2007/48

■

Single item lot-sizing with non-decreasing capacities

Yves Pochet and Laurence Wolsey

CORE DISCUSSION PAPER
2007/48

Single item lot-sizing with non-decreasing capacities

Yves POCHET [1] and Laurence A. WOLSEY[2]

August 2007

Abstract

We consider the single item lot-sizing problem with capacities that are non-decreasing over time. When the cost function is i) non-speculative or Wagner-Whitin (for instance, constant unit production costs and non-negative unit holding costs), and ii) the production set-up costs are non-increasing over time, it is known that the minimum cost lot-sizing problem is polynomially solvable using dynamic programming.

When the capacities are non-decreasing, we derive a compact mixed integer programming reformulation whose linear programming relaxation solves the lot-sizing problem to optimality when the objective function satisfies i) and ii). The formulation is based on mixing set relaxations and reduces to the (known) convex hull of solutions when the capacities are constant over time.

We illustrate the use and effectiveness of this improved LP formulation on a new test instances, including instances with and without Wagner-Whitin costs, and with both non-decreasing and arbitrary capacities over time.

Keywords: lot-sizing, mixing set relaxation, compact reformulation, production planning, mixed integer programming

---

[1] CORE and Louvain School of Management (IAG-LSM), Université catholique de Louvain, Belgium.

[2] CORE and INMA, Université catholique de Louvain, Belgium. L. Wolsey is also member of ECORE, the newly created association between CORE and ECARES. E-mail: Laurence.wolsey@uclouvain.be

This work was partly carried out within the framework of ADONET, a European network in Algorithmic Discrete Optimization, contract no. MRTN-CT-2003-504438.

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

1 Introduction

Single item lot-sizing with capacities that vary over time is known to be $\mathcal{NP}$-hard. However a little known result of Bitran and Yanasse establishes that with non-speculative (Wagner-Whitin) production and storage costs, non-decreasing capacities and non-increasing set-up costs, there is a polynomial time dynamic programming algorithm.

The main goal in this paper is to develop a mixed integer programming formulation whose linear programming relaxation solves the lot-sizing problem in this special case. The MIP formulation that we propose has the following features:
i) Its linear programming relaxation solves the lot-sizing problem in the special case
ii) The approach taken is not to develop facet-defining inequalities for the convex hull of feasible solutions, but rather to construct an alternative relaxation for which a tight linear programming (convex hull) representation is known
iii) When the capacities are constant over time, the formulation reduces to the standard formulation used in the Wagner-Whitin case
iv) Whatever the costs, the formulation is valid for the lot-sizing problem with non-decreasing capacities, and can be shown to provide improved solution times on a variety of instances. In addition, it can be adapted for instances with arbitrary capacities.

We now discuss related work. Although most variants of single item lot-sizing with varying capacities, denoted *LS-C* are *NP*-hard, see Florian et al. [9], the single item lot-sizing problem with constant capacities over time, denoted *LS-CC*, is polynomially solvable. This was proved by Florian and Klein [8] with a dynamic programming algorithm running in $O(n^4)$, where n is the number of time periods in the planning horizon. This complexity was later improved to $O(n^3)$ by van Hoesel and Wagelmans [10]. A tight and compact extended formulation for *LS-CC* was proposed by Pochet and Wolsey [13], involving $O(n^3)$ variables and constraints. An explicit linear description of the convex hull of solutions in the original space of variables ($O(n)$ production, setup and inventory variables) is still not known, although a large class of facet defining valid inequalities (the so-called (k, l, S, I) inequalities) was identified in Pochet and Wolsey [13].

These results can be improved in the special case of *LS-CC* in which the objective function satisfies the so-called Wagner-Whitin cost conditions. This problem is denoted by *WW-CC*. The *WW* cost conditions assume that there are no speculative motives to hold inventory, i.e., it always pays to produce as late as possible for any given set of production periods. For WW-CC, Van Vyve [16] proposed an optimization algorithm running in $O(n^2 \log n)$, and Pochet and Wolsey [14] gave a tight and compact reformulation with $O(n^2)$ variables and constraints. The latter was based on a reformulation of the stock minimal solutions leading to mixing set relaxations. They also gave a complete linear description in the original variable space with an exponential number of constraints, and a separation algorithm running in $O(n^2 \log n)$.

As indicated above, the problem LS-C is $\mathcal{NP}$-hard, see [9, 2]. Nothing appears to be known about reformulations for $LS-C$, or any of its variants, apart from the valid inequalities proposed by Pochet [12], derived from flow cover inequalities, and the submodular and lifted submodular inequalities proposed by Atamtürk and Munoz [1]. Most of the results cited above are described in detail in the recent book of Pochet and Wolsey [15].

Here we consider the single item lot-sizing problem with non-decreasing capacities over time, denoted LS-$C(ND)$, and more specifically the case in which the cost function is non-speculative or Wagner-Whitin and, in addition, the production set-up costs are non-increasing over time. This special case is denoted WW^*-$C(ND)$. Bitran and Yanasse [2] showed that WW^*-$C(ND)$ is polynomially solvable. They gave a polynomial time dynamic programming algorithm running in $O(n^4)$. An improved $O(n^2)$ algorithm was proposed later by Chung and Lin [3]. Thus problem WW^*-$C(ND)$ is one of the very few lot-sizing problem with varying capacities for which there is some hope to find a good formulation.

Outline. In Section 2 we describe the two relaxations on which our result is based, and present the main results of the paper. Specifically we describe the relaxation that provides a tight formulation for problem WW-C, as well as a tight extended linear programming formulation for WW-CC. This in turn motivates the second (mixing) set relaxation used to build an improved formulation of problem LS-$C(ND)$.

Sections 3 and 4 are devoted to a proof of the main result. In Section 3 we show that the right hand side values of the constraints defining the relaxation can be constructed in polynomial time, as well as deriving certain properties linking these values. In Section 4 we prove that the mixing set relaxation solves problem WW^*-$C(ND)$. In Section 5 we report on computational tests. Finally, in Section 6 we discuss future directions of research and the use of other mixing set relaxations to build improved formulations for various lot-sizing problems.

2 Formulations and Results

2.1 An Initial MIP Formulation

The single-item lot-sizing problem $LS-C$ is described by the following data. There are n time periods. For each time period t, p'_t, q_t and h'_t represent the unit production cost, the fixed production set-up cost and the unit inventory cost per period, respectively.

The other data defining the problem are the demand D_t and the production capacity C_t in each period t. For feasibility, we assume that $\sum_{i=1}^{t} C_t \geq \sum_{i=1}^{t} D_t$. We assume also that $0 \leq D_t \leq C_t$ for all t. The assumption that $D_t \leq C_t$ is made without loss of generality. This holds because when $D_t > C_t$ it is impossible to produce the amount $D_t - C_t$ in period t. Therefore D_t can be replaced by C_t, the amount $D_t - C_t$ must be produced before period t and can be added to D_{t-1}.

Throughout the paper we use the notation $D_{kt} \equiv \sum_{u=k}^{t} D_u$ when $1 \leq k \leq t \leq n$, and $D_{kt} \equiv 0$ otherwise, and similarly $y_{kt} \equiv \sum_{u=k}^{t} y_u$.

We now present a standard mixed integer programming formulation for $LS - C$.

The decision variables are x_t, y_t and s_t. They model the production lot size in period t, the binary set-up variable which must be set to one when there is positive production in period t, and the inventory at the end of period t, respectively. The initial formulation of problem $LS - C$ is

$$
\begin{aligned}
Z^{LS-C} := \min \quad & \sum_{t=1}^{n} \; (p'_t\, x_t + q_t\, y_t + h'_t\, s_t) && && (1)\\
& s_{t-1} + x_t = D_t + s_t && \text{for } 1 \leq t \leq n && (2)\\
& s_0 = s_n = 0 && && (3)\\
& x_t \leq C_t\, y_t && \text{for } 1 \leq t \leq n && (4)\\
& x_t, s_t \geq 0,\ \ y_t \in \{0,1\} && \text{for } 1 \leq t \leq n, && (5)
\end{aligned}
$$

where the objective (1) is to minimize the sum of production and inventory costs, under the demand satisfaction constraint (2) imposing that the demand D_t in each period t can be satisfied by producing some quantity x_t

in period t or by holding some inventory s_{t-1} from period $t-1$. Constraint (4) forces the set-up variable y_t to take the value 1 when there is a positive production in period t, i.e., $x_t > 0$, and limits the amount produced to C_t. Finally, constraint (3) says that there is no initial and final inventory, and constraint (5) defines the nonnegativity and binary restrictions on the variables.

The costs are non-speculative or Wagner-Whitin (WW) if

$$p'_t + h'_t \geq p'_{t+1} \text{ for all } t.$$

The set-up costs are non-increasing (WW^*) if in addition

$$q_t \geq q_{t+1} \text{ for all } t.$$

The capacities are nondecreasing ($C(ND)$) when

$$C_t \leq C_{t+1} \text{ for all } t.$$

Using the equations (2), it is a simple calculation to show that the variable costs $\sum_{t=1}^n p'_t x_t + \sum_{t=1}^n h'_t s_t = \sum_{t=1}^n p_t x_t + K_1 = \sum_{t=1}^n h_t s_t + K_2$ where $p_t = p'_t + \sum_{u=t}^n h'_u$, and $h_t = p'_t + h'_t - p'_{t+1}$ for all t. Note that the WW condition becomes $p_t \geq p_{t+1}$ for all t, or equivalently $h_t \geq 0$ for all t.

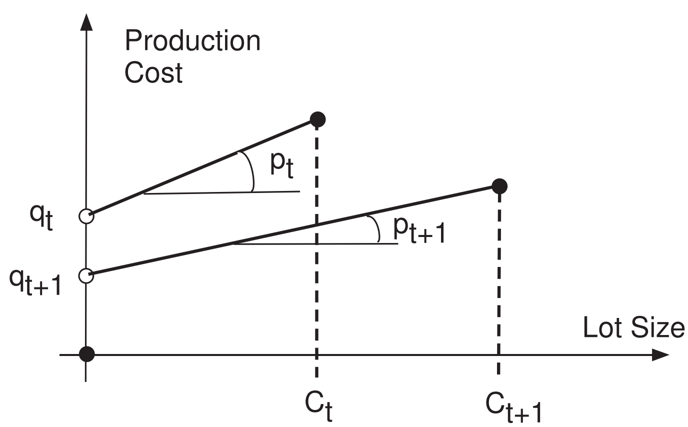

Figure 1: The WW^* cost conditions with variable costs $\sum_{t=1}^n p_t x_t$

As can be seen in Figure 1, the $WW^* - C(ND)$ conditions imply that it always pays to produce as late as possible. In other words, any full batch of size C_t produced in some period t, but not used to satisfy demand in period t, can always be postponed to period $t+1$, where the production and set-up costs will be at least as small, and the capacity at least as large as in period t.

2.2 The Wagner-Whitin Relaxation of LS-C

Aggregating the flow balance constraints (2) for periods $k, \ldots, l$ and using the capacity constraints (4) leads to the first well-known relaxation:

$$\min \sum_{t=1}^{n} h_t s_t + \sum_{t=1}^{n} q_t y_t \quad (6)$$
$$s_{k-1} + \sum_{u=k}^{t} C_u y_u \geq D_{kt} \quad \text{for } 1 \leq k \leq t \leq n \quad (7)$$
$$s_0 = s_n = 0 \quad (8)$$
$$s \in \mathbb{R}_+^{n+1}, y \in \{0,1\}^n \quad (9)$$

with feasible region X^{WW-C}.

The following well-known results indicate why Wagner-Whitin costs lead to special results.

Proposition 1. *[14] In an extreme point of* $\text{conv}(X^{WW-C})$*,*
i) $s_{k-1} = \max_{k=t,\ldots,n}(D_{kt} - \sum_{u=k}^{t} C_u y_u)^+$ *for* $1 \leq k \leq t \leq n$
ii) $0 \leq D_k + s_k - s_{k-1} \leq C_k y_k$ *for* $1 \leq k \leq n$.

Therefore any extreme point of $\text{conv}(X^{WW-C})$ defines a feasible solution of $LS-C$ by taking $x_k = D_k + s_k - s_{k-1}$. This immediately shows the interest of this relaxation.

Theorem 1. *The Wagner-Whitin relaxation*

$$\min\{hs + qy : (s,y) \in X^{WW-C}\}$$

solves $WW - C$ *(i.e. solves problem* $LS - C$ *in the presence of Wagner-Whitin costs).*

Solutions of (2)-(5) satisfying i) of Proposition 1 are called stock-minimal solutions. So Theorem 1 says that with WW costs (i.e. $h_t \geq 0$ for all t), there always exists an optimal stock-minimal solution to $WW - C$.

A second important result concerns the special case when the capacities are constant over time, in which case the set of solutions to (7)-(9) is denoted X^{WW-CC}. Note that X^{WW-CC} can be rewritten as the intersection of n sets, called *mixing* sets, all having a similar structure, namely

$$X^{WW-CC} = \bigcap_{k=1}^{n} X_k^{MIX}$$

where

$$\begin{aligned} X_k^{MIX} \quad = \quad & \{(s_{k-1}, y_k, \ldots, y_n) \in R_+^1 \times \{0,1\}^{n-k+1} : \\ & s_{k-1}/C + y_{kt} \geq D_{kt}/C \text{ for } t = k, \ldots, n\}. \end{aligned}$$

There are two important results concerning such sets.

Theorem 2. *[11]*

$$\text{conv}(X^{WW-CC}) = \bigcap_{k=1}^{n} \text{conv}(X_k^{MIX}).$$

Theorem 3. *[11, 15] A tight and compact extended formulation of* $\text{conv}(X_k^{MIX})$ *is given by*

$$s_{k-1} = C\mu^k + C\textstyle\sum_{j=k}^{n} f_j^k \sigma_j^k$$
$$\textstyle\sum_{j=k}^{n+1} \sigma_j^k = 1$$
$$\mu^k + y_{kt} + \textstyle\sum_{j: f_j^k \geq f_t^k} \sigma_j^k \geq \lfloor \frac{D_{kt}}{C} \rfloor + 1 \text{ for } k \leq t \leq n$$
$$\mu^k \in \mathbb{R}_+^1, y_t \in [0,1] \text{ for } k \leq t \leq n,\ \sigma_j^k \in \mathbb{R}_+^1 \text{ for } k \leq j \leq n+1$$

where $f_t^k = \frac{D_{kt}}{C} - \lfloor \frac{D_{kt}}{C} \rfloor$ *and* $f_{n+1}^k = 0$.

These results suggest that, if we can build a relaxation of $X^{WW-C(ND)}$ that is an intersection of mixing sets, it is then easy to describe the convex hull.

2.3 A Mixing Set Relaxation for $WW - C(ND)$

Here we assume both Wagner-Whitin costs and non-decreasing capacities. The feasible region (7)-(9) is denoted by $X^{WW-C(ND)}$ when C_t is non-decreasing over time, and by $X_0^{WW-C(ND)}$ when, in addition, the constraint $s_0 = 0$ is relaxed to $s_0 \geq 0$.

The right hand-side values that we will need to construct our relaxation are obtained by solving the problem:

$$(P_{kt}) \qquad \delta_{kt} = \min\{s_{k-1} + C_k \sum_{u=k}^{t} y_u : (s,y) \in X_0^{WW-C(ND)}\} \tag{10}$$

for $1 \leq k \leq t \leq n$. We can now describe the second relaxation of $WW - C(ND)$.

$$\min \textstyle\sum_{t=1}^{n} h_t s_t + \sum_{t=1}^{n} q_t y_t \tag{11}$$
$$s_{k-1} + C_k y_{kt} \geq \delta_{kt} \text{ for } 1 \leq k \leq t \leq n \tag{12}$$
$$s_0 = s_n = 0, \tag{13}$$
$$s \in \mathbb{R}_+^{n+1}, y \in \{0,1\}^n \tag{14}$$

with feasible region $X_R^{WW-C(ND)}$. Note that $X^{WW-C(ND)} \subseteq X_R^{WW-C(ND)}$ because all the constraints (12) are valid for $X^{WW-C(ND)}$ by definition of the δ_{kt}.

Our main result can now be stated.

Theorem 4. *The mixing set relaxation*

$$\min\{hs + qy : (s, y) \in X_R^{WW-C(ND)}\}$$

solves $WW^* - C(ND)$.

Example 1. *Consider the instance of WW^*-$C(ND)$ represented in Figure 2. For $k = 2$ and for all $t \in \{2, \ldots, 6\}$, the constraints (12) in the mixing*

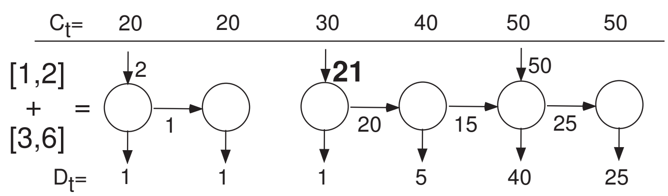

Figure 2: An instance of WW^*-$C(ND)$

set relaxation are the following:

$$\begin{aligned}
&s_1 + 20\, y_2 && \geq \delta_{22} = 1 \\
&s_1 + 20\, (y_2 + y_3) && \geq \delta_{23} = 2 \\
&s_1 + 20\, (y_2 + y_3 + y_4) && \geq \delta_{24} = 7 \\
&s_1 + 20\, (y_2 + y_3 + y_4 + y_5) && \geq \delta_{25} = 27 \\
&s_1 + 20\, (y_2 + y_3 + y_4 + y_5 + y_6) && \geq \delta_{26} = 41.
\end{aligned}$$

The feasible point represented in Figure 2 with $s_1 = 1$, $y_3 = y_5 = 1$ is the optimal solution of (10) for $k = 2$ and $t = 6$ obtained in computing δ_{26}.

As this relaxation is the intersection of n mixing sets, its convex hull is known. What is more the δ_{kt} can be calculated in polynomial time.

Theorem 5. *i) The mixing set relaxation (11)-(14) can be constructed explicitly in polynomial time.*
ii) $\mathrm{conv}(X_R^{WW-C(ND)}) = \bigcap_{k=1}^{n} \mathrm{conv}(X_k^{MIX^*})$ *where*

$$X_k^{MIX^*} = \{(s_{k-1}, y_k, \ldots, y_n) \in \mathbb{R}^1_+ \times [0,1]^{n-k+1} : s_{k-1}/C_k + y_{kt} \geq \delta_{kt}/C_k \text{ for } k \leq t \leq n,\} \quad (15)$$

and conv($X_k^{MIX^*}$) *is given by Theorem 3 (with C_k in place of C and δ_{kt} in place of D_{kt}).*
iii) The linear program

$$\min\{hs + qy : (s,y) \in \text{conv}(X_R^{WW-C(ND)})\}$$

solves $WW^* - C(ND)$.
iv) There is an extended formulation for conv($X_R^{WW-C(ND)}$) *with $O(n^2)$ constraints and $O(n^2)$ variables, or alternatively there is a $O(n^2 \log n)$ separation algorithm in the (s,y) space.*

The next two sections are devoted to the proof of Theorem 4. Theorem 5 ii)-iv) is a direct consequence of Theorems 2 to 4. In Section 3 we describe two different ways to calculate the δ_{kt}, establishing i) of Theorem 5, and we derive different relations between these values. Then in Section 4 we prove that there is an optimal solution of the mixing set relaxation (11)-(14) that is feasible and thus optimal in $X^{WW-C(ND)}$ when the q_t are non-increasing.

3 Calculation and Properties of the δ's

The values δ_{kt} defined in (10) for $1 \leq k \leq t \leq n$ can be computed in polynomial time using either a forward or a backward procedure. Here we describe these procedures and then we examine various properties of the δ's.

Forward computation of δ

For fixed k, we compute the δ_{kt} values for all $t \geq k$. Let α be the possible values of s_{k-1} in the optimal solution to (10).

First observe that we can take $\alpha < C_k$ without loss of generality. This holds because, if $s_{k-1} \geq C_k$ in a solution to (10), then at least as good a solution can be constructed by decreasing α by C_k, and setting $y_q = 1$, where $q = \min\{j : k \leq j \leq t, y_j = 0\}$. If all y's were originally equal to 1, then we can simply decrease α by C_k. This modified solution remains feasible because $D_u \leq C_u \leq C_{u+1}$ for all u.

In order to compute δ_{kt} for fixed $s_{k-1} = \alpha$, we need to solve

$$\min \sum_{u=k}^{t} y_u : \sum_{u=k}^{j} C_u y_u \geq D_{kj} - \alpha,\ y_j \in \{0,1\} \text{ for } j = k, \dots, t. \quad (16)$$

Because $C_u \leq C_{u+1}$ for all u, an optimal solution can be found greedily by producing as late as possible, while maintaining feasibility. Formally, an optimal solution $y^{\alpha,k}$ of (16) is obtained by the following procedure.

1. For $j = k, \dots, t$, let $\phi_j^{\alpha,k} = (D_{kj} - \alpha)^+ - \sum_{u=k}^{j-1} C_u y_u^{\alpha,k}$.
2. If $\phi_j^{\alpha,k} > 0$, set $y_j^{\alpha,k} = 1$, and otherwise set $y_j^{\alpha,k} = 0$.

Observe that the computation of $y^{\alpha,k}$ for fixed k and α can be done in a single pass for all $t \geq k$. So far, we have shown that

$$\delta_{kt} = \min_{0 \leq \alpha < C_k} \{\alpha + C_k \sum_{u=k}^{t} y_u^{\alpha,k}\} \quad (17)$$

This procedure to compute all δ values can implemented in polynomial time because at least one set-up is shifted to a later period for each value of α. Thus, for each k, at most $O(n^2)$ values of $\alpha = s_{k-1}$ need to be considered. Given $k \in \{1, \dots, n\}$, the following procedure selects the values of α that one needs to consider.

1. Set $\alpha = 0$.
2. While $\alpha < C_k$, Compute $y_t^{\alpha,k}$ for all $t = k, \dots, n$.
3. Let $\gamma = \min_{t:\phi_t^{\alpha,k} > 0} \phi_t^{\alpha,k} > 0$.
4. Set $\alpha \leftarrow \alpha + \gamma$ and iterate.

Example 2. *Consider again the instance of WW^*-$C(ND)$ represented in Figure 2. Starting from $\alpha = 0$, the computation of $\delta_{26} = 41$ involves the following iterations.*

$\alpha = 0$	$y_2^{0,2} = y_5^{0,2} = y_6^{0,2} = 1$	$\alpha + 20(y_2^{0,2} + \cdots + y_6^{0,2}) = 60$
$\alpha = 1$	$y_3^{1,2} = y_5^{1,2} = 1$	$\alpha + 20(y_2^{1,2} + \cdots + y_6^{1,2}) = 41$
$\alpha = 2$	$y_4^{2,2} = y_5^{2,2} = 1$	$\alpha + 20(y_2^{2,2} + \cdots + y_6^{2,2}) = 42$
$\alpha = 7$	$y_5^{7,2} = y_6^{7,2} = 1$	$\alpha + 20(y_2^{7,2} + \cdots + y_6^{7,2}) = 47$
$\alpha = 22$	*STOP because* $\alpha \geq C_2$,	$\delta_{26} = \min(60, 41, 42, 47) = 41.$

Backward computation of δ

For fixed $t \in \{1, \ldots, n\}$, the backward procedure computes all δ_{kt} variables for $k = t, t-1, \ldots, 1$. It is similar to the approach taken by Chung and Lin [3] to compute of the minimum cost for a regeneration interval.

Given δ_{kt} from (10), define α_{kt} and β_{kt} by expressing $\delta_{kt} = \alpha_{kt} + C_k\beta_{kt}$ with $0 \leq \alpha_{kt} < C_k$.

Before describing the procedure, we need to prove some properties of the α, β and δ values.

Lemma 1. *For k, t with $1 \leq k \leq t \leq n$,*

i. $\beta_{kt} = \min\{\sum_{u=k}^{t} y_u : (s,y) \in X_0^{WW-C(ND)}, s_{k-1} < C_k\}$.

ii. If $\alpha_{kt} > 0$, $\lceil \frac{\delta_{kt}}{C_k} \rceil = \min\{\sum_{u=k}^{t} y_u : (s,y) \in X_0^{WW-C(ND)}, s_{k-1} < \alpha_{kt}\}$.

Proof. i. Let (s,y) be an optimal solution for problem P_{kt} in (10) with $s_{k-1} = \alpha_{kt} < C_k$ and $y_{kt} = \beta_{kt}$. Such a solution always exists, as we already observed in the discussion of the forward procedure. This solution defines a feasible solution of the problem $\min\{\sum_{u=k}^{t} y_u : (s,y) \in X_0^{WW-C(ND)}, s_{k-1} < C_k\}$. If this solution is not optimal for the latter problem, then there exists a solution $(s^*, y^*) \in X_0^{WW-C(ND)}$ with $s^*_{k-1} < C_k$ and $y^*_{kt} \leq \beta_{kt} - 1$, but then $s^*_{k-1} + C_k y^*_{kt} < C_k + C_k(\beta_{kt} - 1) = C_k\beta_{kt} \leq \delta_{kt}$ contradicting the definition of δ_{kt}.
ii. As $\delta_{kt} = \alpha_{kt} + C_k\beta_{kt}$, there is no feasible solution $(s,y) \in X_0^{WW-C(ND)}$ with $s_{k-1} < \alpha_{kt}$ and $y_{kt} \leq \beta_{kt}$.
Therefore, $\min\{\sum_{u=k}^{t} y_u : (s,y) \in X_0^{WW-C(ND)}, s_{k-1} < \alpha_{kt}\} \geq \beta_{kt} + 1 = \lceil \frac{\delta_{kt}}{C_k} \rceil$, where the last equality follows from $\alpha_{kt} > 0$.
It remains to show that there is a solution $(s,y) \in X_0^{WW-C(ND)}$ with $s_{k-1} < \alpha_{kt}$ and $y_{kt} = \beta_{kt} + 1$. Let (s,y) be an optimal solution for problem P_{kt} in (10) with $s_{k-1} = \alpha_{kt} < C_k$ and $y_{kt} = \beta_{kt}$. Modify this solution by setting $s_{k-1} = 0$, and fixing y_q to 1, where $q = \min[u \in \{k, \ldots, t\} : y_u = 0]$. Note that q is well defined, because $\alpha_{kt} > 0$ implies that there is at least one of the y variables equal to 0. This modified solution remains feasible because $D_u \leq C_u \leq C_{u+1}$ for all u, and satisfies $s_{k-1} = 0 < \alpha_{kt}$ and $y_{kt} = \beta_{kt} + 1$. $\square$

The next proposition provides the main properties of the δ values required to construct the backward procedure.

Proposition 2. *Consider k, t with $1 \leq k \leq t \leq n$.*

i. $\delta_{tt} = D_t$, $\beta_{tt} = \lfloor \delta_{tt}/C_t \rfloor$ *and* $\alpha_{tt} = \delta_{tt} - C_t\beta_{tt}$.

ii. If $k < t$ and $\alpha_{k+1,t} \geq C_k$, then $\delta_{kt} = D_k + C_k(1 + \beta_{k+1,t})$

iii. If $k < t$ and $\alpha_{k+1,t} < C_k$, then $\delta_{kt} = D_k + \alpha_{k+1,t} + C_k\beta_{k+1,t}$.

Proof. Consider problem P_{kt} in (10) defining the value of δ_{kt}. There is always a stock minimal solution (s, y) to (10), i.e., such that

$$s_{k-1} = \max_{t=k,\dots,n} \left[D_{k,t} - \sum_{u=k}^{t} C_u y_u \right]^+ \quad \text{for } k = 1, \dots, n \ ,$$

that is optimal for problem P_{kt}. For such a solution, $D_k - C_k y_k + s_k \leq s_{k-1} \leq D_k + s_k$ for all k, see ii) of Proposition 1.
If (s^*, y^*) and (s, y) are two optimal solutions to P_{kt}, (s^*, y^*) dominates (s, y) lexicographically if there exists $t \in \{1, \dots, n\}$ such that $y^*_u = y_u$ for $1 \leq u \leq t-1$ and $0 = y^*_t < y_t = 1$. A lexico-min solution to P_{kt} is a minimal (optimal) solution that is not lexicographically dominated by any other optimal solution. There always exists a lexico-min solution. In such a solution, production occurs as late as possible, and, in particular, for any u with $k \leq u \leq t$, $s_{u-1} \geq D_u$ implies $y_u = 0$. This holds because $D_u \leq C_u \leq C_{u+1}$, and the fixed costs are positive and constant in the objective function of P_{kt} for $u = k, \dots, t$. Therefore, if (s, y) is such that $s_{u-1} \geq D_u$ and $y_u = 1$, then a lexicographically better solution (s^*, y^*) is obtained by setting $y^*_u = 0$ and, if $\{j : u < j \leq n, y_j = 0\} \neq \emptyset$, then $y^*_q = 1$ with $q = \min[j : u < j \leq n, y_j = 0]$.
Finally, as we already observed in the forward procedure, there always exists an optimal solution (s, y) to P_{kt} with $s_{k-1} < C_k$ for all k.

i. The result is trivial because $D_t \leq C_t$ implies that an optimal solution to P_{tt} is $s_{t-1} = D_t$ and $y_t = 0$, which implies $\delta_{tt} = D_t$. If $D_t < C_t$, then $\beta_{tt} = 0$, $\alpha_{tt} = D_t$. If $D_t = C_t$, then $\beta_{tt} = 1$, $\alpha_{tt} = 0$ (in this case, another optimal solution is $s_{t-1} = 0$ and $y_t = 1$).

ii. The solution (s, y) with $s_{k-1} = D_k$, $y_k = 0$, $s_k = 0$, $y_{k+1,t} = \beta_{k+1,t} + 1$ as constructed in the proof of Lemma 1 is feasible for P_{kt} and has cost $D_k + C_k(\beta_{k+1,t} + 1)$. This proves that $\delta_{kt} \leq D_k + C_k(\beta_{k+1,t} + 1)$.
We prove that this last inequality holds at equality by proving that any lexico-min and stock minimal solution (s, y) to P_{kt} with $s_{k-1} < C_k$ has cost at least equal to $D_k + C_k(\beta_{k+1,t} + 1)$. Let (s, y) be such a solution.

1. If $y_k = 0$ then $s_{k-1} \geq D_k$ and $s_k = s_{k-1} - D_k < C_k \leq \alpha_{k+1,t}$. Therefore, by Lemma 1, $y_{k+1,t} \geq \beta_{k+1,t} + 1$. Such a solution has cost in P_{kt} at least equal to $s_{k-1} + C_k(y_k + y_{k+1,t}) \geq D_k + C_k(\beta_{k+1,t} + 1)$.

2. If $y_k = 1$, then $s_{k-1} < D_k$ and $s_k \leq s_{k-1} + C_k - D_k < C_k \leq \alpha_{k+1,t}$. Therefore, by Lemma 1, $y_{k+1,t} \geq \beta_{k+1,t} + 1$. Such a solution has cost in P_{kt} at least equal to $s_{k-1} + C_k(y_k + y_{k+1,t}) \geq 0 + C_k(1 + \beta_{k+1,t} + 1) \geq D_k + C_k(\beta_{k+1,t} + 1)$.

iii. The solution (s, y) with $s_{k-1} = D_k + \alpha_{k+1,t}$, $y_k = 0$, $s_k = \alpha_{k+1,t}$, $y_{k+1,t} = \beta_{k+1,t}$ (as obtained from problem $P_{k+1,t}$) is feasible for P_{kt} and has cost $D_k + \alpha_{k+1,t} + C_k\beta_{k+1,t}$. This proves that $\delta_{kt} \leq D_k + \alpha_{k+1,t} + C_k\beta_{k+1,t}$. Note that if $D_k + \alpha_{k+1,t} \geq C_k$, another equivalent optimal solution of P_{kt} is obtained by taking $s_{k-1} = D_k + \alpha_{k+1,t} - C_k$ and $y_k = 1$.
We prove that $\delta_{kt} = D_k + \alpha_{k+1,t} + C_k(\beta_{k+1,t})$ by showing that any lexico-min and stock minimal feasible solution (s, y) to P_{kt} with $s_{k-1} < C_k$ has cost at least equal to $D_k + \alpha_{k+1,t} + C_k(\beta_{k+1,t})$. Let (s, y) be such a solution.

1. If $y_k = 0$, then $s_{k-1} \geq D_k$ and $s_k = s_{k-1} - D_k < C_k - D_k$.

   (a) If $s_k < \alpha_{k+1,t}$, then by Lemma 1 $y_{k+1,t} \geq \beta_{k+1,t} + 1$, and (s, y) has cost in P_{kt} at least equal to $s_{k-1} + C_k(y_k + y_{k+1,t}) \geq s_{k-1} + C_k + C_k\beta_{k+1,t} > D_k + \alpha_{k+1,t} + C_k\beta_{k+1,t}$.

   (b) If $s_k \geq \alpha_{k+1,t}$, then $s_k < C_k - D_k \leq C_k \leq C_{k+1}$ and by Lemma 1 $y_{k+1,t} \geq \beta_{k+1,t}$. Therefore (s, y) has cost in P_{kt} at least equal to $s_{k-1} + C_k(y_k + y_{k+1,t}) \geq D_k + s_k + C_k(0 + \beta_{k+1,t}) > D_k + \alpha_{k+1,t} + C_k\beta_{k+1,t}$.

2. If $y_k = 1$, then $0 \leq s_{k-1} < D_k$ and $s_k \leq s_{k-1} + C_k - D_k < C_k$.

   (a) If $s_k < \alpha_{k+1,t} < C_k$, then by Lemma 1 $y_{k+1,t} \geq \beta_{k+1,t} + 1$, and (s, y) has cost in P_{kt} at least equal to $s_{k-1} + C_k(y_k + y_{k+1,t}) \geq s_{k-1} + C_k(1 + \beta_{k+1,t} + 1) = s_{k-1} + 2C_k + C_k(\beta_{k+1,t}) > 0 + D_k + \alpha_{k+1,t} + C_k\beta_{k+1,t}$.

   (b) If $s_k \geq \alpha_{k+1,t}$, then $s_k < C_k \leq C_{k+1}$ and by Lemma 1 $y_{k+1,t} \geq \beta_{k+1,t}$. Therefore (s, y) has cost in P_{kt} at least equal to $s_{k-1} + C_k(y_k + y_{k+1,t}) \geq s_k - C_k + D_k + C_k(1 + \beta_{k+1,t}) \geq \alpha_{k+1,t} + D_k + C_k\beta_{k+1,t}$.

□

The backward procedure based on Proposition 2 works as follows, for all t with $1 \leq t \leq n$.

1. $\delta_{tt} = D_t$, $\beta_{tt} = \lfloor \delta_{tt}/C_t \rfloor$, $\alpha_{tt} = \delta_{tt} - C_t\beta_{tt}$

2. For $k = t-1, t-2, \ldots, 1$,

(a) If $\alpha_{k+1,t} \geq C_k$, then $\delta_{kt} = D_k + C_k(1 + \beta_{k+1,t})$
(b) If $\alpha_{k+1,t} < C_k$, then $\delta_{kt} = D_k + \alpha_{k+1,t} + C_k(\beta_{k+1,t})$
(c) $\beta_{kt} = \lfloor \delta_{kt}/C_k \rfloor$, $\alpha_{kt} = \delta_{kt} - C_k\beta_{kt}$.

This procedure computes all δ values in $O(n^2)$.

Example 3. *Consider again the same instance of WW^*-$C(ND)$ represented in Figure 2. We illustrate the backward computation of $\delta_{26} = 41$.*

$\delta_{66} = 25$	$\beta_{66} = 0$	$\alpha_{66} = 25 < C_5$
$\delta_{56} = 40 + 25 + 50 \times 0 = 65$	$\beta_{56} = 1$	$\alpha_{56} = 15 < C_4$
$\delta_{46} = 5 + 15 + 40 \times 1 = 60$	$\beta_{46} = 1$	$\alpha_{46} = 20 < C_3$
$\delta_{36} = 1 + 20 + 30 \times 1 = 51$	$\beta_{36} = 1$	$\alpha_{36} = 21 \geq C_2$
$\delta_{26} = 1 + 20 \times (1 + 1) = 41$	$\beta_{26} = 2$	$\alpha_{26} = 1 < C_1$.

We will need some additional properties of the δ values.

Lemma 2. *For any (k, p, t) such that $1 \leq k < p \leq t \leq n$,*

i. $\delta_{kt} \leq \delta_{k,p-1} + \lceil \delta_{pt}/C_p \rceil C_k$

ii. $\delta_{kt} \geq \delta_{k,p-1} + \lfloor \delta_{pt}/C_p \rfloor C_k$

iii. If $\alpha_{pt} \geq C_{p-1}$, *then* $\delta_{kt} = \delta_{k,p-1} + \lceil \delta_{pt}/C_p \rceil C_k$.

Proof. i. By Lemma 1, the solution (s, y) such that $s_{k-1} = \alpha_{k,p-1}$, $y_{k,p-1} = \beta_{k,p-1}$, $s_{p-1} = 0$, $y_{pt} = \lceil \delta_{pt}/C_p \rceil$ is feasible for problem P_{kt}. Its objective value in P_{kt} is $s_{k-1} + C_k y_{kt} = \alpha_{k,p-1} + C_k y_{k,p-1} + C_k y_{pt} = \delta_{k,p-1} + C_k \lceil \delta_{pt}/C_p \rceil C_k$ providing an upper bound on the optimal value δ_{kt}.

ii. We derive a lower bound on the cost of any optimal solution of P_{kt}. Consider a lexico-min optimal solution (s, y) to P_{kt} (we know that there exists such an optimal solution for P_{kt}), i.e. $\delta_{kt} = s_{k-1} + C_k y_{kt}$. Using the same argument as in the discussion of the forward procedure, we may assume that $0 \leq s_{p-1} < C_p$. Then, by Lemma 1, $s_{p-1} < C_p$ implies $y_{pt} \geq \beta_{pt} = \lfloor \delta_{pt}/C_p \rfloor$. As $s_{p-1} \geq 0$, one must have $s_{k-1} + C_k y_{k,p-1} \geq \delta_{k,p-1}$. The claim follows because $\delta_{kt} = s_{k-1} + C_k y_{k,p-1} + C_k y_{pt} \geq \delta_{k,p-1} + C_k \lfloor \delta_{pt}/C_p \rfloor$.

iii. This follows directly from Proposition 2, and from the backward procedure to compute δ_{tl}. If $\alpha_{pt} \geq C_{p-1}$, then $\delta_{p-1,t} = D_{p-1} + C_{p-1}(1 + \beta_{pt}) = \delta_{p-1,p-1} + C_{p-1}(\lceil \delta_{pt}/C_p \rceil)$, and therefore $\alpha_{ut} = \alpha_{u,p-1}$ and $\beta_{ut} = \beta_{u,p-1} + \beta_{pt} + 1$ for all $u \leq p - 1$. □

4 The Mixing Relaxation Solves WW^*-$C(ND)$

We are now ready to prove Theorem 4.

Proof. We have established that (11)-(14) is a relaxation of $WW - C(ND)$, so it suffices to show that there exists an optimal solution to (11)-(14) which is feasible for LS-$C(ND)$.

Consider an optimal solution (s, y) of (11)-(14) which is stock minimal in (12)-(14), i.e. such that $s_{j-1} = \max_{t \geq j}[\delta_{jt} - C_j y_{jt}]^+$. Such an optimal solution always exists because $h_u \geq 0$ for all u. We decompose this solution into regeneration intervals, and we consider each regeneration interval $[k, l]$ where $s_{k-1} = s_l = 0$ and $s_t > 0$ for $k \leq t < l$. We prove the Theorem via a series of Claims.

Claim 1. *If $[k, l]$ is a regeneration interval of a stock minimal optimal solution (s, y) of (11)-(14), then*
i. $y_{kj} \geq \lceil \delta_{kj}/C_k \rceil$ *for* $j = k, \ldots, l$,
ii. $y_{l+1,j} \geq \lceil \delta_{l+1,j}/C_{l+1} \rceil$ *for* $j = l + 1, \ldots, n$,
iii. $y_{jl} \leq \beta_{jl} = \lfloor \delta_{jl}/C_j \rfloor$ *for* $j = k + 1, \ldots, l$, *and*
iv. $s_{j-1} = \delta_{jl} - C_j y_{jl}$ *for* $j = k + 1, \ldots, l$.

Proof of Claim 1. i. As $[k, l]$ is a regeneration interval of (s, y), $s_{k-1} = 0$. As (s, y) satisfies (12), me must have $C_k y_{kt} \geq \delta_{kt}$ for $t = k, \ldots, l$. The claim follows from the integrality of y.

ii. Similarly, as $[k, l]$ is a regeneration interval of (s, y), $s_l = 0$. Therefore, $C_{l+1} y_{l+1,j} \geq \delta_{l+1,j}$ for $j = l + 1, \ldots, n$, and the claim follows.

iii. and iv. Note that there is nothing to prove, unless $k < l$. We have that $s_{j-1} = \max_{t \geq j}[\delta_{jt} - C_j y_{jt}] > 0$ for $j = k + 1, \ldots, l$. First we show that $s_{j-1} = \max_{t: j \leq t \leq l}[\delta_{jt} - C_j y_{jt}]$. Consider some period $p > l$.

$$\begin{aligned}
\delta_{jp} - C_j y_{jp} &\leq \delta_{jl} + C_j \lceil \frac{\delta_{l+1,p}}{C_{l+1}} \rceil - C_j y_{jl} - C_j y_{l+1,p} \quad \text{(by Lemma 2 i.)} \\
&\leq \delta_{jl} - C_j y_{jl} \quad \text{(by Claim 1 ii.)}.
\end{aligned}$$

Now define $H(j)$ to be true if iii) and iv) hold for all t such that $j \leq t \leq l$. First consider $H(l)$. As $\delta_{ll} = D_l$, we have that $s_{l-1} = d_l - C_l y_l > 0$. If $y_l = 1$, then $s_{l-1} \leq 0$, a contradiction. Thus $y_l = 0$ and the claim holds for $j = l$, i.e., $H(l)$ is true.
Now suppose that $H(j + 1)$ is true for some $j + 1 \leq l$, $j \geq k + 1$. Thus

$y_{tl} \le \beta_{tl}$ for $j+1 \le t \le l$ and $s_j = \delta_{j+1,l} - C_{j+1}y_{j+1,l}$. Consider any period p with $j < p \le l$. Then

$$\begin{aligned} \delta_{jl} - C_j y_{jl} &\ge \delta_{j,p-1} + C_j \lfloor \frac{\delta_{pl}}{C_p} \rfloor - C_j y_{jl} \quad \text{(by Lemma 2 ii.)} \\ &= \delta_{j,p-1} - C_j y_{j,p-1} + C_j(\lfloor \frac{\delta_{pl}}{C_p} \rfloor - y_{pl}) \\ &\ge \delta_{j,p-1} - C_j y_{j,p-1} \quad (\text{as } y_{pl} \le \beta_{pl}). \end{aligned}$$

Thus $s_{j-1} = \delta_{jl} - C_j y_{jl}$. Finally as $s_{j-1} > 0$, we must have $y_{jl} \le \lfloor \frac{\delta_{jl}}{C_j} \rfloor = \beta_{jl}$, and $H(j)$ is true. Repeating recursively this proof for $j = l-1, l-2, \ldots, k+1$ proves the claim. □

Note that the above proof shows that for any feasible solution (s, y) to (11)-(14) with $s_l = 0$ and $y_{jl} \le \beta_{jl}$ for $j = k+1, \ldots, l$, then $s_{j-1} = \delta_{jl} - C_j y_{jl}$, for $j = k+1, \ldots, l$.

Recall that a lexico-min solution (s, y) to (11)-(14) is an optimal solution that is not lexicographically dominated by any other optimal solution. That is, if $y_t = 1$ for some t, there does not exist another optimal solution (s^*, y^*) with $y_u = y_u^*$ for $u < t$, and $y_t^* = 0$.

Claim 2. *If (s, y) a stock minimal lexico-min solution to (11)-(14), and $[k, l]$ is a regeneration interval of (s, y), then $y_{jl} = \beta_{jl}$ and $s_{j-1} = \alpha_{jl}$ for $j = k+1, \ldots, l$.*

Proof of Claim 2. By Claim 1, $y_{jl} = \beta_{jl}$ implies $s_{j-1} = \alpha_{jl}$ for $j = k+1, \ldots, l$. Therefore we only need to prove that $y_{jl} = \beta_{jl}$ for $j = k+1, \ldots, l$. Note also that there is nothing to prove unless $k < l$.

Let (s, y) be a stock minimal lexico-min solution to (11)-(14), and $[k, l]$ be a regeneration interval of (s, y). By contradiction, assume that $y_{pl} < \beta_{pl}$ for some $p \ge k+1$, and $y_{tl} = \beta_{tl}$ for $p+1 \le t \le l$. We distinguish the two cases: $y_p = 0$ and $y_p = 1$.

Case $y_p = 0$. Let $q = \max[j : k \le j < p, y_j = 1]$. Such a q always exists because $y_{pl} < \beta_{pl} \le \beta_{kl} \le \lceil \delta_{kl}/C_k \rceil \le y_{kl}$ and therefore $y_{k,p-1} > 0$. Also $y_{jl} < \beta_{jl}$ for $q < j \le p$ because $y_{jl} = y_{pl} < \beta_{pl} \le \beta_{jl}$.

Now we construct the solution (s^*, y^*) as $y^* = y - e_q + e_p$, where e_j is the unit vector with a 1 in position j, and $s_{j-1}^* = \max_{t \ge j}[\delta_{jt} - C_j Y_{jt}^*]^+$. This solution is feasible in (11)-(14), and dominates (s, y) lexicographically.

To obtain a contradiction, it remains to show that the cost of (s^*, y^*) is not greater than that of (s, y).

By construction, $y^*_{jl} \leq \beta_{jl}$ for all $k < j \leq l$, and $s^*_l = s_l = 0$. So, by the proof of Claim 1, we still have that $s^*_{j-1} = \delta_{jl} - C_j y^*_{jl}$, for $j = k+1, \ldots, l$. Therefore

$$
\begin{array}{ll}
s^*_{j-1} = s_{j-1} & \text{for } j > l \\
s^*_{j-1} = \delta_{jl} - C_j y^*_{jl} = \delta_{jl} - C_j y_{jl} = \alpha_{jl} = s_{j-1} & \text{for } p < j \leq l \\
s^*_{j-1} = \delta_{jl} - C_j y^*_{jl} = \delta_{jl} - C_j(y_{jl} + 1) < \delta_{jl} - C_j y_{jl} = s_{j-1} & \text{for } q < j \leq p \\
s^*_{j-1} = \delta_{jl} - C_j y^*_{jl} = \delta_{jl} - C_j y_{jl} = s_{j-1} & \text{for } k < j \leq q.
\end{array}
$$

We check now that $s^*_{k-1} = s_{k-1} = 0$ which implies that $s^*_{j-1} = s_{j-1}$ for all $j \leq k$, as shown in the proof of Claim 1. By Lemma 2 *ii.* $\delta_{kl} \geq \delta_{k,j-1} + C_k \beta_{jl}$ for $j = q+1, \ldots, p$, and by Claim 1 $y_{kl} \geq \lceil \delta_{kl}/C_k \rceil$. Therefore $y_{kl} \geq \lceil \delta_{k,j-1}/C_k \rceil + \beta_{jl} > \lceil \delta_{k,j-1}/C_k \rceil + y_{jl}$ implying that $y_{k,j-1} > \lceil \delta_{k,j-1}/C_k \rceil$ and $y^*_{k,j-1} = y_{k,j-1} - 1 \geq \lceil \delta_{k,j-1}/C_k \rceil$ for $j = q+1, \ldots, p$. As $y^*_{k,j} = y_{k,j}$ for $j = k, \ldots, q-1$ and $j = p, \ldots, l$, we have $y^*_{kj} \geq \lceil \delta_{k,j}/C_k \rceil$ for $j = k, \ldots, l$. Together with $s^*_l = 0$, this implies that $s^*_{k-1} = 0$.

So, we have shown that $s^* \leq s$, and the solution (s^*, y^*) has cost not larger than (s, y) because set-up costs are non-increasing and inventory costs are non-negative in (11)-(14).

Case $y_p = 1$. Note that $p < l$ in this case, because $y_l = y_{ll} = 1$ and $y_{ll} < \beta_{ll}$ is impossible. Note also that this case, with $y_{pl} = y_{p+1,l} + 1 = \beta_{p+1,l} + 1 < \beta_{pl}$, can only occur if $\beta_{pl} = \beta_{p+1,l} + 2$, which happens if and only if $D_p = C_p$ and $\alpha_{p+1,l} \geq C_p$.

Let $q = \min[j : p < j \leq l, y_j = 0]$. Such a q always exists because $y_l = y_{ll} = \beta_{ll} = 0$ if $D_l < C_l$, and if $D_l = C_l$ then $y_l = y_{ll} = \beta_{ll} = 1$ implies $k = l$ (i.e., $s_{l-1} = 0$) and there is nothing to prove.

Now we construct the solution (s^*, y^*) as $y^* = y - e_p + e_q$, and $s^*_{j-1} = \max_{t \geq j}[\delta_{jt} - C_j y^*_{jt}]^+$. Again, this solution is feasible in (11)-(14), and dominates (s, y) lexicographically. To obtain a contradiction, it remains to show that $s^* \leq s$, which implies that the cost of (s^*, y^*) is not greater than that of (s, y).

i. For $j > q$, we have $s^*_{j-1} = s_{j-1}$.

ii. By Claim 1 and $y_{ql} = \beta_{ql}$, $s_{q-1} = \max_{t \geq q}[\delta_{qt} - C_q y_{qt}]^+ = \alpha_{ql} < C_q$. Since $y^*_{qt} = y_{qt} + 1$ for all $t \geq q$, $s^*_{q-1} = \max_{t \geq q}[\delta_{qt} - C_q y^*_{qt}]^+ =$

$\max_{t\geq q}[\delta_{qt} - C_q - C_q y_{qt}]^+ = [s_{q-1} - C_q]^+ = 0$. This implies that $s^*_{q-1} \leq s_{q-1}$ and $y^*_{qt} \geq \lceil \delta_{qt}/C_q \rceil$ for all $t \geq q$.

iii. For $j = p+1, \ldots, q-1$ and $t \geq q$, by Lemma 2 *i.*, $\delta_{jt} - C_j y^*_{jt} \leq \delta_{j,q-1} + \lceil \delta_{qt}/C_q \rceil C_j - C_j y^*_{j,q-1} - C_j y^*_{qt} \leq \delta_{j,q-1} - C_j y^*_{j,q-1}$, where the last inequality holds because $y^*_{qt} \geq \lceil \delta_{qt}/C_q \rceil$ for all $t \geq q$.

Therefore, for $j = p+1, \ldots, q-1$, $s^*_{j-1} = \max_{t\geq j}[\delta_{jt} - C_j y^*_{jt}]^+ = \max_{t:j\leq t\leq q-1}[\delta_{jt} - C_j y_{jt}]^+ = 0$, where the last equality holds because for $j \leq t \leq q-1$, $y^*_u = 1$ for all $u = j, \ldots, t$ and $\delta_{jt} \leq (t-j+1)\, C_j$ by Lemma 2 *i.* (because $\delta_{jt} \leq \delta_{j,t-1} + \lceil \delta_{tt}/C_t \rceil C_j \leq \delta_{j,t-2} + \lceil \delta_{t-1,t-1}/C_{t-1} \rceil C_j + \lceil \delta_{tt}/C_t \rceil C_j \leq \ldots \leq \sum_{u=j}^{t} \lceil \delta_{uu}/C_u \rceil C_j = (t-j+1)\, C_j$).

In particular, $s^*_p = 0$ implies $y^*_{p+1,t} \geq \lceil \delta_{p+1,t}/C_{p+1} \rceil$ for all $t \geq p+1$.

iv. For $j = k+1, \ldots, p$, using Lemma 2 *i.* $\delta_{jl} \leq \delta_{jp} + \lceil \delta_{p+1,l}/C_{p+1} \rceil C_j$. Thus $\beta_{jl} \leq \beta_{j,p} + \lceil \delta_{p+1,l}/C_{p+1} \rceil \leq \beta_{j,p} + y^*_{p+1,l}$. As $y^*_{jl} = y_{jl} \leq \beta_{jl}$, $y^*_{j,p} \leq \beta_{j,p}$, for all $j = k+1, \ldots, p$. Together with $s^*_p = 0$, using the same proof as in Claim 1, this implies that $s^*_{j-1} = \delta_{jp} - C_j y^*_{jp}$ for all $j = k+1, \ldots, p$. In fact this shows that $[k, p]$ is a new regeneration interval in (s^*, y^*).

As $\alpha_{p+1,l} \geq C_p > 0$, by Lemma 2 *iii.*, we have $\delta_{jl} = \delta_{jp} + C_j(1+\beta_{p+1,l})$, for all $j = k+1, \ldots, p$. So for $j = k+1, \ldots, p$,

$$\begin{aligned}
s^*_{j-1} &= \delta_{jp} - C_j y^*_{jp} \\
&= \delta_{jl} - C_j(1+\beta_{p+1,l}) - C_j(y_{jp} - 1) \\
&= \delta_{jl} - C_j(1+y_{p+1,l}) - C_j y_{jp} + C_j \\
&= \delta_{jl} - C_j y_{jl} \\
&= s_{j-1}.
\end{aligned}$$

v. Finally to prove that $s^*_{k-1} = 0$, we only need to show that $y^*_{kj} \geq \lceil \delta_{kj}/C_k \rceil$, for $j = p, \ldots, q-1$, because by Claim 1 $y^*_{kj} = y_{kj} \geq \lceil \delta_{kj}/C_k \rceil$ for $j = k, \ldots, p-1$ and $j = q, \ldots, l$.

As $\alpha_{p+1,l} \geq C_p$, we have $\delta_{kl} = \delta_{kp} + C_k(1+\beta_{p+1,l})$ by Lemma 2 *iii.* Therefore $y^*_{kl} = y_{kl} \geq \lceil \delta_{kl}/C_k \rceil \geq \lceil \delta_{kp}/C_k \rceil + 1 + \beta_{p+1,l} = \lceil \delta_{kp}/C_k \rceil + y^*_{p+1,l}$, which implies $y^*_{kp} \geq \lceil \delta_{kp}/C_k \rceil$.

For $j = p+1, \ldots, q-1$, Lemma 2 *i.* implies that $\delta_{kj} \leq \delta_{kp} + \sum_{u=p+1}^{j} \lceil \delta_{uu}/C_u \rceil C_k = \delta_{kp} + (j-p)C_k$. Therefore, $y^*_{kj} = y^*_{kp} + y^*_{p+1,j} \geq \lceil \delta_{kp}/C_k \rceil + (j-p) \geq \lceil \delta_{kj}/C_k \rceil$. □

Claim 3. *If (s, y) a stock minimal lexico-min solution to (11)-(14), and $[k, l]$ is a regeneration interval of (s, y), then $y_{jl} = \beta_{jl}$ and $s_{j-1} = \alpha_{jl} < C_{j-1}$ for $j = k+1, \ldots, l$.*

Proof of Claim 3. By Claim 2, we know that $y_{jl} = \beta_{jl}$ and $s_{j-1} = \alpha_{jl}$ for $j = k+1, \ldots, l$. So, we assume by contradiction that $s_p = \alpha_{p+1,l} \geq C_p$ for some $p \in \{k, \ldots, l-1\}$. Because $\alpha_{p+1,l} \geq C_p$, we must have $y_{pl} = \beta_{pl} > \beta_{p+1,l} = y_{p+1,l}$, and thus $y_p = 1$. The proof by contradiction is identical to the proof of Claim 2 in the case $y_p = 1$. □

To conclude the proof of the main Theorem, it suffices to show that a stock minimal lexico-min solution (s, y) of (11)-(14) is feasible for *LS-C*(ND). Let (s, y) be a stock minimal lexico-min solution of (11)-(14). So, we have to prove that $x_t = s_t + D_t - s_{t-1}$ satisfies $0 \leq x_t \leq C_t y_t$, for all $t \in \{1, \ldots, n\}$.

Let $[k, l]$ be any regeneration interval of (s, y) with $k < l$.

i. As $s_l = 0$ and $s_{l-1} = \alpha_{ll} = \delta_{ll} - C_l \beta_{ll} = D_l - C_l y_l$ by the previous claims, we have $x_l = s_l + D_l - s_{l-1} = C_l y_l \in [0, C_l y_l]$.

ii. Consider any $j \in \{k+1, \ldots, l-1\}$ with $y_j = 0$. As $y_{jl} = \beta_{jl}$, $y_{j+1,l} = \beta_{j+1,l}$ and $y_{jl} = y_{j+1,l}$, we must have $\beta_{jl} = \beta_{j+1,l}$. This implies that $D_j + \alpha_{j+1,l} < C_j$ and $\alpha_{jl} = D_j + \alpha_{j+1,l}$. Therefore $x_j = s_j + D_j - s_{j-1} = \alpha_{j+1,l} + D_j - \alpha_{jl} = 0 \in [0, C_j y_j]$.

iii. Consider any $j \in \{k+1, \ldots, l-1\}$ with $y_j = 1$. As $y_{jl} = \beta_{jl}$, $y_{j+1,l} = \beta_{j+1,l}$ and $y_{jl} = y_{j+1,l} + 1$, we must have $\beta_{jl} = \beta_{j+1,l} + 1$. As $\alpha_{j+1,l} < C_j$, this implies that $D_j + \alpha_{j+1,l} \geq C_j$ and $\alpha_{jl} = D_j + \alpha_{j+1,l} - C_j$. Therefore $x_j = s_j + D_j - s_{j-1} = \alpha_{j+1,l} + D_j - \alpha_{jl} = C_j \in [0, C_j y_j]$.

iv. Finally, $y_{kl} \geq \lceil \delta_{kl}/C_k \rceil$, $y_{k+1,l} = \beta_{k+1,l}$ and therefore $\delta_{kl} = D_k + \alpha_{k+1,l} + C_k \beta_{k+1,l}$. As $s_k = \alpha_{k+1,l} > 0$ in a regeneration interval, we must have $0 < D_k + \alpha_{k+1,l} \leq C_k$ and $y_k = 1$. Therefore $x_k = s_k + D_k - s_{k-1} = \alpha_{k+1,l} + D_k \in [0, C_k y_k]$.

Finally, if $[k, k]$ is a regeneration interval of (s, y), we must have $y_{kk} \geq \lceil \delta_{kk}/C_k \rceil$, that is $y_k = 1$ if $D_k > 0$. Therefore, $x_k = s_k + D_k - s_{k-1} = D_k \in [0, C_k y_k]$.

□

Example 4. *Figure 3 shows an example of a regeneration interval $[1, 6]$ and a stock minimal solution for an instance of (11)-(14). This solution is* *not a lexico-min solution, and does not correspond to a feasible solution of LS-$C(ND)$.*

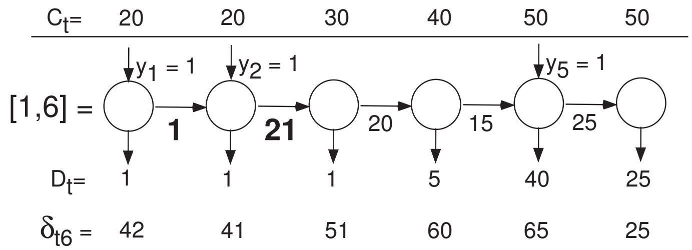

Figure 3: A stock minimal solution, not feasible for LS-$C(ND)$

In this instance, $y_{j6} = \beta_{j6}$ for all $j = 2, \ldots, 6$, but $s_2 = \alpha_{36} \geq C_2$. The proof of Claim 3 shows how to transform this solution to a lexico-dominating solution (here, $p = 2$ and $q = 3$) without increasing the cost. This latter solution is represented in Figure 4. Since it is a lexico-min and stock minimal solution of (11)-(14), it defines a feasible solution of LS-$C(ND)$.

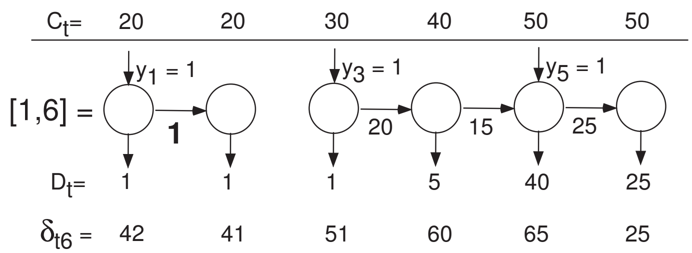

Figure 4: A stock minimal and lexico-min solution, feasible for LS-$C(ND)$

Remark It can be checked that all the reformulation results presented so far remain valid for the case where the integer variables y have arbitrary bounds $y_t \leq v_t$ with $v_t \in \mathbb{Z}^1_+$ or are unbounded $y_t \leq \infty$. In this case preprocessing must again be carried out to ensure that $D_t \leq v_t C_t$. The backward procedure to compute δ is then unchanged, and the proofs can be modified appropriately.

5 Numerical Results for WW^*-$C(ND)$ and LS-C

Here we illustrate the impact of adding the extended formulation for $\text{conv}(X_R^{WW-C(ND)})$ to the initial lot-sizing formulation (1)-(5). Specifically we add the extended formulation of Theorem 3 for each mixing set $X_k^{MIX^*}$ defined in (15).

We first illustrate our reformulation results on an instance with $n = 20$ time periods, $D_t \in [6, 35]$, $h_t \in [0.01, 0.05]$, $y_t \in \mathbb{Z}_+$. From Theorems 4 and 5, this reformulation will solve the problem as an LP, i.e., without any branching, for WW^*-$C(ND)$. The reformulation is also valid and tightens the formulation of other lot-sizing problems. For the following lot-sizing problems, we test the impact of this reformulation on the solution performance using a state-of-the-art mixed integer programming solver.

1. WW-$C(ND)$, where the objective satisfies the WW cost conditions without any assumption on set-up costs,
2. LS-$C(ND)$, where there is no assumption on the objective function coefficients,
3. $Prob$-C, with $Prob$=WW^*, WW or LS, where there is no monotonicity restriction on the capacities, i.e., capacities increase and decrease arbitrarily over time.

To use the reformulation results for the general capacity problems $Prob-C$, we first have to build a valid relaxation $Prob - C(ND)$, in which the capacities are non-decreasing over time. To avoid a very weak relaxation, we build a non-decreasing capacity sequence starting from each period k. Formally, for each k, we define non-decreasing capacities $C_t^{ND_k}$, for $t \geq k$, as $C_k^{ND_k} = C_k$ and $C_t^{ND_k} = \max[C_t, C_{t-1}^{ND_k}]$ for $t > k$. This allows us to compute δ_{kt} values for all $t \geq k$ and define valid mixing set relaxations of the form (15). Note that in contrast to the case of non-decreasing capacities, the computations of δ_{kt} and $\delta_{k+1,t}$, $\delta_{k+2,t}$, ... require different capacities C^{ND_k}, $C^{ND_{k+1}}$, $C^{ND_{k+2}}$, ..., and thus cannot be performed in a single execution of the backward procedure. Therefore, the computation of δ runs in $O(n^3)$ for $Prob - C$.

Table 1 describes the data generation process for the instance solved, where $U([a, b])$ refers to the uniform distribution with values in $[a, b]$, $\nearrow [a, b]$ (resp. $\searrow [a, b]$) refers to a non-decreasing (resp. non-increasing) sequence in $[a, b]$. All data C_t, q_t, p_t are integral. With WW costs, we assume without loss of generality that $p_t = 0$ for all t.

For these lot-sizing instances, we compare the performance of four different formulations using Xpress-MP (on a P-IV running at 1.73 GHz), namely

1. $INIT$: Initial formulation (1)-(5) in the (x, s, y) space,

Problem	C_t	q_t	p_t
$WW^* - C(ND)$	$\nearrow [5, 25]$	$\searrow [2, 21]$	0
$WW - C(ND)$	$\nearrow [5, 25]$	$U([16, 20])$	0
$LS - C(ND)$	$\nearrow [5, 25]$	$U([16, 20])$	$U([0.01, 0.05])$
$WW^* - C$	$U([16, 25])$	$\searrow [2, 21]$	0
$WW - C$	$U([16, 25])$	$U([16, 20])$	0
$LS - C$	$U([16, 25])$	$U([16, 20])$	$U([0.01, 0.05])$

Table 1: Data generation process for the lot-sizing instances

2. $XPRESS$: Initial formulation, plus default Xpress cuts,
3. $MIXING$: Initial, plus extended reformulation in the (x, s, y, μ, σ) space of the mixing set relaxations,
4. $MIX - XPR$: Mixing, plus default Xpress cuts.

With $n = 20$ time periods, the initial formulation involves 40 constraints, 61 variables, and 20 integer variables, and the mixing reformulation involves 290 constraints, 311 variables and 40 integer variables y and μ. These small problems are all solved in 0 or 1 second with all formulations tested. So, we do not compare the running times, but the number of branch-and-bound nodes needed to solve the problems, and the integrality gap obtained at the root node of the enumeration tree, where $Gap = 100 \times$ (Optimal value $-$ Root LP value)/ Optimal value (%). The results are given in Table 2.

Formulation	$INIT$		$XPRESS$		$MIXING$		$MIX - XPR$	
Problem	Gap	nodes	Gap	nodes	Gap	nodes	Gap	nodes
$WW^* - C(ND)$	5.62	1868	2.60	1495	0	1	0	1
$WW - C(ND)$	2.52	823	1.54	666	0.23	46	0	1
$LS - C(ND)$	4.26	9724	2.60	7747	0.21	54	0.14	14
$WW^* - C$	6.53	5653	4.30	1111	1.31	88	1.18	104
$WW - C$	2.84	824	1.76	297	0.42	7	0.05	3
$LS - C$	3.63	4582	3.19	1998	1.30	751	1.14	422

Table 2: Numerical result for instances with $n = 20$.

The results in Table 2 show clearly that the reformulation is effective for all instances. For problems that are not solved at the root node, the best formulation is $MIX-XPR$.

To analyze the impact of the reformulations on the running time, we solved a larger instance of $LS-C$ with $n=100$ time periods, with $D_t \in U([16,35])$, $C_t \in U([16,25])$, $q_t \in U([16,20])$, $h_t \in U([0.01,0.05])$, $p_t \in U([0.01,0.05])$. The initial formulation involves 200 constraints, 301 variables and 100 integer variables, and the mixing reformulation involves 5450 constraints, 5551 variables and 200 integer variables.

As the size of the mixing set reformulation becomes quite large as n is increased, we have also tested a partial or reduced reformulation defined by only including in the mixing sets the constraints $s_{k-1}+C_k \sum_{u=k}^{t} y_u \geq \delta_{kt}$ for which $t-k \leq 10$. This reduces the size of the extended reformulation from $O(n^2)$ to $O(10\,n)$ variables and constraints at the cost of a slightly weaker reformulation. The corresponding formulations are called $MIXING-RED$, and $MIX-RED-XPR$ when Xpress cuts are added. For this instance, the formulation $MIXING-RED$ involves 1445 constraints and 1546 variables. The results with these formulations for an instance with $n=100$ are shown in Table 3.

Formulation	rootLP gap (%)	final gap (%)	nodes	time (secs)
$INIT$	1.14	0.34	$>1\,540\,000$	$>1\,200$
$XPRESS$	0.53	0	389 778	466
$MIXING$	0.20	0	15 506	194
$MIX-XPR$	0.15	0	6 518	192
$MIXING-RED$	0.29	0	14 929	38
$MIX-RED-XPR$	0.18	0	8 126	34

Table 3: Numerical results for an instance of LS-C with $n=100$.

The initial formulation cannot solve the problem to optimality in 1200 seconds. The final gap after 1200 seconds is still 0.34 %. The other formulations solve the instance in less than 1200 seconds. Although the integrality gap at the root node is larger with the reduced reformulation, this has little or no effect on the total number of nodes needed to solve the instance to optimality. Since each LP is smaller, the total running time with reduced reformulations is substantially lower than with the complete reformulation. The best reformulation $MIX-RED-XPR$ is able to solve the instance 6 times faster than the complete mixing reformulation, and 14 times faster than default Xpress.

6 Conclusion

We have described a compact LP formulation for solving the polynomial problem WW^*-$C(ND)$, based on a mixing set relaxation, and its known reformulations. We have also shown that this reformulation approach can be used to build improved formulations for the $NP-hard$ capacitated lot-sizing problem LS-C.

As a first extension, it is possible to derive tighter mixing set relaxations for problem LS-C. For instance, if the capacities are non-decreasing in all but one period, i.e. $C_t \leq C_{t+1}$, for all $t \neq q$, and $C_q > C_{q+1}$. Then it is easy to show that the problem P_{kt} defined in (10) can still be solved in polynomial time by using a combination of the forward and backward procedures proposed in this paper to compute δ. Therefore, a mixed integer set relaxation can be built efficiently, and its extended reformulation used to improve the formulation of LS-C. Such extensions could be investigated further.

More generally mixing sets have been used to model a wide variety of simple mixed integer sets and constant capacity lot-sizing sets, for example: Conforti et al. [4] study the mixing set with flows; Miller and Wolsey [11], and Van Vyve [17] study the continuous mixing set, whose reformulations have been used in Van Vyve [18] to propose extended formulations for lot-sizing problems with backlogging and constant capacity; Di Summa and Wolsey [6] have used mixing sets to model lot-sizing problems on a tree, leading to improved formulations for the stochastic lot-sizing problem with a tree of scenarios; Conforti and Wolsey [5] and Van Vyve [16] have studied an extension of the mixing set with two divisible capacities, that could lead to improved formulations for variants of LS-C, and recently de Farias and Zhao [7] have studied mixing sets with any number of divisible capacities.

The natural question to ask is whether the approach of this paper can be extended to some of these models, so as to provide effective formulations for variants with arbitrarily varying capacities.

More generally the link between such extensions of mixing sets and formulations of various lot-sizing problems still seems to merit further investigation.

References

[1] Atamtürk, A., Munoz, J.: A study of the lot-sizing polytope. Mathematical Programming **99**, 443–466 (2004)

[2] Bitran, G., Yanasse, H.: Computational complexity of the capacitated lot size problem. Management Science **28**, 1174–1186 (1982)

[3] Chung, C., Lin, M.: An O(T^2) algorithm for the $NI/G/NI/ND$ capacitated single item lot size problem. Management Science **34**, 420–426 (1988)

[4] Conforti, M., Di Summa, M., Wolsey, L.: The mixing set with flows. SIAM Journal of Discrete Mathematics **29**, 396–407 (2007)

[5] Conforti, M., Wolsey, L.: Compact formulations as a union of polyhedra. Mathematical Programming **doi 10.1007/s10107-007-0101-0** (2007)

[6] Di Summa, M., Wolsey, L.: Lot-sizing on a tree. Operations Research Letters **doi:10.1016/j.orl.2007.04.007** (2007)

[7] de Farias Jr., I., Zhao, M.: The mixing-mir set with divisible capacities. Working paper, University of Buffalo, U.S.A (2006)

[8] Florian, M., Klein, M.: Deterministic production planning with concave costs and capacity constraints. Management Science **18**, 12–20 (1971)

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

[10] van Hoesel, C., Wagelmans, A.: An $O(T^3)$ algorithm for the economic lot-sizing problem with constant capacities. Management Science **42**, 142–150 (1996)

[11] Miller, A., Wolsey, L.: Tight formulations for some simple MIPs and convex objective IPs. Mathematical Programming B **98**, 73–88 (2003)

[12] Pochet, Y.: Valid inequalities and separation for capacitated economic lot-sizing. Operations Research Letters **7**, 109–116 (1988)

[13] Pochet, Y., Wolsey, L.: Lot-sizing with constant batches: Formulation and valid inequalities. Mathematics of Operations Research **18**, 767–785 (1993)

[14] Pochet, Y., Wolsey, L.: Polyhedra for lot-sizing with Wagner-Whitin costs. Mathematical Programming **67**, 297–324 (1994)

[15] Pochet, Y., Wolsey, L.: Production Planning by Mixed Integer Programming. Springer Series in Operations Research and Financial Engineering, New York (2006)

[16] Van Vyve, M.: Algorithms for single item constant capacity lotsizing problems. Discussion Paper DP03/07, CORE, Universite catholique de Louvain, Louvain-la-Neuve (2003)

[17] Van Vyve, M.: The continuous mixing polyhedron. Mathematics of Operations Research **30**, 441–452 (2005)

[18] Van Vyve, M.: Linear programming extended formulations for the single-item lot-sizing problem with backlogging and constant capacity. Mathematical Programming **108**, 53–78 (2006)