MAIN LOOP	REPAIR

for $i = 1, \ldots, P$
  for $t = 2, \ldots, T$
    Recompute $d_{i,t-l_i-1}$
    Recompute $Z_{i,t-l_i-1}$
    for $s = t - l_i, \ldots, t-1$
      Compute $d_{i,s}$
      if $Z_{i,s-1} + x_{i,s} < d_{i,s}$
        go to Repair
      else
        Compute $Z_{i,s}$
        Compute $b_{i,s}$
    for $s = t, \ldots, t+W-1$
      Compute $d_{i,s}$
      Compute $Z_{i,s}$
      Compute $b_{i,s}$
    Apply any rule on $\{b_{i,s}\}_{s=t,\ldots t+W-1}$
    (we obtain $\{p_{i,s}\}_{s=t,\ldots,t+W-1}$)
    if $i = P$
      for $j = 1, \ldots, P$
        set $x_{j,t} = p_{j,t}$
        set $r_{j,t+l_j} = x_{j,t}$
        for $s = t+1, \ldots t + L_j$
          $C_{j,s} = p_{j,s}$
	for $i = 1, \ldots, P$
	  for $s = t - l_i - 1, \ldots, t+W-1$
	    Compute $d_{i,s}$
	    Compute $Z_{i,s}$
	    Compute $b_{i,s}$
	  if $L_i > 0$
	    Apply any rule on $\{b_{i,s}\}_{s=t,\ldots,t+W-1}$
	    with respect to the capacities $\{C_{i,s}\}_{s=t,\ldots,t+L_i-1}$
	    (we obtain $\{p_{i,s}\}_{s=t,\ldots,t+W-1}$)
	  else
	    Apply any rule on $\{b_{i,s}\}_{s=t,\ldots,t+W-1}$
	  set $x_{i,t} = p_{i,t}$
	  set $r_{i,t+l_i} = x_{i,t}$
	  for $s = t+1, \ldots, t+L_i$
	    $C_{i,s} = p_{i,s}$

Table 9: Main loop and repair procedure.

The repair procedure is implemented as soon as an item does not fit the feasibility condition. For all items, we determine the net requirements and then apply any rule which has been modified to meet conditions (10). We shall give further details on the capacitated version of the lot-sizing rules in the next section. At the end of the whole procedure, we compute the ending inventories $I_{i,t}$ for all items and all periods. We then compute the total cost associated with a given lot-sizing rule.

4 Experimental framework

4.1 Lot-sizing procedures

We selected six single-level lot-sizing rules for use in the study. The Wagner-Whitin algorithm (1958), the Silver-Meal technique (1972), the incremental part period algorithm (1968), the Silver version of the economic order quantity (1976), the periodic order quantity and the least unit cost method. Despite the development of heuristics specifically designed to account for interdependencies among stages, the chosen heuristics are still widely adopted in practice under rolling and multi-level conditions.

The Wagner-Whitin algorithm (WW) provides the optimal solution to the single-level lot-sizing problem by use of dynamic programming. The economic order quantity (EOQ) is the traditional Wilson lot-sizing model which balances the inventory carrying costs and order costs. The Silver version of this method lumps an integer number of future demands closest to the EOQ value. The periodic order quantity (POQ) uses the EOQ to determine the reorder time cycle and then orders what is actually forecasted for that time cycle. The incremental part period algorithm (IPPA) increases the size of the order until carrying costs are equal or less than the set-up cost. The Silver-Meal heuristic selects the order quantity so as to minimize the cost per unit time over the time periods the order lasts. The least unit cost (LUC) selects the order quantity so as to minimize the cost per unit (cumulation of the requirements until the cost per unit starts to increase).

Each technique is either combined with a safety stock strategy as described in paragraph 3.1 or with our repair procedure as detailed in paragraph 3.2.1. To obtain a 'capacitated' version of the heuristics, we have applied the same principles as those employed for WW in paragraph 3.2.1. Each heuristic simply lumps the demands until the criterion is minimized unless the lot size exceeds the available capacity.

Our incentive is to compare the cost performance of our repair procedure to the costs produced by the safety stock strategy which was specifically designed to reach the same service as our repair, say 100% (see paragraph 3.1). From a practical standpoint, this safety stock strategy is not ready for use, as the whole story of demands for end items needs to be known to set the appropriate values of safety stocks. Safety stocks are therefore artificially set to a minimum value for which there is not a single lost demand for end items. In practice however, safety stocks are decided beforehand and a 100% service level can only be achieved at the price of extremely large amounts of safety stocks. In that respect, our peculiar implementation of the safety stock strategy (where levels of stocks are determined retrospectively) tends to sanction the performance of our repair procedure. However, comparing our repair procedure to more operational safety stock policies would have led to a ticklish bicriteria comparison (cost and service level). To avoid this, while comparing our repair to an existing method, we chose this specific safety stock strategy for lack of existing operational techniques preventing stockout situations.

4.2 Demand generation

Although several end items could have been included in the present study we chose to take only one end item per product structure for the sake of simplicity. Demand for the end item $\{d_s\}_{s=t,\ldots,t+W-1}$ may follow different patterns. We chose to generate demand for the end item from a uniform distribution of mean 50. We have $d_s \sim U[0, 2\overline{d}]$. Standard deviation of demands therefore equals $\left(1/12(100-0)^2\right)^{1/2} = 28.86$. We also chose to use a normal distribution with same average and standard deviation: $d_s \sim N[50, 28.86]$.

4.3 Cost generation

In accordance with the assumption of value-added holding costs, carrying costs for each item were defined as follows

$$h_i = e_i + \sum_{j \in \Gamma^{-1}(i)} c_{j,i} \cdot h_j \text{ with } e_i = 0.0005 + 0.02 \times u,$$

where u is selected from a uniform distribution between 0 and 1.

We used the TBO (time between orders) factor to determine set-up cost parameters. TBO values are usually larger for components than they are for end items and subassemblies. Indeed, successful JIT principles implementation in many companies has lead to substantial set up cost reductions for these items whereas freedom of action is still limited for components at deeper levels and purchased items. We therefore consider TBO values of 1 and 2 for the end item and subassemblies (TBO_l) and TBO values of 2 and 6 for products at deeper levels (TBO_h). We then have the following combinations: $(TBO_l, TBO_h) = \{(1, 2), (2, 2), (1, 6), (2, 6)\}$. For the sake of curiosity, we also examine $(TBO_l, TBO_h) = (4, 6)$.

For each item we set

$$\begin{aligned} S_i &= \begin{cases} 0.5 \cdot \overline{d}_i \cdot h_i \cdot TBO_l^2 & \text{if } i \text{ is at level 0 or 1,} \\ 0.5 \cdot \overline{d}_i \cdot h_i \cdot TBO_h^2 & \text{otherwise,} \end{cases} \\ \text{with } \overline{d}_i &= \sum_{h \in \Gamma(i)} c_{i,h} \cdot \overline{d}_h. \end{aligned}$$

4.4 Other parameters

In our experiment, we considered structures with one end item made of 9 components ($P = 10$) distributed over four levels $(0, 1, 2, 3)$. Product structures were defined in terms of the complexity index $\mathcal{C}$. We chose 4 values of $\mathcal{C}$ in the set $\{0.00, 0.25, 0.50, 0.75\}$. We stopped the rolling horizon at period 100 ($T = 100$). Execution time of WW legitimizes the choice of instances of moderate size. For instance to solve one rolling problem with $P = 10$ and $T = 100$, each single-level technique is applied approximately $P \times T = 1000$ times and determination of appropriate values for safety stocks requires on average 3 passes or runs of the full rolling procedure. Thus, to get cost observations associated with one technique and one scenario, we need to apply the concerned technique $3 \times P \times T = 3000$ times. Thus, despite the development of ever more powerful computers, computational disadvantage of WW is unmistakable under multi-level rolling conditions.

Two levels of lead times—low and high—were selected. In the first case, lead times for components with no predecessor were uniformly drawn at random in the set $\{1, 2, 3\}$ and in the set $\{0, 1\}$ for any other item. In the second case, lead times for components were drawn in the set $\{2, \ldots, 6\}$ and in the set $\{1, 2\}$ for items at higher levels. Resulting cumulative lead times were therefore higher in the latter situation. Forecast window lengths were chosen accordingly: $W = \{6, \ldots, 13\}$ for low lead times and $W = \{11, \ldots, 18\}$ for high lead times. In this way, we avoid instances in which cumulative lead times exceed the forecast window, a situation requiring the introduction of safety stocks even in our repair procedure since some productions can not be launched in time within the forecast window to meet some requirements. .

Each of the $2 \times 5 \times 4 \times 2 \times 8 = 640$ experiments (2 demand patterns, 5 levels of TBO, 4 complexities, 2 types of lead times, 8 window lengths) was replicated 5 times, therefore providing 3200 cost observations per lot-sizing rule.

5 Simulation results

Figure 2 displays the average cost deviations relative to WW when combined to our repair procedure (namely WWRP). This technique was used as a benchmark as it provides the best overall performance. Each rule is appended with RP when used with the repair procedure and with SS when associated with the safety stock policy. Cost observations for each rule and each scenario were divided by the cost of WWRP. The resultant cost ratios were averaged over all cases with a same factor value. For instance POQRP provides a cost that is 1.06% higher than the cost of WWRP for all cases with assembly structures ($\mathcal{C} = 0.00$).

	WW		IPPA		POQ		SM		EOQ		LUC	
	RP	SS	RP	SS	RP	SS	RP	SS	RP	SS	RP	SS
OVERALL	0.00	13.53	0.92	1.45	1.43	9.89	3.38	2.38	6.03	33.20	19.92	11.01
C=0.00	0.00	16.76	-0.01	2.80	1.06	13.33	3.86	3.30	4.31	43.78	12.86	6.78
C=0.25	0.00	15.64	1.28	2.22	1.08	10.31	3.27	3.08	6.82	33.35	21.27	12.65
C=0.50	0.00	11.89	0.69	-0.37	1.34	8.20	2.51	0.59	5.89	27.62	22.51	10.62
C=0.75	0.00	9.83	1.71	1.14	2.26	7.72	3.87	2.54	7.12	28.05	23.02	14.01
TBO=(1,2)	0.00	19.83	0.30	0.26	9.29	14.96	0.62	0.33	14.36	47.09	14.77	14.85
TBO=(1,6)	0.00	14.01	2.50	2.74	7.56	17.39	2.88	2.20	15.77	62.36	12.48	10.48
TBO=(2,2)	0.00	9.45	0.71	-0.12	-3.10	1.98	4.96	3.72	-0.49	9.92	21.80	13.23
TBO=(2,6)	0.00	10.47	0.87	0.53	-2.43	4.48	2.49	1.01	2.18	26.13	19.88	8.65
TBO=(4,6)	0.00	13.89	0.21	3.85	-4.14	10.66	5.94	4.63	-1.65	20.50	30.65	7.86
Unif. Dem.	0.00	12.79	1.37	2.00	3.18	13.43	3.92	2.88	7.44	35.82	21.70	12.62
Norm. Dem	0.00	14.27	0.46	0.90	-0.31	6.36	2.84	1.88	4.63	30.57	18.13	9.41
Low l_i	0.00	11.72	0.31	1.56	0.78	8.52	2.33	1.49	4.80	26.43	17.51	10.02
High l_i	0.00	15.34	1.52	1.34	2.09	11.27	4.42	3.27	7.26	39.97	22.32	12.00
W=6	0.00	4.72	-4.99	-6.87	-4.87	-2.77	-5.78	-12.65	-1.25	9.58	17.32	-3.79
W=7	0.00	12.95	-5.33	-1.80	-4.70	6.89	-6.68	-7.14	-0.78	23.55	19.57	2.46
W=8	0.00	15.58	-3.75	-1.20	-2.98	10.39	-3.71	-3.49	0.74	26.57	17.93	6.16
W=9	0.00	14.80	-0.96	0.83	-0.65	8.93	0.45	0.62	3.17	29.31	16.07	9.87
W=10	0.00	11.73	1.34	2.79	0.84	7.72	3.88	4.01	5.24	28.44	15.17	12.60
W=11	0.00	10.57	3.50	4.44	3.03	10.30	7.20	7.25	7.58	29.37	15.60	15.03
W=12	0.00	11.83	5.62	6.47	6.35	12.22	10.43	10.45	10.58	30.18	18.32	17.79
W=13	0.00	11.55	7.09	7.84	9.18	14.48	12.85	12.85	13.14	34.40	20.09	20.08
W=11	0.00	6.52	-5.42	-10.97	-4.27	-0.36	-5.15	-13.13	-1.27	20.77	19.42	-4.21
W=12	0.00	16.47	-4.85	-5.01	-3.23	6.07	-4.57	-5.99	1.04	31.05	26.02	4.06
W=13	0.00	18.08	-2.93	-2.05	-2.00	10.13	-1.36	-1.53	2.83	37.38	24.86	8.23
W=14	0.00	17.38	0.86	1.78	0.98	12.79	3.52	3.65	6.14	40.79	22.08	13.07
W=15	0.00	17.41	3.77	4.80	3.87	13.60	7.42	7.51	9.15	44.69	21.96	16.37
W=16	0.00	16.46	5.70	6.40	5.63	14.87	10.09	10.16	11.74	47.79	21.58	18.21
W=17	0.00	15.81	7.01	7.50	7.19	16.35	12.02	12.05	13.68	48.49	21.41	19.62
W=18	0.00	14.61	7.99	8.26	8.60	16.67	13.43	13.44	14.82	48.82	21.23	20.68

Figure 2: Average cost deviations relative to benchmark.

Figure 2 shows the superiority of the repair procedure over the safety stock strategy for all techniques, except LUC and SM. For these rules, the safety stock policy provides better results than our repair. But the difficulties of implementing such a policy in practice should not be forgotten. WWRP offers the best overall performance and the popular POQ and IPPA behave pretty well when associated with our repair.

The Kruskal-Wallis ANOVA test was used to appraise the impact of factor levels on the relative cost performance. Performance of all techniques is affected by factor values except in a few isolated cases. With the exception of SM, relative performance of all rules combined with the repair tends to degrade when the level of complexity is increased. This is probably due to the number of times the repair procedure is applied. This number undoubtedly increases with the number of common parts and each time the repair is applied, the 'optimality principle' of each rule is disrupted. It should be noted that IPPARP produces costs as good as those of WWRP for pure assembly structures. The ANOVA test revealed that the relative performance of POQRP—apparently decreasing with $\mathcal{C}$—is actually not affected by the complexity (we were not able to reject the null hypothesis of equality at the 5% level). EOQRP whose performance decreases as soon as $\mathcal{C}$ becomes positive is no longer affected by positive levels of complexity (for $\mathcal{C} > 0$, the critical probability equals 5.80%).

The relationship between product structures complexity and performance of the techniques associated with the safety stock strategy is unclear: for some rules (like WWSS, POQSS and EOQSS), relative performance continuously improves as $\mathcal{C}$ is augmented, for others (like IPPASS and SMSS) the influence of this factor is undefined whereas LUCSS exhibits a better behaviour for less complex product structures.

To analyze the impact of TBO levels on cost performances, we used the Wilcoxon test to compare pairs of cost ratio samples associated with meaningful pairs of TBO levels. For instance, switching from $TBO = (1, 2)$ to $TBO = (1, 6)$ corresponds to a TBO increase for components only. In such a situation, the Wilcoxon test will tell, for any rule, whether cost ratios are significantly different or not under these two configurations. To complement the analysis, we utilized a sign test whenever the Wilcoxon test suggested to reject the null hypothesis of equality. Table 10 summarizes the test results for each rule under two major cases: TBO is augmented for components only; TBO is increased for end items and subassemblies only. Each case is characterized by a subset of pairs of TBO reported in the second row of table 10. A TBO increase for components is defined by a change of TBO from $TBO = (1, 2)$ to $(2, 2)$ as well as a change from $TBO = (2, 2)$ to $(2, 6)$. Similarly, a switch of TBO from $(1, 2)$ to $(2, 2)$ or $(1, 6)$ to $(2, 6)$ corresponds to a TBO increase for end items and subassemblies only. Intuitively, we would have expected a clear inverse relationship between TBO levels and the performance of our repair procedure as it favours more orders and less inventory. Thus a TBO increase (or equivalently an increase of set up cost parameters) should then tend to degrade the performance of our repair procedure. Table 10 shows that relative performance of IPPARP and EOQRP decreases when TBO for components is augmented whereas it improves when TBO for end items is increased. The same effect is obtained for EOQSS. In most situations, the impact of TBO increases remains indefinite. When a TBO increase does not alter the relative performance (this occurs for instance for IPPARP for a TBO change from $(2, 2)$ to $(2, 6)$), we reported the critical probability (pc) in percent resulting from the Wilcoxon test in Table 10. A sign test performed on gross cost data revealed that set up costs are indeed significantly higher for RP procedures than those produced by SS methods. This only means the share of set ups costs in total cost is higher for RP procedures. This fact however does not provide any useful information to predict the nature of the relationship between TBO and relative cost performance of

RP techniques as outlined by results in Table 10.

	$\nearrow$TBO **for components**			$\nearrow$TBO **for end items and subass.**				
	$(1,2) \downarrow (1,6)$	$(2,2) \downarrow (2,6)$	Result	$(1,2) \downarrow (2,2)$	$(1,6) \downarrow (2,6)$	$(1,6) \downarrow (4,6)$	$(2,6) \downarrow (4,6)$	Result
IPPARP	$\searrow$	= (69.97)	$\searrow$	= (25.00)	$\nearrow$	$\nearrow$	$\nearrow$	$\nearrow$
POQRP	$\nearrow$	$\searrow$	indef.	$\nearrow$	$\nearrow$	$\nearrow$	$\searrow$	indef.
EOQRP	$\searrow$	$\searrow$	$\searrow$	$\nearrow$	$\nearrow$	$\nearrow$	$\nearrow$	$\nearrow$
SMRP	$\searrow$	$\nearrow$	indef.	$\searrow$	= (29.56)	$\searrow$	$\searrow$	$\searrow$
LUCRP	$\nearrow$	$\nearrow$	$\nearrow$	$\searrow$	$\searrow$	$\searrow$	$\searrow$	$\searrow$
WWSS	$\nearrow$	$\searrow$	indef.	$\nearrow$	$\nearrow$	= (62.61)	$\searrow$	indef.
IPPASS	$\searrow$	= (16.56)	$\searrow$	= (74.93)	$\nearrow$	$\searrow$	$\searrow$	indef.
POQSS	$\searrow$	$\searrow$	$\searrow$	$\nearrow$	$\nearrow$	$\nearrow$	$\searrow$	indef.
EOQSS	$\searrow$	$\searrow$	$\searrow$	$\nearrow$	$\nearrow$	$\nearrow$	$\nearrow$	$\nearrow$
SMSS	$\searrow$	$\nearrow$	indef.	$\searrow$	$\nearrow$	$\searrow$	$\searrow$	indef.
LUCSS	$\nearrow$	$\nearrow$	$\nearrow$	= (13.11)	$\nearrow$	$\nearrow$	$\nearrow$	$\nearrow$

Table 10: Assessing the impact of TBO increases on relative cost performances.

The choice of a particular demand pattern also has an influence on the relative performance of the techniques. Relative performance is clearly better under the assumption of a normal demand for all techniques except WWSS. With an average of 50 and a standard deviation of 28.86, uniform draws of demand take more often extreme values (farer from the average) than normal draws. WW puts up better with this kind of demand shape than other heuristics that definitely behave better when extreme values of demand are sporadic.

Switching from low to high levels of lead time is detrimental to relative performance of all methods except IPPASS for which there is no significant impact (the ANOVA test on cost ratios gives $pc = 12.15\%$). Higher lead times imply less flexibility to respond to production plan changes stemming from a disclosure of new demands for end items. With higher lead times, larger safety stocks are required, the repair procedure is applied more often therefore entailing higher total costs.

When the forecast window is increased, total costs produced by all techniques globally decrease but not at the same pace. Figure 3 plots the average cost deviations relative to WWRP over forecast window values W. Except LUCRP whose performance is always quite poor, relative performance of all RP decreases as W is augmented, be the level of lead times low or high (see both graphics on top of Figure 3). For short forecast windows ($W = \{6, 7, 8\}$ in the low lead times case and $W = \{11, 12, 13\}$ in the high case), IPPARP, SMRP and POQRP outperform WWRP. For longer windows however ($W \geq 9$ in the first case and $W \geq 14$ in the second case) WWRP becomes by far the best method. This is consistent with prior studies claiming that more information is better for WW. The same tendency is observed for SS-based methods: there exists a threshold of W below which IPPASS and

SMSS (but not EOQSS) behave better than WWRP (see both graphics on bottom of Figure 3). As soon as $W \geq 9$ (low lead times) and $W \geq 14$ (high lead times), WWRP again exhibits the best performance. It should ne noted that relative performance of WWSS is enhanced with longer windows. This definitely confirms that WW is positively receptive to an extension of information.

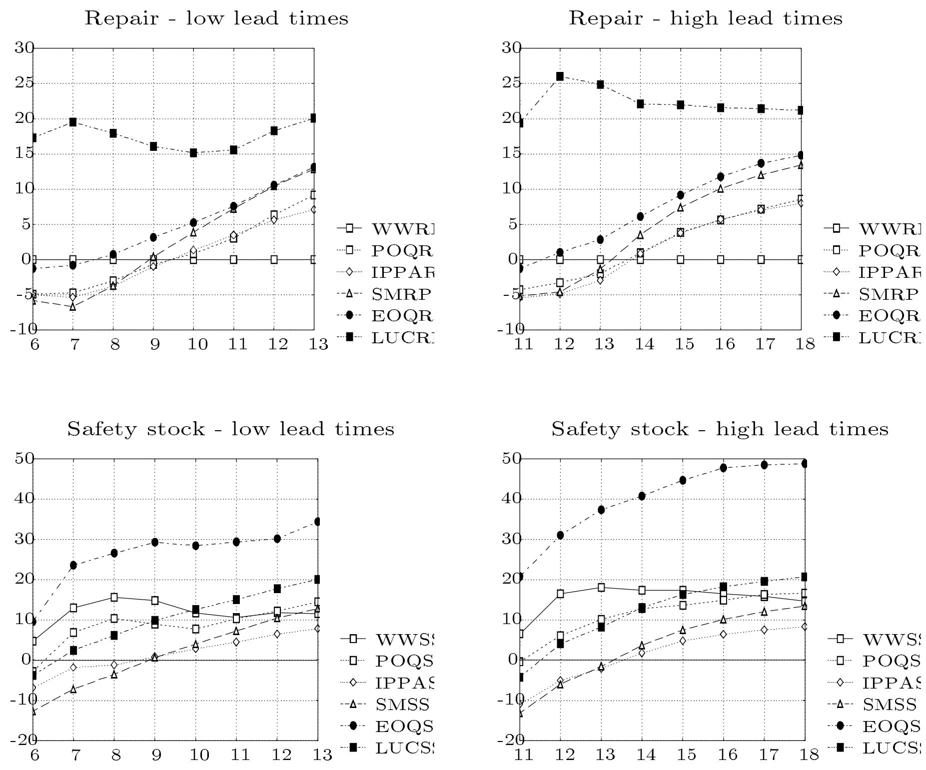


Figure 3: Average cost ratios in percent over forecast window values W.

6 Conclusion

Many past studies have assumed zero lead times and claimed this assumption could be done ‘without loss of generality’ under multi-level rolling horizons. By means of a simple example, this paper shows this assumption is definitely not innocent as the introduction of positive lead times may induce stockout situations even when demand for end items is deterministic within the forecast window. We thus propose a repair procedure to cope with such stockouts and we compare its performance to a strategy that consists in introducing safety stocks at all levels in the product structure. This safety stock strategy is borrowed from the literature examining multi-level lot-sizing problems under rolling conditions when demand is subject to forecast errors. Simulation results show the superiority of our procedure over the safety stock policy in most cases. The Wagner-Whitin algorithm provides the best overall performance when combined to our repair procedure. For short forecast windows however would we recommend the use of less sophisticated heuristics like SMRP, IPPARP or POQRP as they clearly outperform WWRP. On the other hand, when more is known about future demand, myopic behaviour of these simpler techniques is detrimental to their performance, whereas cost effectiveness of WW ostentatiously increases.

Safety stocks-based techniques behave pretty well under short windows but were specifically designed to provide a 100% service level for the sake of comparison with the repair procedure. Appropriate amounts of safety inventories were computed *ex-post* whereas in practice they have to be decided in advance. Thus, to avoid any lost demand in a real setting, an operational safety stock policy would lead to impute *ex-ante* extremely large values to safety inventories. To that extent, our approach of safety policies artificially boosts its performance and thus penalizes the relative performance of the repair procedure. In spite of all, our incentive was to compare the repair procedure to existing methods and we faced a lack of suitable approaches in this framework of analysis (multi-level, rolling schedule and positive lead times). This paper is probably the first that analyses thoroughly the impact of positive lead times in such a context, while providing an efficient procedure which guarantees a 100% service level.

In a future work, we could compare our repair procedure to more operational safety stock policies on the basis of two criteria: the service level and the cost performance. Heuristics that are specifically designed to account for interdependencies among stages could also be included in the simulation experiment and larger product structures could be examined.

References

[1] D.C. Aucamp. A variable demand lot-sizing procedure and a comparison with various well known strategies. *Production and Inventory Management*, 2:1–20, 1985.

[2] J. D. Blackburn and R. A. Millen. Improved heuristics for mutli-stage requirements planning systems. *Management Science*, 28:44–56, 1982.

[3] J.D. Blackburn and R.A. Millen. Heuristic lot-sizing performance in a rolling-schedule environment. *Decision Sciences*, 11:691–701, 1980.

[4] J. H. Bookbinder and J-Y. Tan. Two lot-sizing heuristics for the case of deterministic time-varying demands. *International Journal of Operations and Production Management*, 5:30–42, 1985.

[5] J.H. Bookbinder and B.T. H'ng. Production lot sizing for deterministic rolling schedules. *Journal of Operations Management*, 3:349–362, 1986.

[6] R.C. Carlson, S.L. Beckman, and D.H. Kropp. The effectiveness of extending the horizon in rolling production scheduling. *Decision Sciences*, 13:129–146, 1982.

[7] S. Chand. A note on dynamic lot sizing in a rolling-horizon environment. *Decision Sciences*, 13:113–119, 1982.

[8] S. Chand. Rolling horizon procedures for the facilities in series inventory model with nested schedules. *Management Science*, 29:237–249, 1983.

[9] M.A. De Bodt, L.N. Van Wassenhove, and L.F. Gelders. Lot sizing and safety stock decisions in an mrp systems with demand uncertainty. *Engineering Costs and Production Economics*, 6:67–75, 1982.

[10] G. K. Groff. A lot sizing rule for time-phased component demand. *Production and Inventory Management*, 20:47–53, 1979.

[11] Y.P. Gupta, Y.K. Keung, and M.C. Gupta. Comparative analysis of lot-sizing models for multi-stage systems: a simulation study. *International Journal of Production Research*, 30:695–716, 1992.

[12] D.H. Kropp, R.C. Carlson, and S.L. Beckman. A note on stopping rules for rolling production scheduling. *Journal of Operations Management*, 3:113–119, 1983.

[13] R.A. Lundin and T.E. Morton. Planning horizons for the dynamic lot size model: Zabel vs protective procedures and computational results. *Operations Research*, 23:711–734, 1975.

[14] R.A. Russel and T.L. Urban. Horizon extension for rolling production schedules: Length and accuracy requirements. *International Journal of Production Economics*, 29:111–122, 1993.

[15] E. A. Silver. The response of a concerned management scientist. *Production and Inventory Management*, 17:108–112, 1976.

[16] E. A. Silver and H. C. Meal. A heuristic for selecting lot size requirements for the case of a deterministic time-varying demand rate and discrete opportunities for replenishment. *Production and Inventory Management*, 14:64–74, 1973.

[17] N.C. Simpson. Multiple level production in rolling horizon assembly environments. *European Journal of Operational Research*, 114:15–28, 1999.

[18] H. Stadtler. Improved rolling schedules for the dynamic single-level lot-sizing problem. *Management Science*, 46:318–326, 2000.

[19] H. M. Wagner and T. M. Whitin. Dynamic version of the economic lot size model. *Management Science*, 5:89–96, 1958.