



# The single item uncapacitated lot-sizing problem with time-dependent batch sizes: NP-hard and polynomial cases

Ayse Akbalik, Christophe Rapine

## ► To cite this version:

Ayse Akbalik, Christophe Rapine. The single item uncapacitated lot-sizing problem with time-dependent batch sizes: NP-hard and polynomial cases. *European Journal of Operational Research*, 2013, 229 (2), pp.353-363. 10.1016/j.ejor.2013.02.052 . hal-00820635

**HAL Id: hal-00820635**

**<https://hal.science/hal-00820635>**

Submitted on 6 May 2013

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# The single item uncapacitated lot-sizing problem with time-dependent batch sizes : NP-hard and polynomial cases

A. Akbalik, C. Rapine

Université de Lorraine, Laboratoire LGIPM, Ile du Saulcy, Metz, F-57045, France

---

## Abstract

This paper considers the uncapacitated lot sizing problem with batch delivery, focusing on the general case of time-dependent batch sizes. We study the complexity of the problem, depending on the other cost parameters, namely the setup cost, the fixed cost per batch, the unit procurement cost and the unit holding cost. We establish that if any one of the cost parameters is allowed to be time-dependent, the problem is NP-hard. On the contrary, if all the cost parameters are stationary, and assuming no unit holding cost, we show that the problem is polynomially solvable in time  $O(T^3)$ , where  $T$  denotes the number of periods of the horizon. We also show that, in the case of divisible batch sizes, the problem with time varying setup costs, a stationary fixed cost per batch and no unit procurement nor holding cost can be solved in time  $O(T^3 \log T)$ .

*Keywords:* Inventory, uncapacitated lot sizing, batch delivery, stepwise cost, polynomial time algorithm, complexity.

---

## 1. Introduction

We consider in this article the single item uncapacitated lot-sizing problem where the quantities ordered are delivered by batch (typically truck or container) from an external supplier. The same problem can be seen as a single machine with an unlimited capacity, producing a single item per batch of a certain size. In the remaining of the paper this problem is called UBLS for uncapacitated batch lot sizing. Like in the classical lot sizing problem, ordering a positive amount incurs a fixed cost  $f_t$ , also called *setup* cost, for ordering in period  $t$ , plus a procurement cost  $p_t$  per unit ordered, and a holding cost  $h_t$  per unit of product in stock between period  $t$  and  $t + 1$ . In this paper, we adopt an FTL (Full Truck Load) cost structure, where, in addition to the classical cost structure above, a fixed cost  $k_t$  is paid for each batch used in period  $t$ . Observe that this model differs from other batch models where a solution is constrained to use only full batches. We do allow in our model incomplete batches, also called *fractional* batches, but the same fixed cost  $k_t$  is paid in period  $t$  whatever the number of units actually in a batch. We denote by  $B_t$  the size of the batch in period  $t$ . Demands

are known over a planning horizon of  $T$  periods and are to be satisfied without backlogging. Note that the demands and the batch sizes are both considered integer in the whole paper. The overall procurement cost  $q_t(x)$  for an amount  $x$  of products ordered in period  $t$  is thus given by :

$$q(0) = 0 \text{ and } q_t(x) = f_t + p_t x + \lceil x/B_t \rceil k_t \text{ for } x > 0$$

A formulation of the problem is given below, where  $x_t$  represents the amount of products ordered in period  $t$  and  $s_t$  the stock level at the end of the period  $t$ . Without loss of generality we assume no initial inventory, that is  $s_0 = 0$ .

$$(UBLS) \begin{cases} \min & \sum_{t=1}^T (q_t(x_t) + h_t s_t) \\ s.t. & s_{t-1} + x_t = d_t + s_t \quad \forall t = 1, \dots, T \\ & x_t, s_t \in \mathbb{R}_+ \quad \forall t = 1, \dots, T \end{cases}$$

The first constraint is the classical flow conservation. Notice that this formulation is non-linear due to the procurement cost  $q_t(x)$ .

The contribution of this article is to classify the complexity of the UBLS problem with time-dependent batch sizes according to the different cost parameters, and to better identify the frontier between polynomially solvable and NP-hard problems. To the best of our

---

*Email addresses:* ayse.akbalik@univ-lorraine.fr (A. Akbalik), christophe.rapine@univ-lorraine.fr (C. Rapine)

knowledge, while the case of a stationary batch size has been studied in the literature (see Section 2), the case of time-dependent batch sizes is mainly unexplored. Our results show that this problem is NP-hard if none but one of the cost parameters (setup, fixed cost per batch, unit procurement or unit holding costs) is allowed to be time-dependent. Moreover, the NP-hardness results hold under very stringent conditions, that is even if almost all the other cost parameters are null. On the opposite, when all cost parameters are stationary, we establish that the UBLSP problem with time-dependent batch sizes can be solved in  $O(T^3)$  time complexity, assuming no holding cost. Figure 1 gives a synthetic representation of our results and the corresponding sections where they appear.

For ease of the reading, in the remaining of the paper the notation  $(f_i/k_i/p_i/h_i)$  is used to designate the assumptions adopted on the cost parameters (setup cost/fixed cost per batch/unit procurement cost/unit holding cost). The field for a parameter  $\alpha$  will take either the value '-' if  $\alpha$  is null, ' $\alpha$ ' if it is assumed stationary and ' $\alpha_t$ ' if it is allowed to be time-dependent. For example, the rightmost problem in Figure 1 is designated by  $(f_i/k_i/-/-)$  in our notation, corresponding to UBLSP instances with time-dependent setup costs, stationary fixed cost per batch and no unit procurement nor holding cost.

This paper is organized as follows. We start by presenting in Section 2 some relevant papers studying lot sizing problems, unbounded or capacitated, with batch deliveries. All these papers restrict to a stationary batch size. Section 3 gives some insights of the difficulty of the UBLSP problem with time-dependent batch sizes and time-dependent cost parameters. The NP-hardness of the problem is formally proven in the succeeding sections considering in its turn each cost parameter as the only time-varying parameter together with the batch sizes. More precisely, Section 4 is devoted to the subproblem  $(-/k_i/-/-)$ , Section 5 considers both  $(-/k_i/p_i/-)$  and  $(-/k_i/-/h_i)$ , that is time-varying unit costs. In Section 6 we study the case  $(f_i/k_i/-/-)$  with time-dependent setup costs, and propose an  $O(T \sum_t B_t)$  pseudo-polynomial time algorithm. This section establishes also that this special case becomes polynomially solvable if the batch sizes are divisible (*i.e.* for any periods  $u$  and  $v$ , either  $B_u|B_v$  or  $B_v|B_u$  holds). Finally, in Section 7, we consider the case  $(f/k_i/-/-)$  of stationary cost parameters and null holding cost, and we propose an  $O(T^3)$  time algorithm. We then discuss some open problems and some possible extensions of this work in Section 8.

## 2. Literature review

The UBLSP problem is well-studied in the literature, but as far as we know all the papers restrict to the case of a stationary batch size. Depending on the authors, the procurement cost considered in this paper is also called a stepwise cost or multiple setup cost. To the best of our knowledge, the first study considering a stepwise cost (but without setup cost) is due to Lippman (1969). The author proposes an  $O(T^5)$  algorithm for the single-item UBLSP problem for the case  $(-/k_i/p_i/h_i)$  without backlogging. For the same problem, Pochet and Wolsey (1993) improve the result of Lippman by proposing an  $O(T^2 \min(T, B))$  time algorithm. Lee (1989) studies a similar structure, with a non null setup cost but assuming stationary cost parameters: in our notation  $(f/k_i/p/h)$ . The author proposes an  $O(T^4)$  time algorithm. Li *et al.* (2004) consider the more general problem both with time-dependent cost parameters  $(f_i/k_i/p_i/h_i)$  and with backlogging, and propose an  $O(T^3)$  time algorithm. Table 1 gives an overview of these different results. Recall that all of them assume a stationary batch size.

The capacitated lot-sizing problem (CLSP) with batch delivery has also been studied in the literature. Recently Van Vyve (2007) proposes an  $O(T^3)$  algorithm for a general case with time-dependent costs,  $(-/k_i/p_i/h_i)$ , time-dependent production capacity and allowing backlog. The time complexity of the algorithm reduces to  $O(T^2 \log(T))$  in the case without backlogging. However, the author assumes null setup cost and a stationary batch size. Another paper on the CLSP with batch delivery is due to Akbalik and Rapine (2012). The authors study the constant capacitated CLSP with stationary batch sizes and propose polynomial time algorithms for two cases: an  $O(T^4)$  algorithm for the case with production capacity being a multiple of the batch size and an  $O(T^6)$  algorithm for the case of an arbitrary fixed capacity.

As a conclusion, when restricted to stationary batch sizes, stepwise costs appear not to alter the complexity status of the lot-sizing problem. That is, both the uncapacitated and capacitated lot-sizing problems remain polynomial, even when backlog is allowed. In contrast, we establish in this paper that problem UBLSP with time-dependent batch sizes is NP-hard, as soon as all but one cost parameters are stationary.

## 3. Preliminaries

To our knowledge, no dominance property has been proposed in the literature for the most general ver-

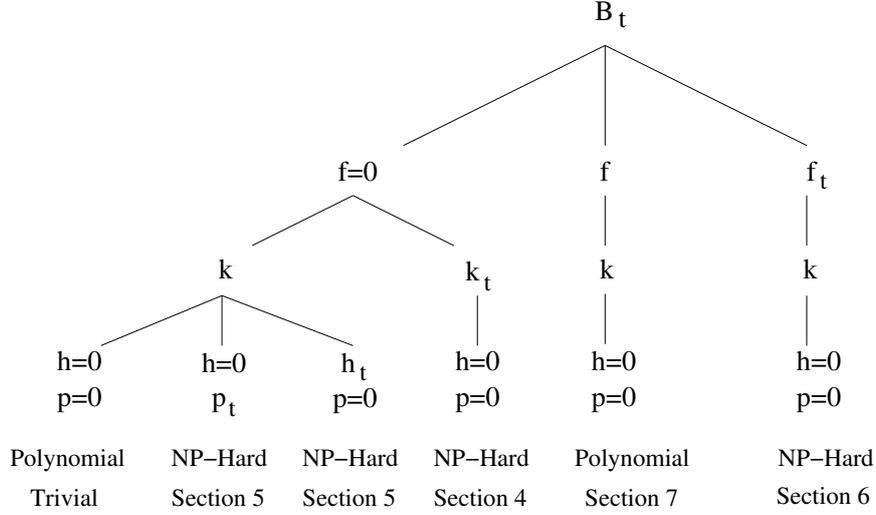


Figure 1: Complexity classification of different extensions of UBLS proposed in this paper.

Table 1: Classification of the studies on the UBLS with stationary batch size  $B$ . (\*Additional backlog assumption)

Articles	Cost structure	Complexity
Lippman (1969)	$(-/k_i/p_i/h_i)$	$O(T^5)$
Pochet and Wolsey (1993)	$(-/k_i/p_i/h_i)$	$O(T^2 \min(B, T))$
Lee (1989)	$(f/k/p/h)$	$O(T^4)$
Li <i>et al.</i> (2004)	$(f_i/k_i/p_i/h_i^*)$	$O(T^3 \log T)$

sion of UBLS with the cost parameters being all time-dependent, in our notations  $(f_i/k_i/p_i/h_i)$ . Among the classical dominance properties in lot-sizing, the ZIO property is certainly the most common, see the seminal paper from Wagner and Whitin (1958). Recall that a policy is ZIO (*Zero Inventory ordering*) if it orders only if its current stock level is null. We show with a simple illustrative example (see Figure 2) that the ZIO property is not dominant with time-dependent batch sizes. The example also demonstrates that we can not restrict to plannings where the entering stock level of an ordering period is bounded by its batch size. That is, the quantity stored at the beginning of a period where an order takes place may exceed the current batch size, and in fact any batch size, in any optimal planning. The instance consists of 2 periods with no demand at the first period and a demand of 12 units at the second period. The batch sizes are respectively 5 and 2, the fixed costs per batch are respectively 3 and 2, the other costs are null. The only optimal planning consists in producing two full batches in the first period (which represents 10

units) and one full batch in the second period (which represents 2 units), leading to a total cost of 8. Notice that the optimal policy orders in the second period while we have 10 units on hand, which is larger than the maximal batch size of the instance. We can easily modify the example by introducing unit procurement costs and decreasing the demand by one unit, such that the optimal policy orders a fractional batch in the second period.

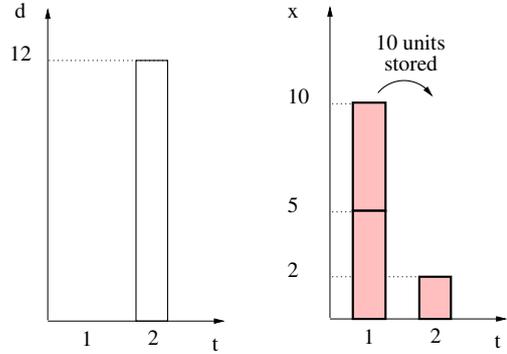


Figure 2: ZIO policies are not dominant for UBLS.

Observe that the UBLS problem allows to have a null batch size in some periods. It means that ordering in these periods is simply not possible. We can remove these periods to obtain an equivalent instance where all the batch sizes are positive. Indeed, since backlogging is not allowed, the demand of a period  $t$  with a null batch size is necessarily ordered in a previous period. Proceeding backward from the end of the time horizon, we

can delete each period with a null batch size from the instance and shift its demand to the preceding period. This procedure is a simple scan of the periods and can be achieved in linear time. This is summarized in the following remark:

**Remark 1.** *Any instance of the UBLS problem can be transformed in linear time  $O(T)$  into an equivalent instance with only positive batch sizes.*

In the literature, some papers propose heuristics and exact approaches for more general lot-sizing problems, assuming for instance piecewise concave procurement costs. In particular, the pseudo-polynomial time dynamic programming algorithm proposed by Shaw and Wagelmans (1998) for piecewise linear procurement costs and general holding costs can be adapted to solve the UBLS problem. Their algorithm solves the capacitated version of the problem in time  $O(DQ)$ , with  $D = \sum_t d_t$  the overall demand and  $Q$  the overall number of pieces required to represent the production cost functions. To apply the algorithm of Shaw and Wagelmans (1998) to the UBLS problem, we can consider a virtual capacity of  $D$  at each period. In a period  $t$ , we have as many affine pieces as the possible number of batches, that is  $O(D/\min\{B_t, D\})$ . Denoting  $\bar{B}_t = \min\{B_t, D\}$ , the time complexity of the problem is then in  $O(\sum_t D^2/\bar{B}_t)$ . Since we can restrict ourselves to the instances with only positive batch sizes (cf Remark 1), this complexity is included in  $O(TD^2)$ .

**Theorem 1.** *Problem UBLS can be solved in pseudo-polynomial time  $O(D^2 \sum_t \frac{1}{\bar{B}_t})$ .*

Though such a pseudo-polynomial algorithm is of little practical use for large demands and small batch sizes, it is important in theory since it implies that the UBLS problem can not be NP-hard in the strong sense. As a consequence, the complexity results of the following sections establish the NP-hardness of UBLS in the ordinary sense.

In the following, we are interested in showing that the UBLS problem is computationally difficult to solve even on very restricted classes of instances. In particular, we often restrict to a stationary unit procurement cost  $p$ . It is well known that in this case we can assume without loss of generality that  $p = 0$ , as long as the final inventory level is null. Also, we often consider a null unit holding cost. Observe that, contrary to the case of unit procurement cost, an arbitrary stationary holding cost  $h$  can not be reduced to the case of a null holding cost. Quite inevitably the special case with  $p = 0$  and  $h = 0$  will be encountered in what follows. Observe that when

$p = 0$  and  $h = 0$ , the costs incurred by a batch do not depend on the number of units in it. Hence we can make the following remark:

**Remark 2.** *With no unit procurement cost nor holding cost ( $p = h = 0$ ), it is dominant to produce only full batches.*

One can legitimately wonder for the instances with a stationary unit procurement cost  $p$  and a null holding cost if we can both assume that  $p = 0$  and restrict to full batch plannings. Obviously we have little chance to end with a null inventory. However, it is easy to see that the optimal planning obtained with no unit procurement cost can be transformed into an optimal planning for a stationary unit procurement cost  $p$ , simply by removing the last batch ordered so that the planning satisfies exactly the total demand.

Finally, we never consider in this article the case of a null fixed cost per batch ( $k = 0$ ). Clearly if  $k = 0$  we do not need to pay attention to the size of the batches. The UBLS problem is then a classical uncapacitated lot sizing problem, and can be solved in linear time, see Aggarwal and Park (1993), even if all the other cost parameters are time-dependent.

#### 4. Time-dependent fixed costs per batch

We consider a first restricted version of the UBLS: The only time-dependent parameters are the fixed cost per batch ( $k_t$ ) and the batch sizes ( $B_t$ ). The next theorem states that the problem UBLS is NP-hard when both fixed cost per batch and batch sizes are time-dependent, even with no setup cost, no unit procurement cost and no unit holding cost.

**Theorem 2 (NP-hardness of  $(-|k_t| - | - )$ ).** *Problem UBLS is NP-hard even with null setup costs and null unit procurement and holding costs.*

**Proof.** The reduction to our restricted UBLS problem from the UNBOUNDED KNAPSACK PROBLEM (UKP) is quite immediate. The UKP has been proven NP-complete by Lueker (1975). Recall that an instance of the UNBOUNDED KNAPSACK PROBLEM is constituted of  $n$  objects, each object  $i$  being associated with a profit  $u_i$  and a weight  $w_i$ . Given a knapsack of size  $W$  and a value  $U$ , it is asked whether there exists an integer vector  $y = (y_1, \dots, y_n)$  satisfying  $\sum_i w_i y_i \leq W$  and  $\sum_i u_i y_i \geq U$ . Observe that contrary to the 0-1-KNAPSACK PROBLEM, we can select multiple copies of each object in the solution. We transform an instance  $I$  of UKP into an instance  $f(I)$  of the UBLS problem as follows :

- We have  $T = n + 1$  periods. The only positive demand appears in the last period, with  $d_T = U$
- The size of a batch in a period  $i \in [1, n]$  is  $B_i = u_i$  and its fixed cost is  $k_i = w_i$ . The batch size in the last period is null.
- Setup costs, unit procurement and holding costs are null.
- It is asked if a solution of cost at most  $W$  exists.

It is dominant to produce only full batches, and thus a planning is entirely described by the number  $y_i$  of batches ordered in each period  $i$ . Clearly  $f(I)$  is positive if and only if there exists  $y \in \mathbb{Z}_+^n$  such that  $\sum_i B_i y_i \geq d_T$  and  $\sum_i k_i y_i \leq W$ , that is, if and only if the instance  $I$  is positive.  $\square$

Theorem 2 still holds even for stationary demands, that is  $d_i = d$  for all periods. For this, one can modify the reduction of the proof, adding a first period 0 with fixed cost  $k_0 = W + 1$  and batch size  $B_0 = nU$ . The demand to satisfy in each period is now equal to  $U$ . It is asked if a solution of cost at most  $2W + 1$  exists. Clearly, any feasible solution must order in period 0. Thus a solution of cost at most  $2W + 1$  orders exactly one (full) batch at the first period, leaving only the demand  $d_T = U$  to satisfy. Hence we are exactly in the situation of the reduction of Theorem 2, that is we are asked if we can satisfy the last demand  $d_T$  at cost at most  $W$  from the periods 1 to  $n$ . We assume in the rest of this paper that the fixed costs per batch are stationary.

## 5. Time-dependent unit procurement or holding costs

We consider next the case of a time-dependent unit procurement cost  $p_i$ . We show that even if all other cost parameters are stationary, problem UBLS with time-dependent batch sizes is NP-hard :

**Theorem 3 (NP-hardness of  $(-|k|/p_i/-)$ ).** *Problem UBLS is NP-hard even with stationary fixed costs per batch, no setup cost and no holding costs.*

**Proof.** The proof is very similar to the proof proposed by Florian and Klein (1971) for the capacitated lot-sizing problem (CLSP), except that the reduction is done from a restricted version of the UNBOUNDED KNAPSACK PROBLEM called MONEY CHANGE PROBLEM (MCP), see Böcker and Lipták (2007). In the MONEY CHANGE PROBLEM, we are given  $n + 1$  integers  $a_1, \dots, a_n$  and  $A$ . It is asked whether there exists an integer vector  $y \in \mathbb{Z}_+^n$

such that  $\sum_i a_i y_i = A$ . That is, is it possible to constitute exactly the amount  $A$  of money by using coins of facial values  $a_1, \dots, a_n$ ? For a proof of the NP-completeness of MCP, the reader is referred to Lueker (1975).

We transform an instance  $I$  of the MONEY CHANGE PROBLEM into an instance  $f(I)$  of UBLS problem as follows :

- We have  $T = n + 1$  periods. The only positive demand appears in the last period, with  $d_T = A$
- The size of a batch in a period  $i \in [1, n]$  is  $B_i = a_i$  and its unit procurement cost is  $p_i = \frac{a_i - 1}{a_i}$ . The batch size in the last period is null.
- The fixed cost per batch equals 1 in each period
- Setup costs and holding costs are null in all periods
- It is asked if a solution of cost at most  $A$  exists.

As in Florian and Klein (1971), the keystone of the proof is to notice that for any period  $i = 1, \dots, n$ , the overall procurement cost  $q_i(x) = \lceil x/a_i \rceil + (a_i - 1)(x/a_i)$  for an amount  $x$  satisfies  $q_i(x) \geq x$ , and,  $q_i(x) = x$  if and only if  $x$  is a multiple of  $a_i$ . That is, only full batch deliveries permit an average cost per product of 1. If the MCP instance  $I$  is positive, clearly ordering  $y_i$  full batches in each period  $i$  results in a feasible planning for the UBLS instance  $f(I)$ , of cost exactly  $A$ . Conversely, if  $f(I)$  is positive, since the total demand to serve is of  $A$  units and the average procurement cost per product is at least 1, there must exist a feasible planning ordering exactly  $A$  units in total and using only full batches. The number  $y_i$  of batches ordered in each period is thus a valid vector for the instance  $I$ .  $\square$

We have the counterpart of Theorem 3 for time-dependent holding costs:

**Theorem 4 (NP-hardness of  $(-|k| - |h_i|)$ ).** *Problem UBLS is NP-hard even with stationary fixed costs per batch, no setup costs and no unit procurement costs.*

**Proof.** We use the classical result, see Pochet and Wolsey (2006) or Wagelmans *et al.* (1992), that any instance of the lot-sizing problem can be transformed into an equivalent instance with no holding costs and modified unit procurement costs. For this, we redefine the unit procurement cost at period  $t$  as  $\tilde{p}_t = p_t + \sum_{u=1}^T h_u$ .  $\square$

Once again Theorem 3 and 4 state very restrictive conditions under which the problem UBLS is NP-hard if the fixed cost per batch is stationary and unit (procurement or holding) costs are time-dependent. In the following we assume that both unit costs are stationary.

## 6. Time-dependent setup costs

We consider here the setup costs ( $f_t$ ) as the only time-dependent cost parameters. In addition, we restrict ourselves to instances with no unit procurement cost ( $p = 0$ ) and no unit holding cost ( $h = 0$ ). This means that once the setup paid, any batch in any period has the same cost  $k$ . But of course the size of the batches may differ from one period to another. Obviously, it is dominant to produce only full batches. Although this problem seems simple, it happens to be NP-hard as stated in section 6.1. In section 6.2 we show that the problem is pseudopolynomially solvable and it becomes polynomially solvable in the case of *divisible* batch sizes (see section 6.3).

### 6.1. NP-hardness proof for UBLS ( $f_t/k/ - / -$ )

**Theorem 5 (NP-hardness of ( $f_t/k/ - / -$ )).** *Problem UBLS is NP-hard even with stationary fixed cost per batch and null unit procurement and holding costs.*

**Proof.** We show that the 0-1-KNAPSACK PROBLEM can be polynomially reduced to the problem UBLS. Recall that in the KNAPSACK PROBLEM we are given  $n$  objects, with a profit  $u_i$  and a weight  $w_i$  for each object  $i$ , together with a knapsack size  $W$  and a utility value  $U$ . It is asked whether there exists a subset  $S$  of objects whose total weight does not exceed  $W$  and whose total profit is at least  $U$ . That is, if there exists a binary vector  $y = (y_1, \dots, y_n) \in \{0, 1\}^n$  satisfying  $\sum_i w_i y_i \leq W$  and  $\sum_i u_i y_i \geq U$ .

*The transformation  $f$ .* We transform an instance  $I$  of the KNAPSACK PROBLEM into an instance  $f(I)$  of UBLS as follows: We have  $T = 2n + 1$  periods. Each couple of consecutive periods  $(2i - 1, 2i)$  is related somehow to the object  $i$ . The same idea is used in Helmrich *et al.* (2012). We call the couple  $(2i - 1, 2i)$  of periods the *pair  $i$* . The last period  $T$  is different and has a particular role. The idea of the reduction is to constrain  $f(I)$  to be positive if and only if there exists a planning which orders exactly one batch in each pair and no batch in period  $T$ . In each pair  $i$ , the choice between ordering in the even or in the odd period will correspond to the choice between selecting or not the object  $i$  in the instance  $I$ . We now define explicitly the instance  $f(I)$ :

- The demand in each period  $t$  is:

$$d_t = \begin{cases} 0 & \text{if } t \text{ is odd, } t < T \\ 3^{t/2}B & \text{if } t \text{ is even, } t < T \\ U & \text{if } t = T \end{cases}$$

where  $B = \sum_i u_i$  the total utility of the knapsack instance  $I$ .

- The batch size in each period  $t$  is:

$$B_t = \begin{cases} 3^{(t+1)/2}B & \text{if } t \text{ is odd, } t < T \\ 3^{t/2}B + u_{t/2} & \text{if } t \text{ is even, } t < T \\ U & \text{if } t = T \end{cases}$$

- The setup cost in each period  $t$  is:

$$f_t = \begin{cases} 0 & \text{if } t \text{ is odd} \\ w_{t/2} & \text{if } t \text{ is even} \end{cases}$$

- For all periods the fixed cost per batch is  $k = W + 1$
- It is asked if a solution of cost at most  $Z = nk + W$  exists.

The resulting instance  $f(I)$  looks as follows:

$t$	demand	batch size	setup cost
1	0	$3B$	0
2	$3B$	$3B + u_1$	$w_1$
3	0	$9B$	0
4	$9B$	$9B + u_2$	$w_2$
...			
$2i - 1$	0	$3^i B$	0
$2i$	$3^i B$	$3^i B + u_i$	$w_i$
...			
$2n - 1$	0	$3^n B$	0
$2n$	$3^n B$	$3^n B + u_n$	$w_n$
$T$	$U$	$U$	0

This transformation  $f$  is polynomial. The number of periods in the instance  $f(I)$  is clearly polynomial in the size of instance  $x$ ; the non-trivial point to check is that too large numbers do not appear in the transformation. The largest value  $\text{MAX}(f(I))$  appearing in the instance is either  $k = W + 1$ , or the demand  $d_{2n} = 3^n B$ . This value can clearly be encoded in space  $O(n + \log(\text{MAX}(I)))$  and thus the size of  $f(I)$  is polynomially bounded by the size of  $I$ .

*The transformation  $f$  is a reduction.* To prove that  $f$  is a reduction, we introduce 2 particular types of plannings, namely the *1-plannings* and the *extended 1-plannings*. We say that a planning is a *1-planning* if it orders exactly one batch in each pair  $i$ , either at period  $2i - 1$  or at period  $2i$ , and does not order at the last period. An *extended 1-planning* is simply a *1-planning* plus eventually an additional batch ordered at the last period  $T$ . In the following, we establish that:

- There is a natural one-to-one mapping between subsets of objects of  $I$  and *1-plannings* of  $f(I)$ , such that,

if a subset  $S$  of objects and a 1-planning  $\pi$  are in correspondence, we have the following relations between the feasibility of these solutions and their costs :

$$\begin{aligned} S \text{ has a utility } \geq U &\Leftrightarrow \pi \text{ satisfies all the demands} \\ S \text{ has a weight } \leq W &\Leftrightarrow \pi \text{ has a cost } \leq Z \end{aligned}$$

**b.** Extended 1-plannings are dominant for the instance  $f(I)$ .

Based on these 2 results, the proof that  $f$  is a reduction is quite straightforward. Assume that  $I$  is a positive instance. It implies that there exists a subset  $S$  of utility at least  $U$  and of weight at most  $W$ . Due to Equivalences 1 and 2, its associated 1-planning  $\pi$  is feasible and costs at most  $Z$ , showing that  $f(I)$  is a positive instance.

Conversely, assume that  $f(I)$  is a positive instance: There exists a feasible planning  $\pi$  of cost at most  $Z$ . Due to the dominance of extended 1-plannings, we can choose  $\pi$  to be an extended 1-planning. Now observe that an extended 1-planning ordering at the last period costs at least  $(n+1)k > Z$ . It follows that an extended 1-planning of cost at most  $Z$  is in fact a 1-planning. Since  $\pi$  is a feasible 1-planning of cost at most  $Z$ , Equivalences 1 and 2 imply that its associated subset  $S$  of objects has a weight at most  $W$  and a utility at least  $U$ , showing that  $I$  is a positive instance.

The remaining of the proof of Theorem 5 is thus devoted to prove (a) and (b): the correspondence between subsets  $S$  and 1-plannings, and the dominance of extended 1-plannings.

*Correspondence with 1-plannings.* Recall that a 1-planning is a planning ordering exactly one batch in each pair  $i$ , either at period  $2i-1$  or at period  $2i$ , and ordering no batch at the last period. We exhibit now a natural one-to-one correspondence between 1-plannings for  $f(I)$  and subset  $S$  of objects for  $I$  :

- To any 1-planning  $\pi$ , we associate the subset  $S_\pi$  defined as the objects  $i$  such that  $\pi$  orders a batch in the even period of the pair  $i$ . That is  $S_\pi = \{ i \mid \pi \text{ orders in period } 2i \}$
- To any subset  $S$ , we associate the planning  $\pi_S$  where for each pair  $i$ , a batch is ordered in the even period ( $2i$ ) if the object  $i$  belongs to  $S$ , and in the odd period ( $2i-1$ ) if  $i \notin S$  and  $i < T$ . By construction, exactly one batch is produced in any couple  $(2i-1, 2i)$  of periods, and thus  $\pi_S$  is a 1-planning.

These transformations clearly define a mapping between subsets of objects and 1-plannings. That is, for

any subset  $S$  and 1-planning  $\pi$ , we have  $\pi = \pi_S$  if and only if  $S = S_\pi$ . Consider a subset  $S$  and a planning  $\pi$  in correspondence (that is  $\pi = \pi_S$  and  $S = S_\pi$ ). We want to establish Equivalences 1 and 2 between  $S$  and  $\pi$ .

(1) To prove the first equivalence, observe that any (Q)-planning satisfies all the demands, except possibly in the last period. Thus, the planning  $\pi$  is feasible if and only if the last demand can be satisfied, which is equivalent to have a stock level  $s_{2n}$  at the end of period  $2n$  that covers  $d_T = U$ . By definition  $s_{2n}$  is equal to the total capacity of the batches produced minus the total demand on  $[1, 2n]$ , that is :

$$\begin{aligned} s_{2n} &= \left[ \sum_{i \in S} (3^i B + u_i) + \sum_{i \notin S, i \leq n} 3^i B \right] - \sum_{i=1}^{2n} d_i \\ &= \sum_{i \in S} u_i \end{aligned}$$

Hence  $\pi$  satisfies all the demands if and only if  $\sum_{i \in S} u_i \geq U$ . To prove the second equivalence, we simply write that the cost of the planning  $\pi_S$  is equal to:

$$\begin{aligned} z(\pi_S) &= \sum_{i \notin S, i \leq n} k + \sum_{i \in S} (w_i + k) \\ &= nk + \sum_{i \in S} w_i \end{aligned}$$

Hence  $\pi$  is of cost at most  $Z = nk + W$  if and only if  $\sum_{i \in S} w_i \leq W$ , which establishes Equivalence 2.

*Dominance of extended 1-plannings.* Recall that an extended 1-planning is a 1-planning plus eventually an additional batch ordered at the last period  $T$ . We establish here that these plannings are dominant for the instance  $f(I)$ . Notice that any extended 1-planning ordering at the last period is feasible. To prove the dominance of extended 1-plannings, consider an optimal planning  $\pi^*$ . Let us denote by  $a_i^*$  the number of batches ordered in the pair  $i$ ,  $i = 1, \dots, n$ , with  $a_{n+1}^*$  being by extension the number of batches ordered in the last period. For the sake of contradiction assume that no optimal planning is an extended 1-planning. Since backlogging is not allowed, any policy does not need to order more than  $U$  units, the demand of the last period, at the last period. The size of a batch in the last period being precisely  $U$ , an optimal planning clearly orders at most one batch in period  $T$ , that is,  $a_{n+1}^* \leq 1$ . It implies that for some indices  $i \leq n$ , we have  $a_i^* \neq 1$ , otherwise by definition  $\pi^*$  would be an extended 1-planning. Let  $m$  be the first index such that  $a_m^* \neq 1$ . Among all the optimal plannings, we assume that we have chosen  $\pi^*$  such that the index  $m$  is maximal. We distinguish 3 cases according to the number of batches ordered in the pair  $m$ :

1.  $a_m^* = 0$ . By construction exactly one batch is ordered in the pairs 1 to  $m-1$ . It is easy to see that the stock level at the beginning of the period  $(2m-1)$

is at most  $u_1 + \dots + u_{m-1} < B$ . As a consequence, the demand of the period  $2m$  can not be satisfied, which contradicts that  $\pi^*$  is a feasible solution.

2.  $a_m^* \geq 3$ . Consider the planning  $\pi$  identical to  $\pi^*$  except that 2 batches are suppressed in the pair  $m$  and replaced by one additional batch ordered in period  $(2m + 1)$ , that is in the odd period of the pair  $m + 1$  (or the last period  $T$  if  $m = n$ ). We claim that  $\pi$  is feasible: Since at least one batch is still ordered in  $m$ , the demands of periods  $1, \dots, m$  can clearly be satisfied. For the following periods, the two batches suppressed in the pair  $m$  represent at most  $2(3^m B + u_m)$  units, which is less than the additional batch of size  $3^{m+1} B$  added in period  $2m + 1$  if  $m < n$ . In case where  $m$  is the last pair, the batch added in period  $T$  is only of size  $U$ , but this is clearly sufficient to serve the demand  $d_T$ . If we compute the cost of this feasible planning, noticing that the setup is null in period  $(2m + 1)$ , we obtain that  $z(\pi) \leq z(\pi^*) - k$ , which contradicts that  $\pi^*$  is optimal.
3.  $a_m^* = 2$ . Consider the planning  $\pi$  identical to  $\pi^*$  except that 1 batch is suppressed in the pair  $m$  and replaced by one additional batch ordered in period  $(2m + 1)$ . As in the previous case the planning  $\pi$  is feasible, and costs at most the cost of  $\pi^*$  since we have a null setup cost in period  $(2m + 1)$ . It follows that  $\pi$  is also optimal, and orders exactly one batch in periods 1 to  $m$  included. This contradicts our choice of  $\pi^*$  such that its index  $m$  is maximal.

In any case we obtain a contradiction. It results that extended 1-plannings are dominant for the instance  $f(I)$ .  $\square$

## 6.2. A pseudo-polynomial time algorithm for UBLS $(f_i/k/ - / -)$

We show in this section that the problem UBLS with a stationary fixed cost per batch  $k$  and no unit procurement nor holding cost  $((f_i/k/ - / -)$ , see Theorem 5) can be solved in pseudo-polynomial time more efficiently than with the general dynamic algorithm of Section 3. However, the main purpose of this section is to exhibit some simple dominance properties which will be useful to derive polynomial time algorithms for special cases, namely the divisible batch sizes case and the stationary setup cost case.

We restrict our attention to policies ordering only full batches, see Remark 2. Without loss of generality we can also assume that there is no null batch size due to Remark 1. Notice that if the setup costs are stationary,

we can also consider without loss of generality that the batch sizes are increasing over the periods, that is  $u < v \Rightarrow B_u < B_v$ . Indeed if  $B_u$  happens to be greater than  $B_v$ , there exists an optimal solution that does not order in period  $v$ . Thus, again we can discard the period  $v$  and shift its demand to the previous period. The next proposition states that it is still true for time-dependent setup costs, in the sense that it is dominant for the batch sizes of the ordering periods of a policy to be increasing. In addition, it is also dominant that the stock level at the beginning of an ordering period be lower than the batch size of the preceding ordering period.

**Property 1.** *For problem UBLS  $(f_i/k/ - / -)$ , there exists an optimal policy such that, for any two consecutive ordering periods  $u$  and  $v$ , with  $u < v$ , the inequalities  $s_{v-1} < B_u < B_v$  hold.*

**Proof.** The proof is immediate by an interchange argument. Consider any optimal policy  $\pi$  and let  $u$  and  $v$  be two consecutive ordering periods. Let  $m_t$  be the number of batches ordered in period  $t$  by  $\pi$ . Consider first that  $B_u \geq B_v$ . We can modify  $\pi$  by ordering  $(m_u + m_v)$  batches in period  $u$  and nothing in period  $v$ . The resulting planning is clearly still feasible and optimal. Repeating this transformation we obtain an optimal policy where the batch sizes of the ordering periods are increasing. Now consider that in this optimal policy we have  $s_{v-1} \geq B_u$ . Write  $s_{v-1} = aB_u + b$  with  $b < B_u$ . We can again modify the policy by cancelling  $a$  batches in period  $u$  and ordering  $a$  additional batches in period  $v$ . Since the batch sizes are increasing, the resulting policy is still feasible and also optimal. Repeating this transformation leads to an optimal planning verifying the inequalities of Property 1.  $\square$

Property 1 implies that it is dominant to order at a period the minimum number of batches needed to satisfy the demands till the next ordering period. More precisely consider an optimal policy  $\pi$ , and  $u$  and  $v$  two consecutive ordering periods. We denote by  $D_{u,v-1} \equiv d_u + \dots + d_{v-1}$  the sum of the demands between the period  $u$  and the period  $v$ ,  $v$  excluded. We introduce  $b_{uv}$  as the rest in the Euclidean division of this demand by the batch size  $B_u$ , that is :

$$b_{uv} = D_{u,v-1} \bmod B_u \quad (3)$$

Let  $s_{u-1}$  be the stock level at the beginning of period  $u$ . Since it is dominant to have a stock level at the beginning of period  $v$  lower than  $B_u$ , it results that it is dominant to order in period  $u$  a number of batches equal to  $\lceil (D_{u,v-1} - s_{u-1}) / B_u \rceil$ . Moreover, Property 1 also implies

that  $s_{u-1} < B_u$ . It results that the number  $m_{uv}$  ordered in a dominant planning, depending on the stock level  $s$  at the beginning of period  $u$ , is equal to :

$$m_{uv}(s) = \begin{cases} \lfloor D_{u,v-1}/B_u \rfloor + 1 & \text{if } s < b_{uv} \\ \lfloor D_{u,v-1}/B_u \rfloor & \text{if } s \geq b_{uv} \end{cases} \quad (4)$$

We simply use the notation  $m_{uv}$  in the following if the stock level  $s_{u-1}$  is clear from the context. Notice that  $m_{uv}$  corresponds to the minimal number of full batches needed to satisfy the demands till the beginning of period  $v$ . In a dominant planning we also have a simple relation between the entering stock level of two consecutive ordering periods  $u$  and  $v$ . Indeed using the fact that  $s_{v-1} \in \{0, 1, \dots, B_u - 1\}$  due to Property 1, we can write modulo  $B_u$  the flow conservation equation  $s_{v-1} = s_{u-1} + m_{uv}B_u - D_{u,v-1}$  to obtain:

$$s_{v-1} = (s_{u-1} - b_{uv}) \bmod B_u \quad (5)$$

For short we denote by  $\sigma_{uv}$  the function that associates to a value  $s$  the quantity  $\sigma_{uv}(s) = (s - b_{uv}) \bmod B_u$ . A representation of  $\sigma_{uv}$  is given in Figure 3. Notice that this function is not monotone with  $s$ .

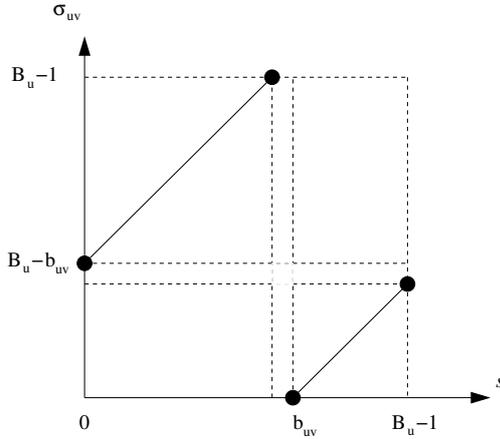


Figure 3: The entering stock level  $\sigma_{uv}(s)$  at  $v$  in terms of the entering stock level at  $u$ .

We summarize our results and notations in Property 2 below, which is a corollary of Property 1:

**Property 2.** For problem UBLS  $(f_i/k| - /-)$ , there exists an optimal policy such that the number  $m_t$  of batches ordered at period  $t$  and the stock level  $s_t$  at the end of period  $t$  verify, for any two consecutive ordering periods  $u$  and  $v$ :

$$s_{v-1} = \sigma_{uv}(s_{u-1}) \text{ and } m_u = m_{uv}(s_{u-1})$$

Property 2 means that, given a set of ordering periods, an optimal policy orders each time the minimal number of (full) batches to avoid a backlog. It also implies that if one can guess the ordering periods of an optimal policy, the quantities to order can be determined in linear time. Based on these dominance properties, it is easy to derive a dynamic programming algorithm. Let  $\text{OPT}(u, s)$  be the minimal cost to satisfy all the demands over  $[u, T]$  given an entering stock level of  $s$  at period  $u$ , and assuming that  $u$  is an ordering period. By convention we set  $\text{OPT}(T + 1, s) = 0$  for any stock level  $s$ . From what precedes, we have:

$$\text{OPT}(u, s) = f_u + \min_{v > u} \{ m_{uv}(s)k + \text{OPT}(v, \sigma_{uv}(s)) \} \quad (6)$$

The optimal cost is given by  $\min_u \{ \text{OPT}(u, 0) \mid d_1 + \dots + d_{u-1} = 0 \}$ . We now analyze the time complexity of this dynamic programming approach. Consider a period  $u$  and a stock level  $s$ . Observe that all the  $D_{uv}$  values for  $u \leq v \leq T$  can be computed in linear time by accumulation, and thus all the quantities  $m_{uv}(s)$  and  $b_{uv}$  can also be determined in time  $O(T)$ . It results from Equation 6 that each value  $\text{OPT}(u, s)$  can be determined in time  $O(T)$ , that is each state can be evaluated in linear time. Due to Property 1, it is sufficient for each period  $u$  to consider the values of  $s$  in the set  $\{0, \dots, B_u - 1\}$ . Thus the number of states in the formulation is equal to  $\sum_t B_t$ . We have the following result :

**Theorem 6.** Problem UBLS  $(f_i/k| - /-)$  with stationary fixed cost per batch and null unit procurement and holding costs can be solved in pseudo-polynomial time  $O(T \sum_t B_t)$ .

Hence if the largest batch size  $B_{\max}$  appearing in the instance is reasonably small, we have an efficient algorithm running in time  $O(T^2 B_{\max})$ . We show in the next section how this pseudo-polynomial time algorithm can be transformed into a polynomial one in the case of divisible batch sizes.

### 6.3. A polynomial time algorithm for UBLS $(f_i/k| - /-)$ with divisible batches

We consider in this section the special case of divisible batch sizes: For any periods  $u$  and  $v$ , either  $B_u | B_v$  or  $B_v | B_u$  holds. To derive a polynomial time algorithm from the dynamic programming of the previous section, the basic idea is to reduce the number of states to consider in the recursive equation (6). That is, we want to show that among the  $B_u$  possible stock levels at the beginning of a period  $u$ , only a small subset (of cardinality polynomially bounded in  $T$ ) is of interest for the optimal policy. Observe that Equation (4) implies that, to decide

how many batches to order in a period  $u$ , given that  $v$  is the next ordering period, we do not need to know the precise value of the stock level  $s_{u-1}$  but only if  $s_{u-1}$  is lower than  $b_{uv}$  or not. Going along this line, for a given period  $u$ , consider all the different values  $b_{uv}$  for  $v > u$ . These values partition the set  $[0, B_u - 1]$  into at most  $T$  intervals. More precisely, assuming that the  $b_{uv}$ 's take  $l$  distinct positive values, let  $0 < \alpha_1^u < \dots < \alpha_l^u < B_u$  be these values indexed in increasing order. For convenience we introduce  $\alpha_0^u = 0$  and  $\alpha_{l+1}^u = B_u$ . We call for short the interval  $[\alpha_i^u, \alpha_{i+1}^u)$  the  $i$ th  $\alpha^u$ -interval.

Notice that Equation (4) implies that the number of batches ordered in a dominant planning is the same for any stock level of a given  $\alpha^u$ -interval. Indeed, the number  $m_{uv}$  of batches only depends on the relative position of  $s_{u-1}$  and  $b_{uv}$ . If  $s_{u-1}$  belongs to the interval  $[\alpha_i^u, \alpha_{i+1}^u)$ , clearly we have the equivalence  $s_{u-1} \geq b_{uv}$  if and only if  $\alpha_i^u \geq b_{uv}$ . Since all the stock levels in an  $\alpha^u$ -interval lead to the same optimal ordering decision, we can represent the  $B_u$  states  $(u, s)$  of the dynamic programming in a compact way, namely  $\{(u, 0), (u, 1), \dots, (u, l+1)\}$ , where the new state  $(u, i)$  means that the entering stock level at period  $u$  belongs to the  $i$ th  $\alpha^u$ -interval. This compact description reduces the number of states from  $\sum_i B_i$  to at most  $T^2$ .

The matter is that in general we can not deduce from the fact that  $s_{u-1} \in [\alpha_i^u, \alpha_{i+1}^u)$  in which interval  $[\alpha_j^v, \alpha_{j+1}^v)$  the next entering stock level  $s_{v-1} = \sigma_{uv}(s_{u-1})$  belongs. In fact, the next entering stock level  $s_{v-1}$  does depend on the current stock level  $s_{u-1}$  and not only on its  $\alpha^u$ -interval. That is, contrary to the number of batches to order, depending on the actual value of  $s_{u-1}$  inside its  $\alpha^u$ -interval, the resulting stock level  $\sigma_{uv}(s_{u-1})$  may belong to different  $\alpha^v$ -intervals. To illustrate this fact, consider the case where  $s_{u-1} \geq b_{uv}$ . The entering stock level at period  $v$  is  $\sigma_{uv}(s_{u-1}) = s_{u-1} - b_{uv}$ . It results that for a given period  $w > v$ , we have  $\sigma_{uv}(s_{u-1}) \geq b_{vw}$  if and only if  $s_{uv} \geq b_{uv} + b_{vw}$ . Though this latter condition is simple, in general the quantity  $b_{uv} + b_{vw}$  does not correspond to any of the values  $\alpha_i^v$ , and thus we have no clue to determine the position of  $\sigma_{uv}(s_{u-1})$  relatively to the  $\alpha_i^v$ 's. The rest of this section is devoted to show that in the case of divisible batch sizes, all the entering stock levels  $s_{u-1}$  in the same  $\alpha^u$ -interval lead to stock levels  $s_{v-1}$  all in the same  $\alpha^v$ -interval (see Figure 4 and Property 3 below), and that the index of this interval can be computed quite efficiently (see the end of this section). Observe that to determine in which  $\alpha^v$ -interval belongs  $s_{v-1}$ , it is by definition sufficient to be able to decide for any period  $w > v$  if  $s_{v-1} \geq b_{vw}$ . This is precisely the purpose of Property 3:

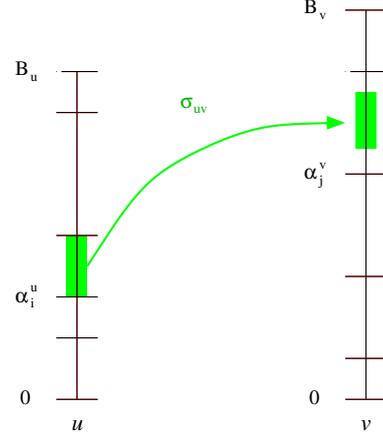


Figure 4: An example of the image of an  $\alpha^u$ -interval by the application of  $\sigma_{uv}$ . All the resulting stock levels  $\sigma_{uv}(s)$  belong to the same  $\alpha^v$ -interval.

**Property 3.** *If the batch sizes are divisible, for any periods  $u < v < w$ , we have in a dominant planning:*

- For any stock level  $s \geq b_{uv}$ ,  $\sigma_{uv}(s) \geq b_{vw}$  if and only if  $(s \geq b_{uv})$  and  $(b_{uv} + b_{vw} < B_u)$*
- For any stock level  $s < b_{uv}$ ,  $\sigma_{uv}(s) \geq b_{vw}$  if and only if  $(s \geq b_{uv})$  and  $(b_{uv} + b_{vw} < 2B_u)$  or  $(b_{uv} + b_{vw} < B_u)$*

**Proof.** We first establish that for any periods  $u < v < w$ , we have:

$$B_u | B_v \Rightarrow (b_{uv} + b_{vw}) \bmod B_u = b_{uv} \quad (7)$$

Assume that  $B_u | B_v$ . We use the definition of the  $b_{vw}$  given by Equation (3) to write that  $b_{vw} \bmod B_u = (D_{v,w-1} \bmod B_v) \bmod B_u = D_{v,w-1} \bmod B_u$ . Notice that  $b_{vw}$  is the rest in the Euclidean division by  $B_v$ , and not  $B_u$ , and thus the divisibility of  $B_v$  by  $B_u$  is required for the previous equation. Now writing also the definition of  $b_{uv}$  we get:  $(b_{uv} + b_{vw}) \bmod B_u = (D_{u,v-1} + D_{v,w-1}) \bmod B_u = D_{u,w-1} \bmod B_u$ , which is equal by definition to  $b_{uw}$ .

Based on this result, we now prove Property 3. Observe that, due to Property 1, the fact that  $u$  and  $v$  are two consecutive ordering periods implies that  $B_u < B_v$ , and thus we necessarily have  $B_u | B_v$  when considering a divisible batch size instance.

First, assume that  $s \geq b_{uv}$ , and thus  $\sigma_{uv}(s) = s - b_{uv}$ , see Equation (5). We have  $\sigma_{uv}(s) \geq b_{vw}$  if and only if:

$$\begin{aligned}
& s \geq b_{uv} + b_{vw} \\
\Leftrightarrow & (s \geq b_{uv} + b_{vw} \text{ and } B_u > b_{uv} + b_{vw}) \\
& \text{or } (s \geq b_{uv} + b_{vw} \text{ and } b_{uv} + b_{vw} \geq B_u) \\
\Leftrightarrow & (s \geq b_{uv} + b_{vw} \text{ and } B_u > b_{uv} + b_{vw}) \\
\Leftrightarrow & s \geq b_{uv} \text{ and } B_u > b_{uv} + b_{vw}
\end{aligned}$$

The equivalence between the first and the second lines is simply due to the tautology  $(b_{uv} + b_{vw} < B_u)$  or  $(b_{uv} + b_{vw} \geq B_u)$ . The equivalence between the second and the third lines comes from the fact that  $s < B_u$  in a dominant policy due to Property 1, which implies that the second clause is always false. Finally, the equivalence between the last two lines is due to Equation (7), since  $B_u > b_{uv} + b_{vw}$  is equivalent to  $b_{uv} + b_{vw} = (b_{uv} + b_{vw}) \bmod B_u$  for integers.

Second, consider that  $s < b_{uv}$ , and thus  $\sigma_{uv}(s) = s - b_{uv} + B_u$ . In a similar way we have  $\sigma_{uv}(s) \geq b_{vw}$  if and only if:

$$\begin{aligned}
& s + B_u \geq b_{uv} + b_{vw} \\
\Leftrightarrow & (s + B_u \geq b_{uv} + b_{vw} \text{ and } b_{uv} + b_{vw} < B_u) \\
& \text{or } (s + B_u \geq b_{uv} + b_{vw} \text{ and } B_u \leq b_{uv} + b_{vw} < 2B_u) \\
& \text{or } (s + B_u \geq b_{uv} + b_{vw} \text{ and } 2B_u \leq b_{uv} + b_{vw}) \\
\Leftrightarrow & (s + B_u \geq b_{uv} + b_{vw}) \\
& \text{or } (s + B_u \geq b_{uv} + b_{vw} \text{ and } B_u \leq b_{uv} + b_{vw} < 2B_u) \\
\Leftrightarrow & (b_{uv} + b_{vw} \leq B_u) \text{ or } (s \geq b_{uv} \text{ and } b_{uv} + b_{vw} < 2B_u)
\end{aligned}$$

The first equivalence simply uses the fact that the quantity  $(b_{uv} + b_{vw})$  is either smaller than  $B_u$ , or is in the interval  $[B_u, 2B_u)$ , or is greater than  $2B_u$ . The second equivalence comes from the fact that  $s < B_u$  due to Property 1, which implies that the third clause of the second line is always false. Again, Equation (7) allows to write the last equivalence, since  $B_u \leq b_{uv} + b_{vw} < 2B_u \Leftrightarrow b_{uv} + b_{vw} = b_{uv} + B_u$ .  $\square$

Property 3 implies that for any stock level  $s_{u-1}$  in the  $i$ th  $\alpha^u$ -interval, the resulting stock level  $\sigma_{uv}(s_{u-1})$  is in the same  $j$ th  $\alpha^v$ -interval, for some index  $j$ . We now turn our attention to the computation of this index. To simplify the forthcoming expressions, instead of representing the interval  $[\alpha_i^u, \alpha_{i+1}^u)$  by its index  $i$ , we choose to give a period  $\bar{v}$  such that  $b_{u\bar{v}} = \alpha_i^u$ . By convention, if  $\alpha_i^u = 0$ , we set  $\bar{v} = 0$  and  $b_{u0} = 0$ . Clearly, for any period  $v > u$ , if we know that  $\bar{v} = \arg \max_t \{ b_{ut} \mid b_{ut} \leq s \}$ , with the convention that  $\arg \max \emptyset = 0$ , then we have  $s \geq b_{uv}$  if and only if  $b_{uv} \leq b_{u\bar{v}}$ . Now let  $\bar{w}$  be a period such that  $b_{v\bar{w}} = \alpha_j^v$ . Property 3 allows us to determine  $\bar{w}$  from  $\bar{v}$  as follows:

If  $b_{uv} \leq b_{u\bar{v}}$  then

$$\bar{w} = \arg \max_w \{ b_{vw} \mid b_{uv} \leq b_{u\bar{v}} \text{ and } b_{vw} < B_u - b_{uv} \}$$

Similarly, if  $b_{uv} < b_{u\bar{v}}$ , then

$$\bar{w} = \arg \max_w \{ b_{vw} \mid (b_{uv} \leq b_{u\bar{v}}) \text{ and } (b_{vw} < 2B_u - b_{uv}) \text{ or } (b_{vw} < B_u - b_{uv}) \}$$

Notice that if we pre-compute all the values  $b_{uv}$ 's for any couple of periods and sort them for each period  $u$ , then determining  $\bar{w}$  for a given  $\bar{v}$  using the preceding expressions can be done in time  $O(\log T)$  by dichotomic search. We are now ready to give the description of the dynamic programming algorithm.

Let us denote  $\text{OPT}(u, \bar{v})$  the minimal cost to satisfy all the demands over the periods  $[u, T]$ , assuming that  $u$  is an ordering period, and given any entering stock level  $s$  such that  $b_{u\bar{v}} = \max_v \{ b_{uv} \mid b_{uv} \leq s \}$ . We can write the sub-optimality principle as:

$$\text{OPT}(u, \bar{v}) = f_u + \min_{v > u} \{ \lceil (D_{u,v-1} - b_{u\bar{v}}) / B_u \rceil k + \text{OPT}(v, \bar{w}) \}$$

In this dynamic programming, we have  $O(T^2)$  states to consider. For each state  $(u, \bar{v})$ , the computation of  $\text{OPT}(u, \bar{v})$  can be done in time  $O(T \log T)$  by inspection on the different periods  $v > u$ , each period  $v$  requiring the determination of  $\bar{w}$ . Since the precomputing and sorting of the  $b_{uv}$ 's is negligible compared to the time complexity of computing  $\text{OPT}(u, \bar{v})$  for all the states, we obtain the following result:

**Theorem 7.** *Problem UBLS  $(f_i/k/ - / -)$  with stationary fixed cost per batch and null unit procurement and holding costs can be solved in time  $O(T^3 \log T)$  if the batch sizes are divisible.*

## 7. Stationary setup cost and fixed cost per batch, no unit cost

We finally consider the case where only the batch sizes are time-dependent, all the cost parameters  $(f, k, p$  and  $h)$  being stationary. In addition we restrict our attention to a null unit holding cost, that is  $h = 0$ . For short we note this problem as  $(f/k/ - / -)$ . Notice that even with no holding cost, this problem becomes NP-hard as soon as one cost parameter is time-dependent. Without loss of generality we can assume that the batch sizes are increasing with respect to the periods, see Section 6.2. In the following section 7.1, we derive a polynomial time algorithm in  $O(T^3)$  time complexity using the property that the cost of a policy belongs to a restricted set of possible values. In Section 7.2 a linear time algorithm is derived for the case with no setup cost  $(-/k/ - / -)$ .

### 7.1. A polynomial time algorithm for UBLS ( $f/k/-/-$ )

The basic idea to derive a polynomial time algorithm is to observe that the cost of a policy belongs to a restricted set of possible values. Indeed, a policy making  $\alpha$  setups and ordering  $\beta$  batches over the time horizon incurs a cost of  $\alpha f + \beta k$ , whenever it does actually order. Thus we propose a dynamic programming approach where the states are based upon the cost paid by a policy.

To introduce this approach, consider the set  $\Pi(u, \alpha, \beta)$  of feasible plannings incurring exactly  $\alpha$  setups and ordering exactly  $\beta$  batches on the time interval  $[1, \dots, u-1]$ , and such that  $u$  is an ordering period. Recall that  $\text{OPT}(u, s)$ , introduced in section 6.2, denotes the minimal cost of a policy to satisfy the remaining demands on time interval  $[u, T]$  given an entering stock level of  $s$  at period  $u$ . The optimal cost over the whole horizon for a policy of  $\Pi(u, \alpha, \beta)$  assuming an entering stock level of  $s$  at period  $u$  is clearly equal to  $\alpha f + \beta k + \text{OPT}(u, s)$ . Now observe that  $\text{OPT}(u, s)$  is a non-increasing function of the stock level  $s$ . This is due to the fact that a feasible policy for a stock level  $s$  is also feasible for any stock level  $s' > s$ . Thus the minimal cost  $z^*(u, \alpha, \beta)$  of a planning belonging to  $\Pi(u, \alpha, \beta)$  is realized by the policy maximizing the stock level  $s_{u-1}$ . Let us define  $\bar{s}(u, \alpha, \beta)$  as the maximal stock level at the beginning of the period  $u$  over all plannings in  $\Pi(u, \alpha, \beta)$ . By convention we set  $\bar{s} = -\infty$  if  $\Pi(u, \alpha, \beta)$  is empty, that is no feasible planning can make only  $\alpha$  setups and orders only  $\beta$  batches over  $[1, \dots, u-1]$ . From what precedes we have

$$z^*(u, \alpha, \beta) = \alpha f + \beta k + \text{OPT}(u, \bar{s}(u, \alpha, \beta))$$

And obviously

$$\text{OPT} = \min_{\alpha, \beta} \{ \alpha f + \beta k \mid \bar{s}(T+1, \alpha, \beta) \geq 0 \} \quad (8)$$

Hence our goal is to compute dynamically the maximum stock level  $\bar{s}(u, \alpha, \beta)$  of a feasible policy. Due to Property 2, we can restrict to policies whose evolution of the stock levels obeys  $\sigma$ , that is  $s_{v-1} = \sigma_{uv}(s_{u-1})$  for any two successive ordering periods  $u$  and  $v$ . But it turns out that the computation of  $\bar{s}(u, \alpha, \beta)$  is quite intricate, for  $s \mapsto \sigma_{uv}(s)$  is not an increasing function, see Figure 3. In particular the entering stock level  $s$  at  $u$  maximizing the stock level  $\sigma_{uv}(s)$  at period  $v$  may be different from  $\bar{s}(u, \alpha, \beta)$ : for instance if  $\bar{s}(u, \alpha, \beta)$  equals  $b_{uv}$ , the resulting entering stock level  $\sigma_{uv}(b_{uv})$  is null while any stock level  $s < b_{uv}$  leads to a positive entering stock at period  $v$ .

However, we are not interested in computing  $\bar{s}(u, \alpha, \beta)$  for any values of  $\alpha$  and  $\beta$ , but ideally only for

the ones corresponding to an optimal planning. For this reason, we restrict our attention to a class of dominant policies, that we call the  $\underline{\beta}$ -policies. The following of this section is devoted to define these policies, to prove their dominance, and to show that they can be efficiently computed.

For a given period  $u$  and a given number of setups  $\alpha$ , we say that a number of batches  $\beta$  is minimal if  $\Pi(u, \alpha, \beta - 1)$  is empty and  $\Pi(u, \alpha, \beta)$  contains at least one feasible planning. That is, there exists no feasible planning with  $\alpha$  ordering periods and ordering strictly less than  $\beta$  batches over the time interval  $[1, \dots, u-1]$ . We denote by  $\underline{\beta}(u, \alpha)$  this minimal value. We have by definition:

$$\underline{\beta}(u, \alpha) = \min \{ \beta \mid \bar{s}(u, \alpha, \beta) \geq 0 \}$$

By convention we set  $\underline{\beta}(u, \alpha) = +\infty$  if there does not exist a feasible policy making only  $\alpha$  setups over  $[1, u-1]$ . Equation (8) immediately implies that  $\text{OPT} = \min_{\alpha} \{ \alpha f + \underline{\beta}(T+1, \alpha) k \}$ . Thus we can switch attention to computing dynamically the  $\underline{\beta}$ 's values instead of the  $\bar{s}$ 's values. To derive a recursive expression of  $\underline{\beta}$ , we consider the class of the  $\underline{\beta}$ -policies, using a minimal (overall) number of batches and having a maximal entering stock level at each period it orders :

**Definition 1.** A policy  $\pi$  of  $\Pi(u, \alpha, \beta)$  is  $\underline{\beta}$  at period  $u$  if the number  $\beta_u$  of batches incurred by  $\pi$  on  $[1, \dots, u-1]$  is minimal with respect to its number  $\alpha$  of setups, and its entering stock level  $s_{u-1}$  is maximal with respect to  $\alpha$  and  $\beta$ :

$$\beta = \underline{\beta}(u, \alpha) \text{ and } s_{u-1} = \bar{s}(u, \alpha, \beta)$$

We say that  $\pi$  is a  $\underline{\beta}$ -policy if  $\pi$  is  $\underline{\beta}$  at each of its ordering period.

Observe that for any period  $u$  and any number of setups  $\alpha$ , if there exists a feasible policy incurring  $\alpha$  setups over  $[1, u-1]$ , then, by definition of  $\underline{\beta}(u, \alpha)$ , there exists a feasible policy that is  $\underline{\beta}$  at period  $u$ . On the contrary, the existence of  $\underline{\beta}$ -policies is not immediate: as previously claimed, there does not necessarily exist a planning maximizing the entering stock level of each of its ordering periods. Property 4 proves that  $\underline{\beta}$ -policies do exist and are dominant.

**Property 4.**  $\underline{\beta}$ -policies are dominant

**Proof.** For short we denote by  $\underline{\Pi}(u, \alpha)$  the subset of policies  $\pi$  of  $\Pi(u, \alpha, \beta)$  which are  $\underline{\beta}$  at period  $u$ . By definition, it implies that  $\beta = \underline{\beta}(u, \alpha)$  and that the entering stock level at  $u$  is  $\bar{s}(u, \alpha, \beta)$ . We will establish that

any policy that is  $\underline{\beta}$  at a period  $v$  is also  $\underline{\beta}$  at its preceding ordering period  $u$ . By a direct induction, such a policy is thus  $\underline{\beta}$  at each of its ordering periods up to  $v$ . Property 4 follows, since for some value of  $\alpha^*$ , set  $\underline{\Pi}(T+1, \alpha)$  contains only optimal policies. Thus any policy of  $\underline{\Pi}(T+1, \alpha^*)$  is both an optimal policy and a  $\underline{\beta}$ -policy over the whole horizon.

Consider a period  $v$  and a number of setups  $\alpha > 0$  such that  $\underline{\Pi}(v, \alpha)$  is not empty. Let  $\pi$  be a policy of  $\underline{\Pi}(v, \alpha)$ , that is a  $\underline{\beta}$  at a period  $v$ . For any period  $t$ , we denote by  $\alpha_t$  and  $\beta_t$  the number of setups and the number of batches, respectively, incurred by  $\pi$  on the time horizon  $[1, \dots, t-1]$ , and by  $s_t$  the stock level at the end of period  $t$ . By definition of  $\underline{\Pi}(v, \alpha)$  we have  $\beta_v = \underline{\beta}(v, \alpha)$  and  $s_{v-1} = \bar{s}(v, \alpha, \beta_v)$ .

Let  $u$  be the preceding ordering period. Recall that we want to prove that  $\pi$  is also  $\underline{\beta}$  at period  $u$ . First, observe that the policy  $\pi$  obeys the relations of Property 2 between periods  $u$  and  $v$ . More precisely let  $m$  be the number of batches ordered by  $\pi$  at period  $u$ . Readily, for  $\pi$  to be feasible and to be minimal in the number of batches used over the time interval  $[1, \dots, v-1]$ , we should have  $m = m_{uv}(s_{u-1})$ . That is, the policy should order the minimum possible number of batches at period  $u$  to satisfy the demand till period  $v$ . As a consequence, the entering stock level  $s_{v-1}$  at period  $v$  is lower than  $B_u$ . Also, for  $s_{v-1}$  to be maximal with respect to  $\alpha_v$  and  $\beta_v$ , the entering stock level  $s_{u-1}$  at  $u$  should be less than  $B_u$ : Otherwise the same interchange procedure as in Property 1 would increase by at least one unit  $s_{v-1}$ . It results that the stock levels verify the relation  $s_{v-1} = \sigma_{uv}(s_{u-1})$ .

To prove that  $\pi$  is  $\underline{\beta}$  at period  $u$ , we compare it with another policy  $\pi'$ , constructed as follows.

Clearly the policy  $\pi$  incurs exactly  $\alpha_u = \alpha - 1$  setups on the time horizon  $[1, u-1]$ . In particular the set  $\underline{\Pi}(u, \alpha_u)$  is not empty, and we can choose a policy  $\underline{\pi} \in \underline{\Pi}(u, \alpha_u)$ , which is by definition  $\underline{\beta}$  at period  $u$ . So, we are in the situation where we have:

- a policy ( $\underline{\pi}$ ) which is  $\underline{\beta}$  at period  $u$ ; more precisely  $\underline{\pi} \in \underline{\Pi}(u, \alpha - 1)$
- a policy ( $\pi$ ) which is  $\underline{\beta}$  at period  $v$ ; more precisely  $\pi \in \underline{\Pi}(v, \alpha)$

Using intuitive notations, we denote in the following with  $a'$  all the quantities relative to the policy  $\pi'$ . The policy  $\pi'$  simply follows the policy  $\underline{\pi}$  till period  $u$ . At period  $u$ ,  $\pi'$  orders  $m' = m_{uv}(s'_{u-1}) = \lceil (D_{u,v-1} - s'_{u-1})/B_u \rceil$  batches, according to its entering stock level  $s'_{u-1}$ , that is the minimal number of batches  $m_{uv}$  to satisfy the demands till period  $v$ . Notice that our choice of  $\underline{\pi}$  implies that  $s'_{u-1} = \bar{s}(u, \alpha_u, \beta'_u)$ , which is maximal with respect

to  $\alpha_u$  and  $\beta'_u$ . Finally on the time interval  $[v, T+1]$ , the policy  $\pi'$  follows an optimal policy. By construction, the stock level  $s'_{v-1}$  at period  $v$  can not be negative, since we have chosen  $m'$  such that  $s'_{v-1} = \sigma_{uv}(s'_u)$ . It follows that  $\pi'$  is feasible, and belongs to the set  $\underline{\Pi}(v, \alpha, \beta'_v)$ .

By construction, the policy  $\pi'$  is  $\underline{\beta}$  at period  $u$ . Since both policies incur the same number of setups over  $[1, \dots, u-1]$ , to prove that the policy  $\pi$  is also  $\underline{\beta}$  period  $u$ , it is sufficient to show that  $\beta_u = \beta'_u$  and  $s_{u-1} = s'_{u-1}$ . Recall that  $m$  and  $m'$  are the number of batches ordered at period  $u$  by the policies  $\pi$  and  $\pi'$ , respectively. With our notations we thus have  $\beta_v = \beta_u + m$  and  $\beta'_v = \beta'_u + m'$ . Also observe that we have  $\alpha_u = \alpha'_u = \alpha - 1$  and  $\alpha_v = \alpha'_v = \alpha$  by construction. Hence using the fact that the policy  $\pi'$  is  $\underline{\beta}$  at period  $u$ , we obtain  $\beta'_u \leq \beta_u$ . In the same way, since the policy  $\pi$  is  $\underline{\beta}$  at period  $v$ , we have  $\beta_v \leq \beta'_v$ . We thus have :

$$\beta'_u \leq \beta_u \quad \text{and} \quad \beta_u + m \leq \beta'_u + m' \quad (9)$$

Observe that Equation 9 implies in particular that  $m \leq m'$ , that is at period  $u$  the policy  $\pi'$  orders at least the same number of batches as  $\pi$ . We first establish that  $\beta_u = \beta'_u$ .

For the sake of contradiction, assume that  $\beta_u \neq \beta'_u$ . Due to Equation 9 we thus have  $\beta'_u < \beta_u$ , which implies that  $m < m'$ . Observe that  $m'$  is the minimum number of batches to order to satisfy the demands till period  $v$  with an entering stock level  $s'_{u-1}$  at period  $u$ . Thus, necessarily we have  $s_{u-1} > s'_{u-1}$ . We are in the situation where the policy  $\pi$  has ordered more batches on the time interval  $[1, \dots, u-1]$  than the policy  $\pi'$ , thus ending with a larger inventory at period  $u$ , which enables it in its turn to order less batches to cover the demands till period  $v$ . We now show that it would contradict the fact that  $\pi$  is  $\underline{\beta}$  at period  $v$ . Recall that the number  $m$  of batches ordered by  $\pi$  at period  $u$  is also minimal with respect to its entering stock  $s_{u-1}$ , that is  $m = m_{uv}(s_{u-1})$ . Now, since

$$m_{uv}(s) = \begin{cases} \lfloor D_{u,v-1}/B_u \rfloor + 1 & \text{if } s < b_{uv} \\ \lfloor D_{u,v-1}/B_u \rfloor & \text{if } s \geq b_{uv} \end{cases} \quad (10)$$

it is clear that the minimal number of batches  $m_{uv}$  to order differs by at most one unit depending on the current inventory level. Thus we necessarily have  $m' = m + 1$ , which implies, again according to Equation (9), that  $\beta'_v = \beta_v$ . Turning our attention to the stock levels, we have by construction  $s'_{v-1} = s'_{u-1} + (m+1)B_u - D_{u,v-1} \geq B_u + mB_u - D_{u,v-1}$ . Similarly  $s_{v-1} = s_{u-1} + mB_u - D_{u,v-1}$ . As already noticed, we can assume that  $s_u < B_u$ , which implies that  $s'_{v-1} > s_{v-1}$ . As a conclusion, both policies  $\pi$  and  $\pi'$  belongs to the set  $\underline{\Pi}(v, \alpha_v, \beta_v)$ , but the policy  $\pi'$  has a

larger entering stock level at period  $v$ . This contradicts the fact that  $\pi$  is  $\underline{\beta}$  at period  $v$ .

As a consequence, we necessarily have  $\beta_u = \beta'_u$ . To be able to conclude that  $\pi$  is also  $\underline{\beta}$  at period  $u$ , we need to establish now that  $s_{u-1} = s'_{u-1}$ . Since  $\pi'$  is  $\underline{\beta}$  at period  $u$ , its entering stock level  $s'_{u-1}$  is the largest possible among all policies of  $\Pi(u, \alpha'_u, \beta'_u)$ . It implies that  $s_{u-1} \leq s'_{u-1}$ . For the sake of contradiction, assume that  $s_{u-1} < s'_{u-1}$ . We show that, again, it would contradict the fact that  $\pi$  is  $\underline{\beta}$  at period  $v$ . Observe that the policy  $\pi'$  orders at period  $u$  the minimal number of batches, with respect to its stock level, to satisfy the demands till the beginning of period  $v$ . Hence for the policy  $\pi$  to be feasible starting from a lower inventory level than  $\pi'$ , certainly it has to order at least  $m'$  batches at period  $u$ . Together with Equation 9, we can conclude that  $m' = m$  (both policies order the same number of batches at period  $u$ ) and by consequence that  $\beta'_v = \beta_v$ . In addition, the fact that  $s_{u-1} < s'_{u-1}$  and that both policies order the same amount  $mB_u$  of units at period  $u$  obviously implies that  $s_{v-1} < s'_{v-1}$ . We arrive at the same situation that both policies  $\pi$  and  $\pi'$  belong to the set  $\Pi(v, \alpha_v, \beta_v)$ , but the policy  $\pi'$  has a larger entering stock level at period  $v$ . This contradicts the fact that  $\pi$  is  $\underline{\beta}$  at period  $v$ .

As a consequence we have  $\beta_u = \beta'_u \equiv \underline{\beta}(u, \alpha - 1)$  and  $s_{u-1} = s'_{u-1} \equiv \bar{s}(u, \alpha - 1, \beta'_u)$ . It implies that the policy  $\pi$  is  $\underline{\beta}$  at period  $u$ , which concludes the proof.  $\square$

Property 4 allows us to compute dynamically the different values  $\underline{\beta}(u, \alpha)$ , using the fact that there exists a policy in  $\underline{\Pi}(u, \alpha)$  using a minimal number of batches at each of its ordering period. In addition, the proof of Property 4 also shows that Property 2 is also dominant for  $\underline{\beta}$  policies. The definition of  $\underline{\beta}$  still requires to compute the maximum stock level  $\bar{s}(u, \alpha, \beta)$ , but we can restrict ourselves to the following function  $\hat{s}(u, \alpha, \beta)$  defined by:

$$\hat{s}(u, \alpha, \beta) = \begin{cases} \bar{s}(u, \alpha, \beta) & \text{if } \beta = \underline{\beta}(u, \alpha) \\ -\infty & \text{otherwise} \end{cases}$$

The values  $\underline{\beta}$  and  $\hat{s}$  can be jointly computed dynamically as follows: For any period  $v$  and any value  $\alpha \geq 1$ :

$$\underline{\beta}(v, \alpha) = \min_{u < v} \begin{cases} \underline{\beta}(u, \alpha - 1) + \lfloor D_{u,v-1}/B_u \rfloor & \text{if } \hat{s}(u, \alpha - 1, \beta - \lfloor D_{u,v-1}/B_u \rfloor) \geq b_{uv} \\ \underline{\beta}(u, \alpha - 1) + \lceil D_{u,v-1}/B_u \rceil & \text{if } \hat{s}(u, \alpha - 1, \beta - \lceil D_{u,v-1}/B_u \rceil) < b_{uv} \end{cases}$$

and for  $\beta = \underline{\beta}(v, \alpha) \geq 1$ , we have :

$$\hat{s}(v, \alpha, \beta) = \max_{u < v} \begin{cases} \hat{s}(u, \alpha - 1, \beta - \lfloor D_{u,v-1}/B_u \rfloor) - b_{uv} \\ \hat{s}(u, \alpha - 1, \beta - \lceil D_{u,v-1}/B_u \rceil) - b_{uv} + B_u \end{cases}$$

and  $\hat{s}(v, \alpha, \beta) = -\infty$  for  $\beta \neq \underline{\beta}(v, \alpha)$ .

For  $\alpha = 0$ , the basis of the recursion is given for any period  $u$  by:

$$\underline{\beta}(u, 0) = \begin{cases} 0 & \text{if } D_{1,u-1} = 0 \\ +\infty & \text{otherwise} \end{cases}$$

and  $\hat{s}(u, 0, \beta) = -\underline{\beta}(u, 0)$

These expressions simply correspond to the behavior of a dominant policy as defined by Property 2. We obtain the following theorem :

**Theorem 8 (Polynomiality of  $(f/k) - (-)$ ).** *Problem UBLS with stationary setup cost, stationary fixed cost per batch and null unit procurement and holding costs can be solved in time complexity  $O(T^3)$ .*

**Proof.** The dynamic computation of the  $\underline{\beta}$  values requires to evaluate  $O(T^2)$  states. The value of each  $\underline{\beta}(v, \alpha)$  can be determined in time  $O(T)$  if all quantities  $b_{uv}$  are precomputed (this requires only  $O(T^2)$  operations). For the stock levels  $\hat{s}(v, \alpha, \beta)$ , we have  $O(T^3)$  states to explore. However for a given pair  $(v, \alpha)$ , only the computation for the case  $\beta = \underline{\beta}(v, \alpha)$  requires  $O(T)$  operations. For all other values of  $\beta$ , the computation of  $\hat{s}(v, \alpha, \beta)$  requires only  $O(1)$  operations. The determination of the optimal value OPT is then performed in time  $O(T)$ . The overall time complexity is hence in  $O(T^3)$ .  $\square$

## 7.2. A more efficient algorithm for UBLS $(-/k) - (-)$

In the restricted case with no setup cost,  $(-/k) - (-)$ , the problem can be solved very efficiently in  $O(T)$  by a simple greedy algorithm. First observe that the cheapest way to satisfy a demand is to order in a previous period with the largest possible batch size. Thus an optimal planning orders the first (non null) demand, say  $d_t$ , in the period  $u \leq t$  with the largest batch capacity. The number of full batches to order to satisfy the demand  $d_t$  is  $\lceil \frac{d_t}{B_u} \rceil$ . The algorithm then sequentially scans through the periods till the demand of a period  $t'$  can not be entirely satisfied by the quantity ordered at  $u$ . The algorithm again should order in the period before  $t'$  with the largest batch size, that is  $u' = \arg \max\{B_i | 1 \leq i \leq t'\}$ , to serve this remaining demand in  $t'$ . We repeat these steps until the end of the horizon is reached. A straightforward implementation requires  $O(T^2)$  steps, since at each period we may have to search for the largest batch size. However we can use the fact that for any periods  $t < t'$  we have  $\max\{B_i | 1 \leq i \leq t'\} = \max\{\max\{B_i | 1 \leq i \leq t\}, \max\{B_i | t+1 \leq i \leq t'\}\}$  to find for each period  $t$  the largest batch size period  $u \leq t$  in constant time. Thus the overall complexity of the algorithm is linear.

## 8. Conclusion and perspectives

In this paper we have studied the complexity of the UBLS problem with time-dependent batch sizes, according to the behavior of the cost parameters. To the best of our knowledge no anterior complexity study has been published on the time-varying batch size cases. We have proven that UBLS is NP-hard as soon as one of the cost parameters is allowed to be time-dependent. On the contrary, when all cost parameters are stationary (but assuming no holding cost), we have established that UBLS with time-dependent batch sizes can be solved in time  $O(T^3)$ . Another non trivial polynomial case corresponds to UBLS with the cost structure  $(f_i/k_i - /-)$  and divisible batch sizes, for which an  $O(T^3 \log(T))$  algorithm is proposed.

There are two remaining open cases to be explored for the UBLS problem with time-dependent batch sizes, namely  $(f/k_i - /h)$  and  $(-/k_i - /h)$ . They correspond to the cases of stationary cost parameters but non null holding cost. Indeed, if a stationary procurement cost is equivalent to a null procurement cost, this is not the case for the holding cost. While these problems are NP-hard for time-dependent holding cost and polynomial for a null holding cost, their complexity status is open for a stationary holding cost.

Another open question is the approximability of the UBLS problem. Since the UBLS problem is NP-hard in the ordinary sense, one may hope that it belongs to fully polynomial time approximation scheme. It would be of interest to answer, positively or negatively, to the existence of a fully approximation scheme for this problem.

The polynomial time algorithms proposed in this paper are based on a dynamic programming approach. In the literature, polyhedral approaches have also been investigated for the CLSP with batch delivery. Akbalik and Pochet (2009) propose new classes of valid inequalities for the CLSP with the cost structure  $(f_i/k_i/p_i/h_i)$ , a stationary production capacity and a stationary batch size. Pochet and Wolsey (1993) gives a linear formulation with  $O(T^3)$  variables and constraints for the UBLS problem with the cost structure  $(-/k_i/p_i/h_i)$ . The reader can find a detailed description of the polyhedral approach for various extensions of the lot sizing problem in Pochet and Wolsey (2006). One perspective is to look for interesting valid inequalities for the UBLS problem with a general cost structure  $(f_i/k_i/p_i/h_i)$ , which is proven in our paper to be NP-hard. For the different polynomial cases, finding a compact linear formulation would also be of great interest.

## Acknowledgments

We are grateful to the anonymous referees for their very careful reading and suggestions which helped to improve the presentation of this paper.

## References

- A. Akbalik, Y. Pochet, Valid inequalities for the single-item capacitated lot sizing problem with stepwise costs, *European Journal of Operational Research* 198 (2009) 412–434.
- A. Akbalik, C. Rapine, Polynomial time algorithms for the constant capacitated single-item lot sizing problem with stepwise production cost, *Operations Research Letters*, 40 (2012) 390–397.
- A. Aggarwal, J.K. Park. Improved algorithm for economic lot-size problems, *Operations Research*, 41 (3) (1993) 549–571.
- S. Böcker, Z. Lipták, A Fast and Simple Algorithm for the Money Changing Problem, *Algorithmica* 48 (2007) 413–432.
- M. Florian, M. Klein, Deterministic production planning with concave costs and capacity constraints, *Management Science* 18 (1) (1971) 12–20.
- M.J.R. Helmrich, R. Jans, W. van den Heuvel, A.P.M. Wagelmans, The economic lot-sizing problem with an emission constraint, *Econometric Institute Report EI 2011-41*.
- C.-Y. Lee, A solution to the multiple setup problem with dynamic demand, *IIE Transactions* 21 (3) (1989) 266–270.
- C.-L. Li, V.N. Hsu, W.-Q. Xiao, Dynamic lot sizing with batch ordering and truckload discounts, *Operations Research* 52 (4) (2004) 639–654.
- S.A. Lippman, Optimal inventory policy with multiple set-up costs, *Management Science* 16 (1) (1969) 118–138.
- G.S. Lueker, Two NP-Complete Problems in Nonnegative Integer Programming, Technical Report TR-178, Department of Electrical Engineering, Princeton University (1975).
- Y. Pochet, L.A. Wolsey, Lot-sizing with constant batches: Formulation and valid inequalities, *Mathematics of Operations Research* 18 (4) (1993) 767–785.
- Y. Pochet, L.A. Wolsey, *Production Planning by Mixed Integer Programming*, Springer, 2006.
- D.X. Shaw, A.P.M. Wagelmans, An Algorithm for Single-Item Capacitated Economic Lot Sizing with Piecewise Linear Production Costs and General Holding Costs, *Management Science* 44 (6) (1998) 831–838.
- M. Van Vyve, Algorithms for single-item lot-sizing problems with constant batch size, *Mathematics of Operations Research* 32 (3) (2007) 594–613.
- A. Wagelmans, S. van Hoesel, A. Kolen, Economic Lot Sizing: An  $O(n \log n)$  Algorithm That Runs in Linear Time in the Wagner-Whitin Case, *Operations Research* 40 (1) (1992) S145–S156.
- H.M. Wagner and T.M. Whitin, Dynamic version of the economic lot size model, *Management Science* 5 (1) (1958) 89–96.