RESEARCH INTERNSHIP REPORT

April 2023 - July 2023

Agathe L'Hermite

Supervised by Marilène Cherkesly and Matthieu Gruson

CONTENTS

1
SUMMARY

We introduce, model and solve the one-warehouse multi-retailer problem with production constraints (OWMR-PC), which is an extension of the one-warehouse multi-retailer problem (OWMR). In the OWMR-PC, we consider one warehouse that produces one type of item over a discrete and finite planning horizon. The items are transported to retailers which have to satisfy a known customer-demand. We explore different types of production constraints: 1) production is only permitted at a predefined set of periods, 2) limiting the number of production periods either through a maximum number of production periods or through a minimal or maximal number of periods between production. Those constraints mimic a situation where the length of time periods is different between the warehouse and the retailers. The objective consists of finding a solution which minimizes the operational costs, comprising a fixed production and order cost and an inventory holding cost, which respects the predefined set of constraints including the production constraints. We propose different ways to adapt the state-of-the-art formulations for the OWMR to the OWMR-PC. We conduct extensive computational experiments to show the limitations of each formulation and we derive appropriate managerial insights related to considering production constraints.

Nous introduisons, modélisons et résolvons le problème *one-warehouse multi-retailer* avec des contraintes de production (OWMR-PC), qui est une extension du one-warehouse multi-retailer (OWMR). Dans le OWMR-PC, nous considérons un entrepôt qui produit un type de produit pendant un intervalle de plannification discret et fini. Les produits sont transportés à des détaillants qui doivent satisfaire une demande connue. Nous étudions différents types de contraintes de production : 1) les commandes sont autorisées durant un ensemble prédéfini de périodes, 2) le nombre de périodes de production est limité avec un nombre maximal de périodes de production ou un nombre minimal ou maximal de périodes entre deux périodes de production. Ces contraintes représentent une situation où la durée des périodes de temps est différentes entre l'entrepôt et les détaillants. L'objectif consiste à trouver une solution qui minimise les coûts opérationnels, comprenant les coûts fixes de production et de commande ainsi que des coûts d'inventaire, et qui respecte les contraintes prédéfinies dont les contraintes de production. Nous proposons différentes manières pour adapter les formulations de l'état de l'art pour le OMWR au OWMR-PC. Nous effectuons des expériences numériques pour montrer les limitations de chaque formulation et nous donnons des perspectives managériales en considération des contraintes de production.

2
INTRODUCTION

Supply chain problems have become increasingly important in influencing production and stock management across different facilities. Major companies require effective models to make decisions regarding production schedules and the distribution of their products. In this context, we study a two-level lot-sizing problem with one warehouse and multiple retailers. We consider a general manufacturing company that operates with a single warehouse and multiple retailers. These retailers face a dynamic and known demand over a discrete and finite time horizon T. The objective of the problem is to determine the optimal timing and flow of goods between the warehouse and the retailers while minimizing the operational costs (production, replenishment, holding costs). Essentially, we face a decision problem where one has to decide the ideal quantities to produce, send and keep in stock for each facility and each period. This two-level lot-sizing problem has practical applications, such as companies like IKEA or supermarkets that utilize warehouses as part of their supply chain and sell products in retailers. Many applications of this problem are known in various industries.

Figure 1 represents the OWMR for two consecutive periods and the potential pathways for the product flows.

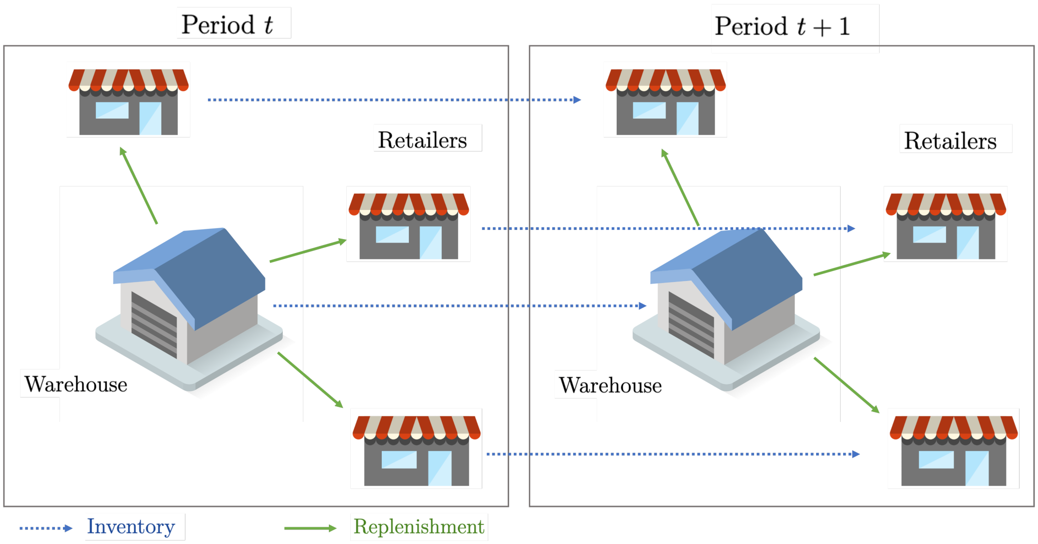

Figure 1: Graphical representation of the problem considered

In our analysis, we consider that the warehouse *produces* an item, whereas retailers *order* an item. We explore different types of constraints related to production at the warehouse, which hold significant practical importance. In fact, production requires additional workload for warehouses. Some days of the week or periods of the year may be impractical for production due to various events or restrictions. For example, it can be more complex to produce items over on Saturdays and Sundays, even though the retailers could pass orders on these days.

Similarly, producing after some holidays (e.g., Chinese new year) might be more complex in terms of workload. Factories also have to manage workers' schedules who need breaks between shifts. Therefore, considering such constraints is important to understand the impacts both in terms of methodological developments as well as practical implications. To the best of our knowledge, we are the first to study the one-warehouse multi-retailer problem (OWMR) with additional production constraints.

We will first work on the different formulations of the problem. Indeed, there exist several formulations for the OWMR. We are going to consider various types of constraints. The first type considers a problem where the set of forbidden periods is pre-defined. This can be the case for public holidays for example or for weekends. We also consider limitations on the production periods. For instance, we consider instances where there is a maximum number of orders that can be made. This situation may arise, particularly in industries with high production costs or those seeking to achieve production balance. Lastly, we consider limitations concerning the number of periods between two consecutive production periods. This can be the case if the time scale for production is long or for industries with perishable goods. For the different production constraints, we compute computational experiments and derive managerial insights as well as theoretical considerations.

3 LITERATURE REVIEW

In this section, we study first the Uncapacitated Lot Sizing Problem (ULS). We give a brief summary of its properties. The ULS is a brick of the OWMR with properties similar to the OWMR. Indeed, in ULS, we only study one facility which can be the warehouse or the retailer. We then study the OWMR and present the different formulations that are known. Lastly, we present the results of computational tests over the different formulations of the OWMR.

3.1 The uncapacitated lot sizing problem

The lot sizing problem has been extensively studied in the literature. We focus here on a simple case of the lot-sizing problem: the Uncapacitated Lot Sizing Problem (ULS). We want to determine planning over a finite planning horizon of $|T|$ periods, $1 \leq 1 \leq |T|$. We consider a single facility and one type of item. At each time period t, the facility can produce from an infinite source for a unit cost p_t, incurring a setup cost f_t. This setup cost accounts for costs such as transportation costs. At the end of each period t, items can be kept in stock, incurring a unit holding cost h_t. The facility must satisfy a demand d_t at each period t. We can write a first mixed integer formulation.

The variable x_t is the amount produced at each period t, s_t is the amount stocked at the end of each period and y_t is a setup variable for the fixed production costs. Since we have no initial stock, we fix $s_{t-1} = 0$. We note $d_{kt} = \sum_{i=k}^{t} d_i$.

$$\text{ULS:} \quad \min \sum_{t \in T} (h_t s_t + f_t y_t + p_t x_t) \tag{ULS.1}$$

$$s_{t-1} + x_t = s_t + d_t, \quad \forall t \in T, \tag{ULS.2}$$

$$x_t \leq d_{k|T|} \cdot y_t, \quad \forall t \in T, \tag{ULS.3}$$

$$s, x \in \mathbb{R}_+^{|T|}, \tag{ULS.4}$$

$$y \in [0,1]^{|T|}, \tag{ULS.5}$$

$$y \in \mathbb{Z}^{|T|}. \tag{ULS.6}$$

The objective function (ULS.1) minimizes the total costs (holding costs, setup costs and production costs). Constraints (ULS.2) represent flow conservation and demand satisfaction and constraints (ULS.3) ensure the valid definition of the setup variables. Constraints (ULS.4) - (ULS.6) are nonnegativity and integrality constraints on the variables. We call $X^{ULS} = \{(x, s, y) : (\text{ULS.2}) - (\text{ULS.6})\}$ the feasible space.

Different classifications exist for the lot sizing problem: number of production stages (single-level or multi-level) or period length (big bucket where more than one item per period can be produced or small bucket where only one item can be produced by period) (see Brahimi et al. (2017)). Back orders can be authorized as well as lost sales. In the single-level structure, each item is independent, whereas, in multi-level structures, the different items are dependent.

The uncapacitated lot sizing problem is known to be solved in $\mathcal{O}(|T|^2)$ with a forward dynamic algorithm proposed by Wagner and Whitin (1958). A faster algorithm in $\mathcal{O}(|T| \log(|T|)$ has been found (see Wagelmans et al. (1992)) with a backward dynamic programming algorithm.

One of the difficulties of solving this problem is finding a description of the convex hull of X^{ULS}, by for example, providing valid inequalities satisfied by all points of X^{ULS}. We introduce here the (l, S) inequalities, where $d_{jl} = \sum_{t=j}^{l} d_t$ (a proof is given in Pochet and Wolsey (2006)).

Proposition 1. *Let $1 \leq l \leq n$, $L = \{1, \ldots, l\}$ and $S \subseteq L$, then the (l, S) inequality*

$$\sum_{j \in S} x_j \leq \sum_{j \in S} d_{jl} y_j + s_l \tag{2}$$

is valid for X^{ULS}.

The (l, S) inequalities give an important result (a proof is given in Pochet and Wolsey (2006)):

Theorem 1. *The constraints* (ULS.2) - (ULS.5), (2) *give a complete linear inequality description of conv(X^{ULS}).*

One can note that however, there is an exponential number of (l, S) inequalities. In branch and cut, the inequalities violated by a given solution can be obtained by separation algorithms. There exist other formulations of the ULS problem which involve more informative variables and have tighter relaxations. We present here two alternative formulations for the ULS.

3.1.1 • ULS - THE TRANSPORTATION FORMULATION

Krarup and Bilde (1977) introduced a formulation with the variable $w_{ut},\ u \leq t$ which is the amount produced in period u to satisfy demand in period t. In this formulation, the variables hold more information than in the formulation ULS.

$$\begin{aligned}
\text{TP-ULS:}\quad \min\quad & \sum_{t\in T}(p_t\sum_{j=t}^{|T|} w_{tj} + f_t y_t + h_t \sum_{k=1}^{t}\sum_{i=t+1}^{|T|} w_{ki}) && \text{(TP-ULS.1)}\\
\text{s.t.}\quad & \text{(ULS.5), (ULS.6) and} && \text{(TP-ULS.2)}\\
& \sum_{u\in T} w_{ut} = d_t, \quad \forall t\in T && \text{(TP-ULS.3)}\\
& w_{ut} \leq d_t y_u, \quad \forall u,t\in T, u\leq t && \text{(TP-ULS.4)}\\
& w_{ut} \geq 0, \quad \forall u,t\in T,\ u\leq t. && \text{(TP-ULS.5)}
\end{aligned}$$

Constraints (TP-ULS.3) are demand-satisfaction constraints. Constraints (TP-ULS.4) are setup constraints. Constraints (TP-ULS.5) and (TP-ULS.2) are nonnegativity and integrality constraints on the variables. We call $(\text{TP-ULS})^{\text{LP}}$ the linear relaxation of formulation TP.

Proposition 2. *The linear program $(\text{TP-ULS})^{LP}$ has an optimal solution with y integer and thus solves LSU.*

A proof of this proposition is given in Barany et al. (1984) where the authors find a solution to ULS with the Wagner-Whitin dynamic programming method, retrieve the value of the variable w and find a solution of the dual of $(\text{TP-ULS})^{\text{LP}}$ which has the same optimal value.

3.1.2 • ULS - THE SHORTEST PATH FORMULATION

There exists an optimal solution to the ULS problem with a special structure called **zero stock policy** (ZIO). This means the facility orders only if it has no stock at the beginning of the period:

Proposition 3. *There exists an optimal solution to ULS in which $s_{t-1}x_t = 0$ for all $t \in T$.*

This result comes from the property that minimum cost networks are acyclic. Eppen and Martin (1987) use this property to introduce a new formulation. We introduce the variable such that $\phi_{ut} = 1$ if an amount $d_{ut} > 0$ is produced in period u, this means that the interval $[u, t]$ is part of the solution. If $[u, t]$ is not part of the solution, we have $\phi_{ut} = 0$.

$$\begin{aligned}
\text{SP-ULS:}\quad \min\quad & \sum_{t\in T}(p_t x_t + q_t y_t + h_t(\sum_{k=1}^{t} x_k - d_{1,t})) && \text{(SP-ULS.1)}\\
\text{s.t.}\quad & \text{(ULS.5), (ULS.6) and} && \text{(SP-ULS.2)}\\
& \sum_{t\in T}\phi_{1t} = 1 && \text{(SP-ULS.3)}\\
& \sum_{u=1}^{t-1}\phi_{u,t-1} - \sum_{\tau=t}^{|T|}\phi_{t\tau} = 0, \quad \forall t\in T,\ t\geq 2 && \text{(SP-ULS.4)}
\end{aligned}$$

$$\sum_{u=1}^{|T|} \phi_{u|T|} = 1 \tag{SP-ULS.5}$$

$$\sum_{\tau=t:d_{t\tau}>0}^{|T|} \phi_{t\tau} \leq y_t, \quad \forall t \in T \tag{SP-ULS.6}$$

$$\sum_{\tau=t}^{|T|} d_{t\tau}\phi_{t\tau} = x_t, \quad \forall t \in T \tag{SP-ULS.7}$$

$$\phi_{ut} \geq 0, \quad \forall u, t \in T, \ u \leq t. \tag{SP-ULS.8}$$

Constraint (SP-ULS.3) ensures that the first period is part of an interval. Constraints (SP-ULS.4) ensure that if an interval finishes in period $t-1$, a new interval begins in period t. Constraint (SP-ULS.5) ensures that an interval finishes at period $|T|$. Indeed, constraints (SP-ULS.3)–(SP-ULS.5) are shortest-path constraints. Constraints (SP-ULS.6) are setup constraints. Constraints (SP-ULS.7) ensure that the right amount of quantity is produced and permit to relate the new variables ϕ to the variables x. Constraints (SP-ULS.8) are nonnegativity and integrality constraints on the variables. We call SP-ULS^{LP} the linear relaxation of formulation SP-ULS.

Proposition 4. *The linear program SP-ULS^{LP} has an optimal solution with y integer and thus solves LSU.*

A proof of this result is given in Pochet and Wolsey (2006) where the authors transform slightly the formulation to show that it is a shortest path problem in an acyclic network, thus the underlying matrix is totally unimodular.

3.2 One Warehouse multi-retailer problem

The one-warehouse multi-retailer problem (OWMR) is a two-level lot-sizing problem. The OWMR is defined over a finite planning horizon of T periods, $1 \leq t \leq |T|$, and only one type of commodity (item) is considered. At each time period $t \in T$, a single warehouse can produce items incurring a setup cost (fixed production cost) denoted by f_t^0. A set of retailers R can then order items from the warehouse, noted by index $c = 0$. We have $C = R \cup \{0\}$ the set of facilities, where index 0 indicates the warehouse. Each retailer $c \in R$ must satisfy a known demand at each time period $t \in T$, denoted by d_t^c. A setup cost (fixed ordering cost) denoted by f_t^c is incurred by retailer $c \in R$ when it orders from the warehouse at period $t \in T$. At the end of each period $t \in T$, items can be kept in stock both at the warehouse and at the retailers incurring a unit holding cost of h_t^c, $c \in C$. A variable cost per unit p_t^c is also incurred whenever a facility $c \in C$ orders or products an item. We define $d_t^0 = \sum_{c \in R} d_t^c$, $d_{ij}^c = \sum_{t=i}^{j} d_t^c$. Since there are no initial inventories, we fix $s_0^c = 0$, $\forall c \in C$. The basic formulation for the OWMR is the following:

$$\text{OWMR:} \quad \min \quad \sum_{t \in T} \sum_{c \in C} (h_t^c s_t^c + f_t^c y_t^c + p_t^c x_t^c) \tag{OWMR.1}$$

$$\text{s.t.}\quad s_{t-1}^0 + x_t^0 = \sum_{c\in R} x_t^c + s_t^0, \quad \forall t \in T, \tag{OWMR.2}$$

$$s_{t-1}^c + x_t^c = d_t^c + s_t^c, \quad \forall c \in R,\ t \in T, \tag{OWMR.3}$$

$$x_t^c \le d_{t,|T|}^c y_t^c, \quad \forall c \in C,\ t \in T, \tag{OWMR.4}$$

$$s, x \in \mathbb{R}_+^{|C|\times|T|}, \tag{OWMR.5}$$

$$y \in \{0,1\}^{|C|\times|T|}. \tag{OWMR.6}$$

The objective (OWMR.1) minimizes the total production cost which includes holding costs, setup costs and production costs. Constraints (OWMR.2) are balance constraints at the warehouse and constraints (OWMR.3) are balance constraints at the retailers, ensuring that the inflow and outflow of goods are equal. Constraints (OWMR.4) are set-up constraints applied to all facilities and constraints (OWMR.5)-(OWMR.6) are non-negativity and integrality constraints.

As with the ULS problem, it is known that ZIO policies are optimal for the OWMR (see Schwarz (1973) for a proof considering an infinite horizon):

Proposition 5. *There exists an optimal solution to OWMR in which $s_{t-1}^c x_t^c = 0$ for all $t \in T$ and for all $c \in C$.*

Arkin et al. (1989) showed that the joint replenishment problem (JRP) is NP-hard. The JRP can be regarded as a special case of the OWMR where storage is not permitted at the warehouse.

Numerous heuristics have been proposed and implemented to tackle the OWMR. Federgruen and Tzur (1999) introduce a time partitioning heuristic. Levi et al. (2008) develop LP rounding techniques that provide an approximation guarantee. Additionally, Gayon et al. (2017) devise a combinatorial algorithm that decomposes the problem into single-echelon problems. Yang et al. (2012) use genetic algorithms to determine replenishment policies.

There exists various formulations for the OWMR. These formulations where introduced by Cunha and Melo (2016) and Solyalı and Süral (2012). Note that we call $F(\cdot)$ the feasible solution space of the LP-relaxation of formulation $(\cdot)$ and $v(\cdot)$ the optimal LP-relaxation objective value of formulation $(\cdot)$.

3.2.1 • Echelon Stock Formulation

We introduce the echelon stock formulation, where $I_t^0 = s_t^0 + \sum_{c\in R} s_t^c$ is the echelon stock (s_t^c is the stock at retailer c at the end of period t and s_t^0 is the stock at the warehouse at the end of period t). The echelon stock at period t represents the total stock in the system at the end of period t. The strength of this formulation is to create individual lot-sizing structures for each facility contrary to the first formulation for OWMR which links production variables and ordering variables with the constraint OWMR.2.

$$\text{ES:}\quad \min \quad \sum_{t\in T}\left(h_t^0 I_t^0 + \sum_{c\in C}(f_t^c y_t^c + x_t^c p_t^c) + \sum_{c\in R}(h_t^c - h_t^0) s_t^c \right) \tag{ES.1}$$

$$
\begin{aligned}
\text{s.t.}\quad & (\text{OWMR.3}), (\text{OWMR.6}) \text{ and} && \text{(ES.2)}\\
& I^0_{t-1} + x^0_t = d^0_t + I^0_t, \quad \forall t \in T, && \text{(ES.3)}\\
& x^c_t \leq d^c_{t,|T|} y^c_t, \quad \forall c \in C,\ t \in T, && \text{(ES.4)}\\
& \sum_{r=1}^{t} x^0_r \geq \sum_{c \in R} \sum_{r=1}^{t} x^c_r, \quad \forall t \in T, && \text{(ES.5)}\\
& x, I \in \mathbb{R}_+^{|C| \times |T|}. && \text{(ES.6)}
\end{aligned}
$$

The objective (ES.1) is equivalent to the objective (OWMR.1). Constraints (ES.3) are flow constraints at the warehouse. Constraints (ES.4) are setup constraints for all facilities. Constraints (ES.5) ensure that the total amount ordered by the retailers is smaller than the total amount ordered by the warehouse. Constraints (ES.6) are nonnegativity and integrality constraints on the variables.

3.2.2 • Strong Echelon Stock Formulation

The strong echelon stock formulation replaces constraints (OWMR.3), (ES.3) and (ES.4) by a stronger transportation formulation that gives the convex hull of feasible solutions for the ULS problem. We have $H^c_{tk} = p^c_t + \sum_{l=t}^{k-1}(h^c_l - h^0_l)$ for $c \in R$ and $H^0_{tk} = p^0_t + \sum_{l=t}^{k-1} h^0_l$ for $c = 0$ which are ordering and echelon stock costs for obtaining an item in period t to satisfy demand at period k at retailers and at the warehouse respectively. Let X^c_{tk} be the quantity ordered by retailer $c \in C$ in period t to satisfy its demand in period k.

$$
\begin{aligned}
\text{SES:}\quad \min\quad & \sum_{c \in C} \sum_{t \in T} f^c_t y^c_t + \sum_{c \in C} \sum_{t \in T} \sum_{k=t}^{|T|} H^c_{tk} X^c_{tk} && \text{(SES.1)}\\
\text{s.t.}\quad & (\text{OWMR.6}) \text{ and} && \text{(SES.2)}\\
& \sum_{t=1}^{k} X^c_{tk} = d^c_k, \quad \forall c \in C,\ k \in T, && \text{(SES.3)}\\
& X^c_{tk} \leq d^c_k y^c_t, \quad \forall c \in C,\ 1 \leq t \leq k \leq |T|, && \text{(SES.4)}\\
& \sum_{r=1}^{t} \sum_{k=r}^{|T|} X^0_{rk} \geq \sum_{c \in C} \sum_{r=1}^{t} \sum_{k=r}^{|T|} X^c_{rk}, \quad \forall t \in T, && \text{(SES.5)}\\
& X^c_{tk} \geq 0, \quad \forall c \in C,\ 1 \leq t \leq k \leq |T|. && \text{(SES.6)}
\end{aligned}
$$

The objective function (SES.1) is equivalent to (OWMR.1). Constraints (SES.3) ensure that the correct amount is ordered for each facility. Constraints (SES.4) are setup constraints and constraints (SES.5) ensure that the total amount ordered by the retailers is at least the total amount that has been ordered by the warehouse. Constraints (SES.2) and (SES.6) are nonnegativity and integrality constraints on the variables.

3.2.3 • Transportation Formulation

The ZIO property for the OWMR states that the total cost of order d^c_t is always provided by a single pair (r, k) such that the warehouse products at period r and the retailer orders at period k. Let W^c_{qtk} be the quantity ordered by the warehouse in period q, sent to the retailer c in period t to satisfy demand in period k. We use the variables X from the transportation formulation at the retailers. Let $H^{c\prime}_{tk} = p^c_t + \sum_{l=t}^{k-1} h^c_l$ be the unit cost of satisfying the demand at $c \in C$ in period k by placing an order in period t. The TP formulation is as follows:

$$\text{TP:} \quad \min \quad \sum_{c\in C}\sum_{t\in T} f^c_t y^c_t + \sum_{c\in R}\sum_{q\in T}\sum_{t=q}^{|T|}\sum_{k=t}^{|T|} H^{0\prime}_{qt} W^c_{qtk} + \sum_{c\in R}\sum_{t\in T}\sum_{k=t}^{|T|} H'_{ctk} X^c_{tk} \tag{TP.1}$$

$$\text{s.t.} \quad \sum_{q=1}^{t} W^c_{qtk} = X^c_{tk}, \quad \forall c \in R, \quad 1 \le t \le k \le |T|, \tag{TP.2}$$

$$\sum_{t=q}^{k} W^c_{qtk} \le d^c_k y^0_q, \quad \forall c \in R,\ 1 \le q \le k \le |T|, \tag{TP.3}$$

$$\sum_{t=1}^{k} X^c_{tk} = d^c_k, \quad \forall c \in R,\ k \in T, \tag{TP.4}$$

$$X^c_{tk} \le d^c_k y^c_t, \quad \forall c \in R,\ 1 \le t \le k \le |T|, \tag{TP.5}$$

$$X^c_{tk} \ge 0 \quad \forall c \in R,\ 1 \le t \le k \le |T|, \tag{TP.6}$$

$$W^c_{qtk} \ge 0, \quad \forall c \in R,\ 1 \le q \le t \le k \le |T|, \tag{TP.7}$$

$$y^c \in [0,1]^{|T|}, \quad \forall c \in R \tag{TP.8}$$

$$y^0 \in \{0,1\}^{|T|}. \tag{TP.9}$$

The objective (TP.1) is equivalent to (OWMR.1). Constraints (TP.2) ensures that the quantity ordered by retailer c at t to satisfy demand in period k is ordered by the warehouse in periods before t. Constraints (TP.3) are setup constraints at the warehouse. Constraints (TP.4) ensure that the demand at period k at retailer c is ordered in periods 1 through k. Constraints (TP.5) are setup constraints at the retailers. Constraints (TP.6)– (TP.9) are nonnegativity and integrality constraints on the variables.

3.2.4 • Compact Transportation Formulation

Let $\hat{H}^c_{qtk} = H'^0_{qt} + H'^c_{tk}$. We eliminate the variables X using equation (TP.2).

$$\text{TPC:} \quad \min \quad \sum_{c\in C}\sum_{t\in T} f^c_t y^c_t + \sum_{c\in C}\sum_{q\in T}\sum_{t=q}^{|T|}\sum_{k=t}^{|T|} \hat{H}^c_{qtk} W^c_{qtk} \tag{TPC.1}$$

$$\text{s.t.} \quad \text{(TP.3), (TP.8), (TP.9) and} \tag{TPC.2}$$

$$\sum_{t=1}^{k}\sum_{q=1}^{t} W^c_{qtk} = d^c_k, \quad \forall c \in R,\ k \in T, \tag{TPC.3}$$

$$\sum_{q=1}^{t} W^c_{qtk} \le d^c_k y^c_t, \quad \forall c \in R,\ 1 \le t \le k \le |T|, \tag{TPC.4}$$

$$W^c_{qtk} \ge 0, \quad \forall c \in C,\ 1 \le q \le t \le k \le |T|. \tag{TPC.5}$$

The objective (TPC.1) is equivalent to (OWMR.1). Constraints (TPC.3) define the variable W at the retailers. Constraints (TPC.4) are setup constraints at the retailers.

3.2.5 • Shortest Path Formulation

In this formulation, we exploit the property that there exists an optimal solution that can be decomposed into sub-intervals with the ZIO policy. We combine here a shortest path formulation for the retailers and a transportation formulation for the warehouse. In this formulation, Z^c_{tk} is the fraction of the total demand at $c \in R$ in period t through k that is ordered by c in period t. Let U^c_{qtk} be the fraction of the total demand at retailer c in period t through k that is ordered by the warehouse in period q and sent to c in t. Let $G^c_{tk} = p^c_t d^c_{t,k} + \sum_{l=t}^{k-1} h^c_l d^c_{l+1,k}$ and let a^c_{tk} be equal to 1 if $d^c_{t,k} > 0$, 0 otherwise.

$$
\begin{aligned}
\text{SP:} \quad \min \quad & \sum_{c \in C} \sum_{t \in T} f^c_t y^c_t + \sum_{c \in R} \sum_{t \in T} \sum_{k=t}^{|T|} \sum_{q=1}^{t} H'^0_{qt} d^c_{t,k} U^c_{qtk} + \sum_{c \in R} \sum_{t \in T} \sum_{k=t}^{|T|} G^c_{tk} Z^c_{tk} && \text{(SP.1)} \\
\text{s.t.} \quad & \text{(TP.8), (TP.9) and} && \text{(SP.2)} \\
& \sum_{q=1}^{t} U^c_{qtk} = Z^c_{tk}, \quad \forall c \in R,\ 1 \le q \le k \le |T|, && \text{(SP.3)} \\
& \sum_{k=q}^{t} \sum_{r=t}^{|T|} a^c_{kr} U^c_{qkr} \le y^0_q, \quad \forall c \in R,\ 1 \le q \le t \le |T|, && \text{(SP.4)} \\
& \sum_{t \in T} Z^c_{1t} = 1, \quad \forall c \in R, && \text{(SP.5)} \\
& \sum_{k=t}^{|T|} Z^c_{tk} - \sum_{k=1}^{t-1} Z^c_{k,t-1} = 0, \quad \forall c \in R,\ 2 \le t \le |T|, && \text{(SP.6)} \\
& \sum_{k=t}^{|T|} a^c_{tk} Z^c_{tk} \le y^c_t, \quad \forall c \in R,\ t \in T, && \text{(SP.7)} \\
& Z^c_{tk} \ge 0, \quad \forall c \in R,\ 1 \le t \le k \le |T|, && \text{(SP.8)} \\
& U^c_{qtk} \ge 0, \quad \forall c \in R,\ 1 \le q \le t \le k \le |T|. && \text{(SP.9)}
\end{aligned}
$$

Constraints (SP.3) links the variables U and Z: a fraction of the demand at retailer c between t and k, sent to c in t has to be ordered by the warehouse in periods 1 through t. Constraints (SP.4) are setup constraints at the warehouse. Constraints (SP.5) - (SP.6) are shortest path constraints at the retailers. Constraints (SP.7) are setup constraints at the retailers. Constraints (SP.8) - (SP.2) are nonnegativity and integrality constraints on the variables.

3.2.6 • Compact Shortest path formulation

We can eliminate the variables Z using equation (SP.3).

$$\text{SPC:} \quad \min \quad \sum_{c\in C}\sum_{t\in T} f_t^c y_t^c + \sum_{c\in R}\sum_{t\in T}\sum_{k=t}^{|T|}\sum_{q=1}^{t}\left(H_{qt}^{0\prime} d_{t,k}^c + G_{tk}^c\right) U_{qtk}^c \tag{SPC.1}$$

$$\text{s.t.} \quad (\text{TP.8}), (\text{TP.9}), (\text{SP.4}), (\text{SP.8}) \text{ and} \tag{SPC.2}$$

$$\sum_{t\in T} U_{11t}^c = 1, \quad \forall c \in R, \tag{SPC.3}$$

$$\sum_{k=t}^{|T|}\sum_{q=1}^{t} U_{qtk}^c - \sum_{k=1}^{t-1}\sum_{q=1}^{k} U_{qk,t-1}^c = 0, \quad \forall c \in R,\ 2 \le t \le |T|, \tag{SPC.4}$$

$$\sum_{k=t}^{|T|}\sum_{q=1}^{t} a_{tk}^c U_{qtk}^c \le y_t^c, \quad \forall c \in R,\ t \in T. \tag{SPC.5}$$

Constraints (SPC.3)- (SPC.4) are shortest path constraints for the variable U. Constraints (SPC.5) are setup constraints.

3.2.7 • Multi Commodity Formulation

In the multicommodity formulation, each demand d_t^c is viewed as a distinct commodity, that is to say, a different item. Let w_{kt}^{0c} be the amount produced at the warehouse in period k to satisfy the demand of retailer c in period t. Let w_{kt}^{1c} be the amount transported from the warehouse to retailer c in period k to satisfy demand in period k. Let σ_{kt}^{0c} (respectively σ_{kt}^{1c}) be the amount stocked at the warehouse (resp. at retailer c) at the end of period k to satisfy the demand of retailer c at period t. Let δ_{kt} be the Kronecker delta symbol.

$$\text{MC:} \quad \min \quad \sum_{t\in T}\left(f_t^0 y_t^0 + \sum_{c\in R}\left[f_t^c y_t^c + \sum_{k=t}^{|T|}(p_t^0 w_{tk}^{0c} + h_t^0 \sigma_{tk}^{0c} + p_t^c w_{tk}^{1c} + h_t^c \sigma_{tk}^{1c})\right]\right) \tag{MC.1}$$

$$\text{s.t.} \quad (\text{TP.8}), (\text{TP.9}) \text{ and} \tag{MC.2}$$

$$\sigma_{k-1,t}^{0c} + w_{kt}^{0c} = w_{kt}^{1c} + \sigma_{kt}^{0c}, \quad \forall c \in R,\ 1 \le k \le t \le T, \tag{MC.3}$$

$$\sigma_{k-1,t}^{1c} + w_{kt}^{1c} = \delta_{kt} d_t^c + (1-\delta_{kt})\,\sigma_{kt}^{1c}, \quad \forall c \in R,\ 1 \le k \le t \le |T|, \tag{MC.4}$$

$$w_{kt}^{0c} \le d_t^c y_k^0, \quad \forall c \in R,\ 1 \le k \le t \le |T|, \tag{MC.5}$$

$$w_{kt}^{1c} \le d_t^c y_k^c, \quad \forall c \in R,\ 1 \le k \le t \le |T|, \tag{MC.6}$$

$$w^0, w^1, \sigma^0, \sigma^1 \in \mathbb{R}_+^{|R|\times|T|\times|T|}. \tag{MC.7}$$

Constraints (MC.3) are flow constraints at the warehouse. Constraints (MC.4) are flow constraints at the retailers. Constraints (MC.5) - (MC.6) are setup constraints respectively at the warehouse and at the retailers. Constraints (MC.7) - (MC.2) are nonnegativity and integrality constraints on the variables.

3.2.8 • Properties of the formulations

Let F be a formulation in [TP, TPC, SP, SPC, MC].

Proposition 6. *When the variables y^0_t, $t \in T$ are fixed, there exists an optimal solution of the formulation F with integral values y^c_t. ($c \in R,\ t \in T$) (Cunha and Melo (2016); Solyalı and Süral (2012))*

For the transportation and multicommodity formulation, the authors show that the problem can be seen as $|C|$ uncapacitated lot sizing problems and that the formulation for each subproblem describes exactly the convex hull of the feasible space. For the shortest path formulation, the authors establish that they obtain a totally unimodular matrix thus the formulation is tight.

Other results obtained in articles by Cunha and Melo (2016) and Solyalı and Süral (2012) cover the tightness of the formulations:

Proposition 7. $v(SES) \leq v(MC) = v(TP) \leq v(SP)$.

3.3 Results

We ran computational tests. All experiments were conducted on a Linux x86_64 machine equipped with an Intel Core i7-7700 3.60 GHz processor and 62Go of RAM. The code was implemented in Python 3.9.14 and CPLEX 22.12 was utilized. During all executions, a time limit of 7,200 seconds was enforced.

Instances are generated randomly, similarly as in Solyalı and Süral (2012). For one value of (C, T) we have four possible instances types, depending if the demand (D) and setup production costs (F) at the warehouse are dynamic (D) or static (S) (independently one from another). Demand at the retailers d^c_t is generated from $U[5, 100]$. Setup production costs at the warehouse are generated from $U[1500, 4500]$. Stock at the warehouse h^0_t is always static and equal to 0.5. stock holding at retailers h^c_t are static and generated in $U[0.5, 1]$, independently for every retailer. Set-up costs at retailers are always dynamic and generated in $U[5, 100]$, independently for every retailer. With the exception of the storage costs, all parameters are integer-valued. We do not consider unit production costs. Ten instances are considered for each combination (DD-DF, DD-SF, SD-SF, SD-DF). We considered $|T| \in \{15, 20, 25, 30\}$ and $|R| \in \{50, 75, 100\}$. Extended results for all computational experiments can be found in the appendix.

Table 1 represents the average solving time for instances with $|R| \in \{50, 75, 100\}$ and $|T| \in \{15, 20, 25\}$. There are no constraints on production periods. These results are aggregated for all types of instances (static and dynamic). Our goal is to study the efficiency of each formulation.

$\|T\|$	$\|R\|$	ES	MC	OWMR	SES	SP	SPC	TP	TPC	Mean
15	50	6.46	0.27	13.09	2.47	0.38	0.35	0.29	0.26	2.95
	75	13.83	0.38	28.09	4.02	0.51	0.48	0.42	0.37	6.01
	100	73.68	0.94	64.64	8.12	1.00	0.85	0.85	0.78	18.86
20	50	15.66	0.43	99.67	5.34	0.85	0.80	0.56	0.50	15.48
	75	149.34	0.96	245.30	14.98	1.68	1.41	1.17	1.01	51.98
	100	400.44	1.56	310.42	30.79	2.11	1.81	1.70	1.52	63.79
25	50	110.56	0.84	871.10	20.06	1.76	1.75	1.13	0.99	126.02
	75	649.37	1.65	1434.15	83.09	3.02	2.85	2.02	1.79	272.24
	100	2119.35	3.23	1568.93	217.55	5.53	4.47	3.72	3.38	490.77

Table 1: Average solving time (in seconds)

We found that formulations TP, TPC, SP, SPC and MC outperform significantly OWMR, ES and SES. The MC formulation is the most effective formulation followed by the transportation formulation. This can be explained by a tradeoff between the tightness of the formulation (and thus exploring fewer nodes in the branch and bound tree for instance) and the complexity of the formulation (for example the number of variables and constraints). The more a formulation has variables and constraints, the harder solving each node of the branch and bound tree is. These results are cohesive with proposition 6 and the respective strength of each relaxation. Our results are coherent with the findings of Solyalı and Süral (2012). The authors found that formulations TPC and SPC perform significantly better than ES and SES. Also, SES performs better than ES. Extensive computational experiments were also conducted in Cunha and Melo (2016). They found that the multicommodity formulation MC performs better than other formulations in most cases.

One of the difficulties encountered was that setting up the model (giving CPLEX all the variables, the constraints and the objective function) could take significantly more time than solving the model itself. This was particularly true for formulations with a high number of variables and/or constraints, see Table 2 for the formulations' sizes. Yet, we did not study extensively ways to solve this problem as our goal was to study the models' solving time and our code was in Python, which is not the most efficient programming language (compared to C++ for instance).

Formulation	Constraints	Binary variables	Continuous variables
OWMR, ES	$\mathcal{O}(\|N\|\|T\|)$	$\mathcal{O}(\|N\|\|T\|)$	$\mathcal{O}(\|N\|\|T\|)$
SES	$\mathcal{O}(\|N\|\|T\|^2)$	$\mathcal{O}(\|N\|\|T\|)$	$\mathcal{O}(\|N\|\|T\|^2)$
TP, TPC	$\mathcal{O}(\|N\|\|T\|^2)$	$\mathcal{O}(\|T\|)$	$\mathcal{O}(\|N\|\|T\|^3)$
SP, SPC	$\mathcal{O}(\|N\|\|T\|^2)$	$\mathcal{O}(\|T\|)$	$\mathcal{O}(\|N\|\|T\|^3)$
MC	$\mathcal{O}(\|N\|\|T\|^2)$	$\mathcal{O}(\|T\|)$	$\mathcal{O}(\|N\|\|T\|^2)$

Table 2: Size of constraints, binary and continuous variables in the formulations

We now study the modelisation of different limitations on the production periods. Section 4 studies the implementation of fixed production periods. This means that the periods where production is permitted are defined in advance. Section 5 studies cases where the production periods are not fixed but are under several limitations concerning the total number of periods or the number of periods between two production periods. For these constraints and limitations, we study different formulations and compute computational experiments.

4 FIXED PRODUCTION PERIODS

The first type of production constraints considers that production can only be done at predefined periods, i.e., production is forbidden at known time periods. We denote by $X \subseteq T$ the set of forbidden periods. We note $Y = T \setminus X$ the set of authorized periods, $Y = \{i_1 < \dots < i_n\}$. We consider that we always have $1 \in Y$ because there is no initial inventory and $d_0^0 > 0$. This means that the first period is always authorized. There are different ways to select the set X of forbidden periods.

- We can select regularly spaced periods with a parameter τ: $X = \{1 + \tau \cdot k \,; k \in \mathbb{N}\}$.
- We can select τ random periods. We always need to select the first period to obtain a feasible problem: $X = \{1\} + \{\sigma_k\}$ where $\{\sigma_k,\ k \in [\![1, \tau - 1]\!]\}$ is a random set of $\tau - 1$ items in $[\![2, |T|]\!]$.

We chose to work on production constraints rather than ordering constraints: for formulations MC, TP, TPC, SP and SPC, the introduction of additional constraints for the variables $y_c^t,\ c \in R$ makes it impossible to relax these variables. The parameterization of the constraints would thus be more advanced. This choice is coherent with a situation where production can only happen every week for instance yet delivery supplies happen every day. We recall that the warehouse *produces* the items whereas retailers *order*.

We study different ways to add such constraints. First, we consider exact methods. The first methods add constraints fixing the y variables. The second method changes the value of parameter $f_t^0,\ \forall t \in X$. The last method introduces echelon stock constraints. Finally, we develop a heuristic based on the uncapacitated two-level lot-sizing problem.

4.1 Exact methods through modelling

We consider three methods to model fixed production periods.

Method 1 consists of fixing the values of the y-variables by adding

$$y_t^0 = 0,\ \forall t \in X. \tag{13}$$

Method 2 considers a high production setup cost at forbidden periods by setting $f_t^0 = M_t,\ \forall t \in X$, where M_t is a high value. We consider two ways to compute M_t: a naive value where $M_t = \infty,\ \forall t \in X$, and a value which relies on computing upper bounds on the costs, $\tilde{M}_t,\ \forall t \in X$. For $M_t = \infty,\ \forall t \in X$ we kept the default CPLEX parameter for infinity, 10^{20}. In order to set $\tilde{M}_t,\ \forall t \in X$, we need hypotheses on the parameters (see section 4.1.1 for proof).

Method 3 consists of imposing additional constraints which model restrictions on the stock between consecutive non-forbidden periods, through the echelon-stock concept. With the notation from the echelon stock formulation, we give here the additional constraints (see section 4.1.2 for proof):

$$I_{i_j}^0 - I_{i_{j+1}-1}^0 \geq \sum_{k=i_j+1}^{i_{j+1}-1} d_k^0, \quad 1 \leq j \leq n-1 \tag{14}$$

4.1.1 • High production setup cost

We call OWMR-M the one-warehouse multi-retailer problem where the value of parameter f_t^0 has been changed for $t \in X$, the set of forbidden periods: $f_t^0 = M_t,\ \forall t \in X$. We consider that we work in a situation where $p_t^c = 0,\ \forall t \in T,\ c \in C$.

Hypothesis 1 (H1)**.** *There exists $\overline{h^0}$ such that $h_t^0 \leq \overline{h^0},\ \forall t \in T$.*

Hypothesis 2 (H2)**.** *There exists $\theta \leq |T|,\ \theta \in \mathbb{N}$ such that $i_{j+1} - i_j \leq \theta + 1,\ 1 \leq j < n$ and $T - i_n \leq \theta$. This means we have at most θ consecutive periods in X.*

Hypothesis 3 (H3)**.** *There exists $\overline{f^0}$ such that $f_t^0 \leq \overline{f^0},\ \forall t \in T$.*

Hypothesis 4 (H4)**.** *There exists $\overline{f}$ such that $f_t^c \leq \overline{f},\ \forall t \in T,\ c \in R$.*

We recall that a **zero stock policy (ZIO)** (s, x, y) has the following property:

$$s_{t-1}^c \cdot x_t^c = 0, \quad \forall t \in T,\ c \in C \tag{15}$$

We recall that we have the following notation: $d_t^0 = \sum_{c \in R} d_t^c$ and $d_{jk}^c = \sum_{t=j}^k d_t^c$.

Lemma 2. *Under (H1), (H2), (H3) and (H4) if*

$$M_t \geq \overline{f^0} + |R| \cdot \overline{f} + \theta \cdot \overline{h^0} \cdot d_{t,|T|}^0 \text{ for all } t \in X \tag{16}$$

then there exists an optimal solution of OWMR-M which respects (13)*.*

Sketch of the proof. Let (s, x, y) be a solution of OWMR-M which respects (15) of value v. For such a solution, we note $\Omega \equiv \{l_1 \leq \cdots \leq l_I \mid y_{l_t}^0 = 1,\ \forall t \in [1, I]\}$ the set of periods where orders are made at the warehouse. We note $L(s, x, y) \equiv \sum_{t \in X} y_t^0$ the number of setups in X. Then, given $A \subset T$ an interval of T, we note $v(A)$ the costs of the solution on interval A:

$$v(A) \equiv \sum_{t \in A} \sum_{c \in C} (h_t^c s_t^c + f_t^c y_t^c + p_t^c x_t^c)$$

We then have:

$$v = v(T) = \sum_{t=1}^{I-1} v([l_t, l_{t+1}]) \equiv \sum_{t=1}^{I-1} v_t$$

We suppose that $L(s,x,y) > 0$. We wish to establish that there exists a solution (s', x', y') with value v' such that $v' < v$ and $L(s,x,y) > L(s',x',y')$.

Let $k \in X \cap \Omega$ such that:

$$\exists j \text{ such that } k = l_j$$
$$i_{j-1} \in Y$$

k is a period in X such that the warehouse products at k and the warehouse's previous production period is in Y. We note $\underline{k} = \max\{t \in Y, t \leq k\}$ the closest period in Y before k. We define the following intervals:

- $A = [l_{j-1}, k]$ so $v(A) = v_{j-1}$
- $B = [k, l_{j+1}]$ so $v(B) = v_j$
- $A' = [l_{j-1}, \underline{k}]$
- $B' = [\underline{k}, l_{j+1}]$
- $C' = [\underline{k}, k]$

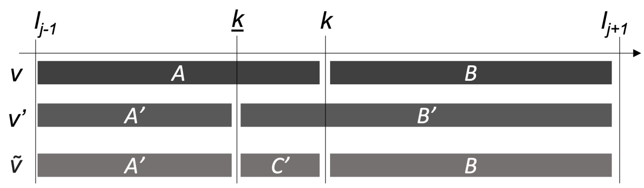

Figure 2: Definition of A, B, A', B' and C'

We want to create an instance (s', x', y') (of value v') with the following production periods: $\{l_1 \dots l_{j-1}, \underline{k}, l_{j+1} \dots l_I\}$ and (s', x', y') is equal to (s, x, y) on all intervals except $[l_{j-1}, l_{j+1}]$. We also create an instance $(\tilde{s}, \tilde{x}, \tilde{y})$ (of value $\tilde{v}$) with the following production periods: $\{l_1 \dots l_{j-1}, \underline{k}, k, l_{j+1} \dots l_I\}$ and $(\tilde{s}, \tilde{x}, \tilde{y})$ is equal to (s, x, y) on all intervals except $[l_{j-1}, l_{j+1}]$. We recall that (s', x', y') and $(\tilde{s}, \tilde{x}, \tilde{y})$ solution values are the sum of the cost other each of their productions intervals. We have the following property in $[l_{j-1}, l_{j+1}]$:

$$v(A) + v(B) \geq v'(A') + v'(B') \Rightarrow v \geq v' \tag{17}$$

We define (s', x', y') and $(\tilde{s}, \tilde{x}, \tilde{y})$:

- On interval A', (s', x', y') and $(\tilde{s}, \tilde{x}, \tilde{y})$ are defined as the optimal solution on interval A' while getting the same initial stock at l_{j-1} that there is in solution (s, x, y) at the same period at each facility and keeping no stock at period $\underline{k}$. We have $(s'_t, x'_t, y'_t) = (\tilde{s}_t, \tilde{x}_t, \tilde{y}_t),\ \forall t \in A'$.

- On interval C', $(\tilde{s}, \tilde{x}, \tilde{y})$ is defined as the optimal solution on interval C' while getting no initial stock at $\underline{k}$ at each facility and keeping at period k the same stock at each facility than in solution (s, x, y) at the period k.

- On interval B', (s', x', y') is defined as the optimal solution on interval B' while getting no initial stock at $\underline{k}$ at each facility and keeping at l_{j+1} the same stock than in solution (s, x, y) at the period l_{j+1}.

- On interval B, $(\tilde{s}, \tilde{x}, \tilde{y})$ is defined equal to (s, x, y).

(s', x', y') and $(\tilde{s}, \tilde{x}, \tilde{y})$ are solutions to the OWMR. We then have, since the initial and final inventories are the same:

$$\tilde{v}(A') + \tilde{v}(C') \leq v(A) + f^0_{\underline{k}} + \sum_{c \in R} f^c_{\underline{k}} \tag{18}$$

We also have:

$$v'(B') \leq \tilde{v}(C') + v(B) - M_k + \sum_{t=\underline{k}}^{k-1} h^0_t \cdot d^0_{k,l_{j+1}} \tag{19}$$

In (19), $v(B) - M_k$ is the cost on the interval B of the solution v if all the items are already at the warehouse in period k at the end of C'. Indeed, M_k is exactly the set-up cost in period $k \in X$. $\sum_{t=\underline{k}}^{k-1} h^0_t \cdot D_{k,l_{j+1}}$ is the maximum cost to bring these items from period $\underline{k}$ to period k. The right term in (19) is the cost to satisfy the demand during the interval B' when we remove the set-up in period k and we manually bring the necessary items from period $\underline{k}$ to period k. Since v' is defined as the optimal solution on interval B' with only one setup at period $\underline{k}$, the inequality is valid. Note that we need some stock at the end of C' because this stock is used in interval B. We then have:

$$\begin{aligned} v(A') + v(B') &\leq^{(19)} v(A') + \tilde{v}(C') + v(B) - M_k + \sum_{t=\underline{k}}^{k-1} h^0_t \cdot d^0_{k,l_{j+1}} \\ &\leq^{(18)} v(A) + f^0_{\underline{k}} + \sum_{c \in R} f^c_{\underline{k}} + v(B) - M_k + \sum_{t=\underline{k}}^{k-1} h^0_t \cdot d^0_{k,l_{j+1}} \end{aligned}$$

Using (17), we want $v(A) + v(B) \geq v'(A') + v'(B')$ so:

$$f^0_{\underline{k}} + \sum_{c \in R} f^c_{\underline{k}} - M_k + \sum_{t=\underline{k}}^{k-1} h^0_t \cdot d^0_{k,l_{j+1}} \leq 0$$

$$(\Leftrightarrow f^0_{\underline{k}} + \sum_{c \in R} f^c_{\underline{k}} + \sum_{t=\underline{k}}^{k-1} h^0_t \cdot d^0_{k,l_{j+1}} \leq M_k)$$

then we can apply (17). We know that:

$$f^0_{\underline{k}} + \sum_{c \in R} f^c_{\underline{k}} + \sum_{t=\underline{k}}^{k-1} h^0_t \cdot d^0_{k,l_{j+1}} \leq \overline{f^0} + |R| \cdot \overline{f} + \theta \cdot \overline{h^0} \cdot d^0_{k,T}$$

If $\overline{f^0} + |R| \cdot \overline{f} + \theta \cdot \overline{h^0} \cdot d^0_{k,T} \leq M_k$ then for every solution which respects the ZIO policy, there exists a solution with strictly fewer production periods in X and a better value. Yet we know that there exists an optimal solution of OWMR-M which respects ZIO policy. Thus there exists a solution of OWMR-M with no orders in X which is also an optimal solution. □

4.1.2 • Echelon stock constraints

Let (ES-F) be the following problem:

$$\text{(ES-F): (OWMR.6), (ES.1)} - \text{(ES.6), (13)}$$

We recall that we have Y the set of authorized periods: $Y = \{i_1, \ldots, i_n\}$. We define the following problem (where $i_{n+1} = |T| + 1$), the echelon stock formulation with echelon stock constraints:

$$\text{(ES-ESC): (OWMR.6), (ES.1)} - \text{(ES.6)} \tag{20}$$

$$I^0_{i_j} - I^0_{i_{j+1}-1} \geq \sum_{k=i_j+1}^{i_{j+1}-1} d^0_k, \quad 1 \leq j \leq n \tag{21}$$

These constraints ensure that there is enough echelon stock at the beginning of an interval of forbidden production periods. To avoid the warehouse from sending its entire stock to retailers and placing orders within a restricted period, it is essential to include the term $I^0_{i_{j+1}-1}$ in the equation. The warehouse could be greedy as it could be cheaper to order in a restricted period and keep the items in its stock. There could still be a ZIO policy that gives such results as the warehouse could send its stock to the retailers.

Lemma 3. *There exists an optimal solution for (ES-ESC) which is also optimal for (ES-F).*

Sketch of the proof. Let (x, I, y) be a feasible solution for ES-F. We show that (x, I, y) is also feasible for ES-ESC. Let $j \in [1, n]$. Using the equation flow (ES.3) on periods in $[i_j, i_{j+1} - 1]$:

$$I^0_{i_j} = \sum_{t=i_j+1}^{i_{j+1}-1} d^0_t - \sum_{t=i_j+1}^{i_{j+1}-1} x^0_t + I^0_{i_{j+1}-1}$$

since we have the setup constraints (ES.4) we have $x^0_t = 0,\ \forall t \in [i_j + 1, i_{j+1} - 1] \subset X$. So:

$$I^0_{i_j} - I^0_{i_{j+1}-1} = \sum_{t=i_j+1}^{i_{j+1}-1} d^0_t$$

(x, I, y) respects (21) and is feasible for (ES-ESC). We have $v(\text{ES-ESC}) \leq v(\text{ES-F})$.

Let (x, I, y) be an optimal solution for (ES-ESC). We show that (x, I, y) is also feasible for (ES-F). We suppose that there exists a period $i \in X$ such that $y_i^0 = 1$. We note j the subscript such that $i_j < i < i_{j+1}$. Since (x, I, y) is optimal, we have $x_i^0 > 0$ (we assume the setup costs y_t^0 are strictly positive). Then, using (ES.3):

$$I_{i_j}^0 = \sum_{t=i_j+1}^{i_{j+1}-1} d_t^0 - \sum_{t=i_j+1}^{i_{j+1}-1} x_t^0 + I_{i_{j+1}-1}^0 < \sum_{t=i_j+1}^{i_{j+1}-1} d_t^0 + I_{i_{j+1}-1}^0$$

$$\text{So} \quad I_{i_j}^0 - I_{i_{j+1}-1}^0 < \sum_{t=i_j+1}^{i_{j+1}-1} d_t^0$$

Yet this is not possible since (x, I, y) is feasible for (ES-ESC). So for all $t \in X$, since $x_t^0 = 0$ and $f_t^0 > 0$, we have $y_t^0 = 0$. (x, I, y) is feasible for (ES-F). So $v(\text{ES-F}) \leq v(\text{ES-ESC})$.

□

The interested reader can find the detailed expression of these constraints for the alternative formulations in the appendix.

4.2 Heuristic

In this section, we build a heuristic based on the article by Melo and Wolsey (2010). They propose a dynamic programming algorithm to solve the uncapacitated two-level lot sizing problem. This problem is a simplification of the OWMR: the difference is that there is only one retailer (called level 1) ordering from the warehouse (called level 0). We present it in this section because an adaptation can be derived from this exact algorithm into a heuristic for our problem. The main idea of the algorithm is that production (at the warehouse level) and ordering (at the retailer level) can be decomposed into intervals, as in the shortest path formulation. If there exists an optimal solution such that $x_j^1 = d_{jt}$ (i.e., the retailer orders for periods j through t) then there exists p, q such that $1 \leq u \leq p \leq j \leq t \leq q$ and $x_u^0 = d_{pq}$. The production batches at level 0 wrap the ordering batches at level 1.

$$\text{2LULS:} \quad \min \quad \sum_{c=0}^{1} \sum_{t=1}^{|T|} p_t^c x_t^c + h_t^c s_t^c + f_t^c y_t^c \tag{22}$$

$$s_{t-1}^0 + x_t^0 = x_t^1 + s_t^0, \quad \forall t \in T \tag{23}$$

$$s_{t-1}^1 + x_t^1 = d_t + s_t^1, \quad \forall t \in T \tag{24}$$

$$x_t^c \leq M y_t^c, \quad \forall c \in \{0, 1\}, \ \forall t \in T \tag{25}$$

$$x, s \in \mathbb{R}_+^{2\times|T|}, \ y \in \{0, 1\}^{2\times|T|} \tag{26}$$

Given $G(t)$ the minimum cost of the two-level problem restricted to periods 1 to t and $H(u, t)$ the minimum cost of satisfying periods u to t at level 1, we have:

$$G(t) = \min_{1\leq j\leq t} \left\{ G(j-1) + \min_{1\leq i\leq j} \left(f_i^0 + \tilde{p}_i^0 d_{jt} \right) + H(j, t) \right\},$$

$$H(u,t) = \min_{u \le j \le t} \left\{ H(u,j-1) + f_j^1 + \tilde{p}_j^1 d_{jt} \right\}.$$

$$\text{where} \quad \tilde{p}_t^0 = p_t^0 + \sum_{l=t}^{|T|} h_l^0, \quad \forall t \in T$$

$$\tilde{p}_t^c = p_t^c + \sum_{l=t}^{|T|} (h_l^c - h_l^0), \quad \forall t \in T,\ c \in R$$

It is important to note that there is a small calculation to be made to remove the variables s from the objective function (using the flow conservation equations (OWMR.2) - (OWMR.3)). We must then retrieve the amount $\sum_{t \in T} h_t^c d_{1t}^c$ from the objective function.

Proposition 8. *Melo and Wolsey (2010) There is a $\mathcal{O}(|T|^2 \log(|T|)$ algorithm for solving the uncapacitated two-level lot sizing problem.*

The full proof is given in the article Melo and Wolsey (2010), but we give here the main ideas. For convenience, we note $\chi_{j,t} = \min_{1 \le i \le j}(f_i^0 + \tilde{p}_i^0 d_{j,t}^c)$ and $\phi^j(b) = \min_{1 \le i \le j}(f_i^0 + \tilde{p}_i^0 b)$. $H(u,t)$ is a ULS problem so it can be calculated in time $\mathcal{O}(|T| \log(|T|)$ for fixed u. The authors use properties of the piecewise linear concave function $\phi^j(b)$ to prove that all its values can be calculated in $\mathcal{O}(|T|^2)$ time. They finally show that the calculation of $G(t)$ for all t can be made in $\mathcal{O}(|T|^2)$ once H is calculated.

We can transform this algorithm to account for forbidden production periods at the warehouse (level 0) and to save the periods where production occurs at the warehouse level. In Melo and Wolsey (2010), the function $\phi^j(b)$ is described by a list of triple points $(\alpha_k^j, \beta_k^j, \gamma_k^j)$ such that α_k^j are the breakpoints with $0 = \alpha_1^j < \alpha_2^j < \cdots < \alpha_{q_j}^j$, β_k^j are the slopes (β_k^j being the slope at the right of point α_k^j) and $\gamma_k^j = \phi^j(\alpha_k^j)$. The parameters α, β and γ are presented in Figure 3. We add a fourth point i describing ϕ^j such that $\beta_k^j = \tilde{p}_i^0$, the subscript of the corresponding period in T. When calculating $\chi_{j,t}$, we can retrieve $\kappa_{j,t} = \text{argmin}_{1 \le i \le j}(f_i^0 + \tilde{p}_i^0 d_{j,t}^c)$. We write $\Omega(t)$ the production periods at level zero for the partial solution $G(t)$.

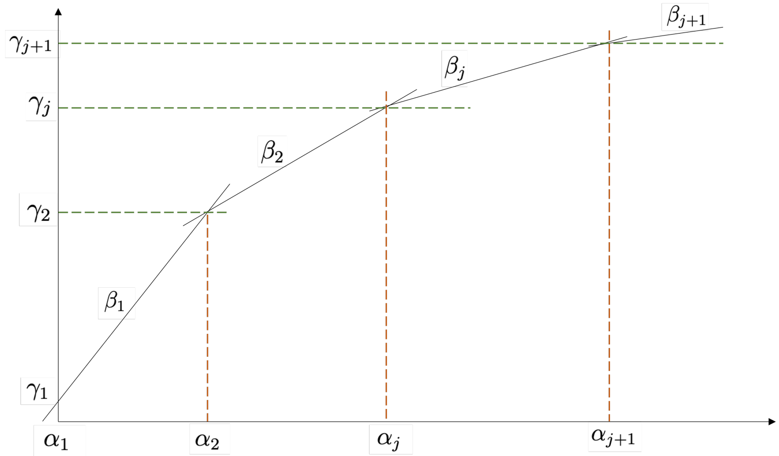

Figure 3: Definition of α_k^j, β_k^j and γ_k^j

We solve the 2LULS in the same way as Melo and Wolsey (2010). Details can be found in Algorithm 1. First, we initialize $H(u,t),\ 1 \leq u \leq t \leq |T|$. Then, we retrieve ϕ^j, taking into account the forbidden periods. We also introduce free periods, i.e, periods where the warehouse does not have to pay the production setup cost. Laslty, we compute $G(t)$ and $\Omega(t)$ for all $t \in T$.

Algorithm 1: Dynamic programming for the two-level uncapacitated lot-sizing problem with forbidden periods

Def `DP`(c, Y, F)**:**
- **for** $u \in [1,T]$ **do**
  - **for** $t \in [u,T]$ **do**
    - $H(u,t) \leftarrow \min_{u \leq j \leq t}(H(u,j-1) + f_j^1 + \tilde{p}_j^1 d_{jt})$;
- $\phi^1 \leftarrow (0, \tilde{p}_1^0, f_1^0)$;
- **for** $j \in [1, |T|-1]$ **do**
  - **if** $j \in F$ **then**
    - Find the breakpoints between ϕ^j and $b \mapsto \tilde{p}_{j+1}^0 b$;
    - Retrieve ϕ^{j+1} ;
  - **else if** $j \in Y$ **then**
    - Find the breakpoints between ϕ^j and $b \mapsto f_{j+1}^0 + \tilde{p}_{j+1}^0 b$;
    - Retrieve ϕ^{j+1} ;
  - **else**
    - $\phi^{j+1} \leftarrow \phi^j$;
- Retrieve $\chi_{j,t}$ and $\kappa_{j,t}$, $1 \leq j \leq t \leq |T|$;
- $G(0) \leftarrow 0$, $\Omega(0) \leftarrow \varnothing$;
- **for** $t \in T$ **do**
  - $G(t) \leftarrow \min_{1 \leq j \leq t}\{G(j-1) + \chi_{j,t} + H(j,t)\}$;
  - $j \leftarrow \operatorname{argmin}_{1 \leq j \leq t}\{G(j-1) + \chi_{j,t} + H(j,t)\}$;
  - $\Omega(t) \leftarrow \Omega(j-1) \cup \{\kappa_{j,t}\}$;
- **return** $G(|T|), \Omega(|T|)$;

We can then solve the 2LULS for all the retailers and join the results. In Algorithm 2, we solve the 2LULS for each retailer. For each period, we count the number of retailers that induce a production period. We retrieve the excessive production setup costs. Algorithm 2 presents the details. In Algorithm 2, `obj[c]` denotes the dynamic programing objective value $G(|T|)$ result for retailer c, `X[c]` denotes the dynamic result set result $\Omega(|T|)$ for retailer c, and `orders[t]` counts the number of production setups at period t.

Algorithm 2: Simple heuristic for the OWMR with fixed production periods

```
Def SimpleHeuristic(c):
    obj, X ← [] ·|R|, [[ ]]·|R| ;
    orders ← [ ] ·|T| ;
    for c ∈ R do
        obj[c], X[c] ← DP(c, Y, ∅) ;
        for i ∈ X[c] do
            orders[i] ← orders[i]+1
    res ← sum(obj[c] for c ∈ R) - sum(f_t^0· max(0, orders[t]-1) for t ∈ T) ;
    return res;
```

We introduce an additional feature that allows for the identification of periods when certain retailers place orders while others do not. During these periods with varied ordering behaviour, we designate them as "free periods." In these free periods, retailers who did not place orders can explore the option of generating production at such free periods to optimize their operations.

Algorithm 3 presents the details. We solve the 2LULS for each retailer. We then count the number of retailers that induce a production period for each period. If a period gets between 0 and $\frac{|R|}{2}$ orders at the warehouse level, we add this period to the set of free periods. We retrieve the excessive production setup costs and add the production set-up for free periods.

Algorithm 3: Heuristic for the OWMR with fixed production periods

```
Def Heuristic(c):
    obj, X ← [ ]·|R|, [[ ]]·|R| ;
    orders ← [ ], condition ← True, F ← ∅, compt ← 1 ;
    while condition and compt ≤ 10 do
        for c ∈ R do
            obj[c], X[c] ← DP(c, Y, F) ;
            for i ∈ X[c] do
                orders[i] ← orders[i]+1
        condition ← False ;
        for t ∈ T do
            if orders[t] > 0 and orders[t] < |R|/2 then
                Add t to F ;
                Condition ← True, compt ← compt+1 ;
        P ← X \ F ;
        res ← sum(obj[c] for c ∈ R) - sum(f_t^0· max(0, orders[t]-1) for t ∈ P)+ sum(f_t^0·
          min(1, orders[t]) for t ∈ F) ;
    return res;
```

4.3 Results

4.3.1 • Exact methods

We ran all 8 formulations. We consider our methods alone and with the additional echelon stock constraints (ESC) (21). This means that when +ESC is written, we added the echelon stock constraints to the formulation. We ran the instances with $X = \{1 + \tau \cdot k;\ k \in \mathbb{N}\}$ with $\tau \in \{3, 4, 5\}$. For the echelon stock constraints, we took parameters $\overline{h^0} = 0.5$, $\overline{f^0} = 4500$, $\overline{f} = 100$ and $\theta = \tau - 1$.

In Table 3 and Table 4, the results are aggregated for every formulation. Table 3 represents the average running time. Fixing the y-variables is the most efficient way to impose fixed production periods while considering a naive value of M_t is the least efficient. One can see that the effect of additional echelon stock constraints is not constant: it can be positive for some methods (naive value of M_t) and harmful for others (fixing the y variables). This can be explained by a tradeoff between the tightness of the formulation's relaxation and the complexity of the formulation, as the ESC increase the complexity of methods already efficient on their own. Indeed, adding the echelon stock constraints adds $\mathcal{O}(|T|)$ constraints. The interested reader can find a graph representing this data in the appendix (Figure 4).

$\|T\|$	$\|R\|$	Exact methods: $\tilde{M}_t$	$\tilde{M}_t$ +ESC	fix y (13)	fix y (13) +ESC	∞	∞ +ESC	ESC	Mean
15	50	0.30	0.32	**0.18**	0.32	0.70	0.32	0.32	0.35
	75	0.46	0.52	**0.28**	0.52	1.15	0.52	0.52	0.57
	100	0.73	0.85	**0.44**	0.85	1.52	0.86	0.84	0.87
20	50	0.70	0.74	**0.38**	0.74	1.26	0.74	0.74	0.76
	75	1.14	1.33	**0.63**	1.31	1.97	1.32	1.30	1.29
	100	1.69	2.09	**0.96**	2.07	2.78	2.08	2.05	1.96
25	50	1.45	1.57	**0.76**	1.57	1.98	1.56	1.55	1.49
	75	2.51	2.94	**1.32**	2.90	3.71	2.96	2.87	2.75
	100	3.76	4.70	**1.98**	4.61	4.88	4.72	4.61	4.18
Mean		1.42	1.67	**0.77**	1.65	2.22	1.68	1.64	1.58

Table 3: Solving time in seconds (ESC = Echelon Stock constraints)

Table 4 represents the cost repartition in the optimal solutions. An analysis of the cost repartition indicates that when the number of forbidden production periods increases, the setup costs (and the number of setups) at the retailers decrease while the holding costs at the warehouse increase. The retailers adapt their orders' frequency to the warehouse.

τ	$\lvert X\rvert$	Setup costs (%)		Holding costs (%)		Total costs
		Retailers	Warehouse	Retailers	Warehouse	
3	16	30	17	39	14	161,773
4	18	25	11	39	25	183,693
5	19	18	7	40	35	213,822

Table 4: Repartition of the total costs ($|T| = 25, |R| = 100$)

Table 5 represents how the formulations yield different results with each method. All formulations, except OWMR, give better results with the method fixing the y variables (13). Some formulations give similar results for all formulations, this is the case for formulations ES, OWMR, TP and TPC (the two latter with the exception of the $M_t = \infty$ method). The SES and SP formulations yield very bad results and are outperformed by the simple formulation OWMR. Also, the formulation MC yields very heterogenous results as adding the echelon stock constraints increases significantly its solving time. The interested reader can find a graph representing this data in the appendix (Figure 5).

	$\tilde{M}_t$		fix y (13)		∞		ESC	Mean
		+ESC		+ESC		+ESC		
ES	2.65	2.42	**1.85**	2.45	2.73	2.44	2.46	2.43
MC	1.15	8.33	**0.54**	7.79	1.73	8.26	7.54	5.05
OWMR	7.59	1.99	2.36	1.94	2.23	1.95	**1.90**	2.85
SES	8.80	7.34	**6.86**	7.20	8.62	7.50	7.54	7.70
SP	3.33	10.58	**1.59**	10.42	3.92	10.62	10.43	7.27
SPC	3.14	4.44	**1.33**	4.46	4.41	4.40	4.48	3.81
TP	1.82	1.59	**0.71**	1.65	7.76	1.63	1.61	2.40
TPC	1.59	0.93	**0.62**	0.96	7.68	0.93	0.93	1.95
Mean	3.76	4.70	**1.98**	4.61	4.88	4.72	4.61	4.18

Table 5: Solving time (in seconds) ($|T| = 25,\ |R| = 100$)

4.3.2 • Heuristic

We compare our heuristic with the fastest formulation known (MC) coupled with the method of fixing the y variables. We computed our test with $|T| \in \{40, 50, 60\}$ and $|R| \in \{50, 75, 100\}$. We selected our periods randomly with $X = \{1\} + \{\sigma_k\}$ where $\{\sigma_k,\ k \in [1, \tau - 1]\}$ is a random set of $\tau - 1$ items in $[2, |T|]$. We took parameter $k \in \{4, 5, 6\}$ such that $\tau = \frac{|T|}{k}$. In this setting, we cannot easily use the echelon stock constraints method or the method with $f_t^0 = M_t,\ \forall t \in X$.

Table 6 presents the results aggregated for each value of parameter k. The column Gap is calculated such that if we note v the solution value of formulation MC and v' the solution value for a heuristic, we have Gap $= 100 \cdot \frac{|v-v'|}{v}$. We can see that the heuristics do not perform well:

the solving time is higher than the formulation MC with the method of fixing the variables y. However, the optimality gap is on average below 200%. Yet, this is not true for all instances taken individually.

$\lvert T\rvert$	$\lvert R\rvert$	MC	Simple Heuristic		Heuristic	
		Time	Time	Gap (%)	Time	Gap (%)
40	50	0.69	0.69	170	0.73	151
	75	1.03	1.02	173	1.11	152
	100	1.41	1.35	179	1.48	152
50	50	1.08	1.27	166	1.37	147
	75	1.66	1.88	175	2.04	153
	100	2.27	2.53	175	2.72	150
60	50	1.63	2.12	172	2.28	149
	75	2.54	3.17	170	3.41	149
	100	3.47	4.00	177	4.35	147

Table 6: Solving time (in seconds) and optimality gap for the heuristics

Table 7 presents the evolution of the solving time with the parameter k. The accuracy of the heuristics increases with parameter k. This is not surprising as the set of authorized production periods Y decreases with k.

$\lvert R\rvert$	k	MC	Simple Heuristic		Heuristic	
		Time	Time	Gap (%)	Time	Gap (%)
50	4	1.66	2.20	190	2.40	163
	5	1.73	1.96	169	2.10	145
	6	1.48	2.21	156	2.35	140
75	4	2.62	3.29	188	3.58	165
	5	2.68	2.93	172	3.14	149
	6	2.31	3.28	147	3.51	133
100	4	3.52	4.06	201	4.47	162
	5	3.72	3.92	173	4.21	143
	6	3.14	4.04	155	4.38	135

Table 7: Solving time (in seconds) and optimality gap for the heuristics, evolution with parameter k ($\lvert T\rvert = 60$)

5 LIMITATIONS ON THE PRODUCTION PERIODS

We now study cases where the production periods are not known in advance. We introduce limitations concerning the maximum total number of periods. Then, we study limitations concerning the number of periods between two production periods: when there is a maximum gap to respect (i.e., two consecutive periods have to be close to each other) and when there is a minimum gap to respect (i.e., two consecutive periods have to be far from each other).

5.1 Total max

In this first scenario, we do not want more than n orders at the warehouse over our finite horizon of $|T|$ periods. This constraint could prove beneficial for supply chains aiming to distribute their orders over time, enabling the balancing of workers' schedules other the weeks, among other advantages. We have two ways to incorporate the new constraints: one way by summing the number of constraints over the interval T and another way by keeping count of the number of production periods used so far.

5.1.1 • Total sum

We add the following constraint:

$$\sum_{t=1}^{|T|} y_t^0 \leq n \tag{27}$$

Constraint (27) counts the number of periods over the interval T and introduces an upper bound on this number. We call $OWMR^1_{TS}$ the problem (OWMR.1) - (OWMR.6), (27).

5.1.2 • Consecutive count of setups

We introduce a variable $v_t,\ t \in T$ which counts the number of setups so far ($\forall i \leq t$). We have the following constraints:

$$v_1 = 1 (= y_t^0) \tag{28}$$
$$v_t - v_{t-1} = y_t, \quad \forall t \in [2, |T|] \tag{29}$$
$$v_{|T|} \leq n \tag{30}$$
$$v_t \in \mathbb{N} \tag{31}$$

We call $OWMR^1_{CS}$ the problem (OWMR.1) – (OWMR.6), (28) – (31).

Lemma 4. $v(OWMR^1_{TS}) = v(OWMR^1_{CS})$

Sketch of the proof. Let $(y, x, s) \in F(OWMR^1_{TS})$. We build v_t such that

$$v_1 = 1 (= y^0_t)$$
$$v_t - v_{t-1} = y_t \ \ \forall t \in [2, |T|]$$

Then we have

$$v_{|T|} = \sum_{t=2}^{|T|} (v_t - v_{t-1}) + v_1 = \sum_{t=1}^{|T|} y^0_{|T|} \leq n$$

So $(y, x, s, v) \in F(OWMR^1_{CS})$.
Let $(y, x, s, v) \in F(OWMR^1_{CS})$.

$$n \geq v_{|T|} = \sum_{t=2}^{|T|} (v_t - v_{t-1}) + v_1 = \sum_{t=1}^{|T|} y^0_{|T|}$$

So $(y, x, s) \in F(OWMR^1_{TS})$. The linear relaxation objective value is then the same for the two methods. □

The consecutive method uses $\mathcal{O}(|T|)$ integer variables more than the total sum method. The total sum method adds one constraint while the consecutive sum method adds $\mathcal{O}(|T|)$ constraints.

5.2 Maximum gap

We require consecutive orders to be sufficiently close together; we do not permit having a gap of n periods without any orders at the warehouse. This is useful for goods that are perishable and need to be replenished frequently. We introduce two ways to add such constraints: the first method is summing the number of periods on consecutive intervals of n periods, making sure we have at least one period for each interval. The second method counts the number of periods since the last ordering period and makes sure this number does not exceed n.

5.2.1 • First method: total sum

We add the following constraints:

$$\sum_{k=t}^{t+n-1} y^0_k \geq 1 \quad \forall t \in [1, |T| - n + 1] \tag{32}$$

Constraints (32) count the number of production periods over each interval on n consecutive periods and ensure that each interval contains at least one ordering period. We call $OWMR^2_{TS}$ the problem (OWMR.1) – (OWMR.6), (32).

5.2.2 • Second method: consecutive sum

We add a new variable $v_t \in \mathbb{N}$ with the following condition:

$$v_t - v_{t-1} = 1 \text{ if } y_t^0 = 0$$
$$v_t = 1 \text{ if } y_t^0 = 1$$

So this means:

$$v_t - v_{t-1} \cdot (1 - y_t^0) = 1$$

The variable v counts the number of periods since the last setup y_t^0. We have to add an additional constraint z_t to linearize these constraints such that

$$z_t = v_{t-1} \cdot (1 - y_t^0)$$

We therefore consider the following constraints:

$$z_t \leq (1 - y_t^0) \cdot |T| \tag{33}$$
$$z_t \leq v_{t-1} \tag{34}$$
$$z_t \geq v_{t-1} - y_t^0 \cdot |T| \tag{35}$$
$$0 \leq v_t \leq n \tag{36}$$
$$v_t = 1 + z_t \tag{37}$$
$$v_0 = 0 \tag{38}$$
$$v, z \in \mathbb{N} \tag{39}$$

Table 8 presents the values of variables y, v and z for one instance.

t	0	1	2	3	4	5	6	7	8
y	-	1	0	0	1	0	0	1	0
v	0	1	2	3	1	2	3	1	2
z	-	0	1	2	0	1	2	0	1

Table 8: Values of v and z given y values

We call $OWMR_{CS}^2$ the problem (OWMR.1) – (OWMR.6), (33) – (39).

Proposition 9. $v(OWMR_{TS}^2) \geq v(OWMR_{CS}^2)$

Let $(y, x, s) \in F(OWMR_{TS}^2)$. We definite z and v as follows:

$$v_0 = 0 \tag{40}$$
$$z_t = \max(0, v_{t-1} - y_t^0 \cdot |T|) \tag{41}$$
$$v_t = 1 + z_t \tag{42}$$

We want to show that z and v are feasible for $OWMR_{CS}^2$. We have:

(33) if $z_t = v_{t-1} - y_t^0 \cdot |T|$ then $v_t = 1 + v_{t-1} - y_t^0 \cdot |T| \leq 1 + v_{t-1}$ so $v_t \leq |T|$ because we have $|T|$ consecutive constraints thus $z_t = v_{t-1} - y_t^0 \cdot |T| \leq (1 - y_t^0) \cdot |T|$. If $z_t = 0$, the inequality is also true,

(34) if $z_t = v_{t-1} - y_t^0 \cdot |T| \leq v_{t-1}$ else $z_t = 0 \leq v_{t-1}$ ($z_t \geq 0$ so $v_t \geq 1$),

(35) $z_t \geq v_{t-1} - y_t^0 \cdot |T|$ by construction,

(36) $v_t \geq 0$ by construction,

(37) $v_t = 1 + z_t$ by construction,

(38) $v_0 = 0$ by construction.

Now we want to check that $v_t \leq n$ for all t.

- For $t \leq n$, we have that $v_t \leq 1 + v_{t-1}$ thus $v_t \leq t \leq n$
- For $t > n$, we establish the following properties, with (z, v) defined with (40) – (42) and y verifying (32):

Lemma 5. *If $v_{t-n} \leq n$ and for all $k \in [t-n+1, t]$, $z_k > 0$, then $v_t \leq n$*

Sketch of the proof. We suppose that for all $k \in [t-n+1, t]$, $z_k > 0$ so $z_k = v_{k-1} - y_k^0 \cdot |T|$ thus $v_k = 1 + v_{k-1} - y_k^0 \cdot |T|$. Therefore:

$$\begin{aligned} \sum_{k=t-n+1}^{t} (v_k - v_{k-1}) &= v_t - v_{t-n} \\ &= n - |T| \sum_{k=t-n+1}^{t} y_k^0 \leq n - |T| \end{aligned}$$

Thus:

$$v_t \leq n - |T| + v_{t-n} \leq 2n - |T| \leq n \quad \text{since } |T| \geq n$$

□

Lemma 6. *If there exists $k \in [t-n+1, t]$ such that $z_k = 0$, then $v_t \leq n$.*

Sketch of the proof. We know that:

$$v_i = 1 + z_i \leq v_{i-1} + 1, \quad \forall i \in T \tag{43}$$

Then:

$$v_t - v_k = \sum_{j=k+1}^{t} (v_j - v_{j-1}) \leq^{(43)} \sum_{j=k+1}^{t} 1 = t - k \leq n$$

□

By recurrence, for all $t \in T$, we have $v_t \leq n$. So if $(y, s, x) \in F(OWMR^1_{TS})$ there exists $v,\ t$ such that $(y, s, x, v, t) \in F(OWMR^2_{CS})$. So $v(OWMR^2_{TS}) \geq v(OWMR^2_{CS})$.

Case of strict inequality:
We take the following instance: $R = \{1\}$, $T = \{1, 2, 3\}$, $n = 2$, $d^1 = (22, 90, 60)$, $h^0 = 0.5$, $h^1 = 0.83$, $f^0 = (2655, 3349, 1527)$, $f^1 = (97, 45, 10)$. We have $v(OWMR^2_{TS}) = 4373.7$ and $v(OWMR^2_{CS}) = 3385.7$.

5.3 Minimum gap

We require consecutive orders to be sufficiently far away. For each interval of n periods, there is at most one order at the warehouse. This limitation can be useful for supply chains where there needs to be a cool-down period at production, for example for machines that need to be cleaned or tend to produce a lot of heat. This can also be useful for workers who need a rest period after dealing with production.

We introduce two ways to add such constraints: the first method counts the number of production periods on n consecutive periods and makes sure this number does not exceed 1. The second method counts the number of periods since the last production setup. When production occurs, it makes sure n periods have elapsed since the last production period.

5.3.1 • First method: total sum

We add the following constraint:

$$\sum_{k=t}^{t+n-1} y^0_k \leq 1 \quad \forall t \in [1, |T| - n] \tag{44}$$

Constraints (44) count the number of production periods over each interval on n consecutive periods and ensure that each interval contains at least one ordering period.

5.3.2 • Second method: consecutive sum

We had a new variable v_t with the following condition:

$$\begin{aligned} v_t - v_{t-1} = -1 \text{ if } y^0_t = 0 \\ v_t = |T| \text{ if } y^0_t = 1 \end{aligned}$$

So this means:

$$v_{t-1} \cdot (1 - y^0_t) - v_t = 1 - (|T| + 1) \cdot y^0_t \tag{45}$$

We also create the variable $z_t = v_{t-1} \cdot (1 - y^0_t)$ and we have $v_0 = 0$.
We then have:

$$z_t - v_t = 1 - (|T| + 1) \cdot y^0_t \tag{46}$$

The significance lies in the value of v_t before a production setup. We want the following property ($y_t = 1 \Rightarrow v_{t-1} \leq |T| - n + 1$) to be true so that at least n periods have passed between two consecutive setups. Since y_t is a binary variable and using the definition of z, $y_t^0 \cdot v_{t-1} = v_{t-1} - z_t \leq |T| - n + 1$. So we must have $v_{t-1} - z_t \leq T - n + 1$. We therefore consider the following constraints:

$$
\begin{aligned}
z_t &\leq (1 - y_t^0) \cdot |T| && (47)\\
z_t &\leq v_{t-1} && (48)\\
z_t &\geq v_{t-1} - y_t^0 \cdot |T| && (49)\\
v_t &= z_t + (|T| + 1) \cdot y_t^0 - 1 && (50)\\
v_0 &= 0 && (51)\\
v_{t-1} - z_t &\leq |T| - n + 1 && (52)\\
v, z &\in \mathbb{N} && (53)
\end{aligned}
$$

Table 9 presents the values of variables y, v and z for one instance.

t	0	1	2	3	4	5	6	7	8
y	-	1	0	0	1	0	0	0	1
v	0	T	T-1	T-2	T	T-1	T-2	T-3	T
z	-	0	T	T-1	0	T	T-1	T-2	0
$v_{t-1} - z_t$	-	0	0	0	T-2	0	0	0	T-3

Table 9: Values of v and z given y values ($T \equiv |T|$ by abuse of notation)

5.4 Results

Since the formulations OWMR, ES and SES are significantly slower than the five other formulations, we performed our tests with only the five fastest formulations (SP, SPC, TP, TPC and MC) for $|T| \in \{15, 20, 25\}$ and $|R| \in \{50, 75, 100\}$.

5.4.1 • Total max

In Table 10 and Table 11 the results are aggregated for the five formulations and static and dynamic instances. We discuss here the solving time of our methods and managerial insights.

Table 10 presents the average solving time for the consecutive sum method and the total sum methods. Both methods yield similar results. Table 11 presents the evolution of the solving time when we authorize fewer production periods. For each instance, we note δ the number of production periods in the optimal solution. If the removed periods' parameter is equal to 8, it means we authorize a maximum of $n = \delta - 8$ periods in total for each instance. One can note that there is a small inversion as the number of removed periods increases (for one removed period, the consecutive sum method performs slightly better, whereas for 8 removed periods the total sum method performs slightly better).

$\lvert T\rvert$	$\lvert R\rvert$	Consecutive sum	Total sum
15	50	0.54	0.54
	75	0.95	0.94
	100	1.68	1.66
20	50	1.53	1.51
	75	2.98	2.94
	100	5.16	5.15
25	50	3.80	3.74
	75	8.05	7.94
	100	12.87	12.86

Table 10: Solving time in seconds

Removed periods	Consecutive sum	Total sum
1	4.64	4.80
2	5.62	5.73
3	6.68	6.83
4	9.13	9.16
5	11.83	11.85
6	15.65	15.55
7	21.24	20.92
8	28.16	28.05

Table 11: Solving time in seconds ($\lvert T\rvert = 25,\ \lvert R\rvert = 100$)

Table 12 analyses the cost repartition and indicates that when we decrease the number of authorized production periods, the setup costs (and the number of setups) at the retailers decrease while the holding costs at the warehouse increase, similarly to the case with fixed production periods.

Removed	Setup costs (%)		Holding costs (%)		Total costs
periods	Retailers	Warehouse	Retailers	Warehouse	
1	39	22	32	7	144,318
2	37	20	34	9	145,694
3	35	17	37	11	148,034
4	32	16	39	13	151,807
5	30	13	40	17	158,031
6	27	11	41	21	167,646
7	24	9	40	27	182,074
8	20	7	38	35	202,631

Table 12: Repartition of the total costs ($\lvert T\rvert = 25,\ \lvert R\rvert = 100$)

5.4.2 • Maximum gap between two consecutive ordering periods

We ran our instances imposing a total max parameter $n \in \{1, 2, 3\}$. Table 13 represents the average solving time for the total sum method and the consecutive sum method. The consecutive sum method performs slightly worse than the total sum method. Table 14 represents the evolution of the solving time with the parameter n. Both methods' solving times increase with n, yet the total sum method increases relatively more.

$\|T\|$	$\|R\|$	Total sum	Consecutive sum
15	50	0.25	0.26
	75	0.37	0.38
	100	0.58	0.6
20	50	0.65	0.72
	75	1.12	1.21
	100	1.63	1.75
25	50	1.31	1.46
	75	2.26	2.44
	100	3.47	3.63

Table 13: Solving time in seconds

$\|R\|$	n	Total sum		Consecutive sum	
		Time	Increase (%)	Time	Increase (%)
50	1	1.12	-	1.1	-
	2	1.41	125	1.31	119
	3	1.85	164	1.53	139
75	1	1.74	-	1.7	-
	2	2.61	150	2.47	145
	3	2.96	170	2.6	153
100	1	2.35	-	2.31	-
	2	3.99	170	3.8	164
	3	4.56	194	4.28	185

Table 14: Solving time (in seconds) evolution with parameter n ($|T| = 25$)

5.4.3 • Minimum gap between two consecutive ordering periods

We ran our instances imposing a total max parameter $n \in \{1, 2, 3\}$. Table 15 represents the average solving time for the total sum method and the consecutive sum method. The consecutive sum method performs significantly worse than the total sum method. Table 16 represents the evolution of the solving time with the parameter n. Both methods' solving times increase with n, yet the consecutive sum method increases relatively significantly more.

$\|T\|$	$\|R\|$	Total sum	Consecutive sum
15	50	0.29	0.35
	75	0.46	0.62
	100	0.78	1.13
20	50	0.71	1.16
	75	1.29	2.32
	100	2.09	4.19
25	50	1.48	2.97
	75	3.04	6.85
	100	5.54	13.71

Table 15: Solving time in seconds

$\|R\|$	n	Total sum		Consecutive sum	
		Time	Increase (%)	Time	Increase (%)
50	2	1.24	-	1.38	-
	3	1.41	114	2.74	198
	4	1.78	144	4.79	347
75	2	2.32	-	2.68	-
	3	3.01	130	7.35	275
	4	3.78	163	10.53	393
100	2	4.33	-	6.35	-
	3	5.36	124	14.6	230
	4	6.95	160	20.19	318

Table 16: Solving time evolution with parameter n ($|T| = 25$)

6
CONCLUDING REMARKS

We have addressed the OWMR with production constraints and introduced several restrictions on the production periods. The first type of constraint consists of fixing in advance the forbidden periods. To solve these constraints, we introduced different methods. A first method fixes the variables y^0, another method modifies the production setup parameter f^0 and the last method introduces echelon stock constraints. We also developed a heuristic based on a dynamic program solving the uncapacitated two-level lot-sizing problem. The second type of constraint concerns limitations on the production periods. The first limitation limits the total number of production periods. The other limitations impose lower or upper bounds on the number of periods between two consecutive ordering periods. For these limitations, we introduced several methods to implement the constraints. We computed extensive computational tests as well as elements of proof for the validity and efficiency of our methods.

One suggestion for further investigation is to explore the different possibilities for the heuristic. A starting point might be to explore managerial insights on the additional costs for the heuristic. An interesting information to study would be the repartition of costs in the optimal solution versus in the heuristic. Where does the increase of cost in the heuristic take place? Are the holding costs at the warehouse bigger or is it instead the setup costs at the retailers level? Also, one of the problems encountered by our method is that the retailers alone tend to order less often than when they regroup together. Indeed, production periods that are considered for the optimal solution will be overlooked by every retailer. Another idea is to implement a metaheuristic. Lastly, it would perhaps be effective to compute $H(u,t)$ with the fastest known algorithm in $\mathcal{O}(|T|\log(|T|)$, whereas we used an algorithm in $\mathcal{O}(|T|^2)$ time.

Other ideas to pursue would be to prove the hierarchy between the different formulations for the minimum gap problem. Lastly, one idea to investigate is the quality of the upper found on the parameter M_t used to change the production setup costs f_t^0. There are two possibilities: our bound is tight or there exists a tighter bound.

REFERENCES

Arkin, E., Joneja, D. and Roundy, R. (1989), ‘Computational complexity of uncapacitated multi-echelon production planning problems’, *Operations research letters* **8**(2), 61–66.

Barany, I., Van Roy, T. and Wolsey, L. A. (1984), ‘Uncapacitated lot-sizing: The convex hull of solutions’.

Brahimi, N., Absi, N., Dauzère-Pérès, S. and Nordli, A. (2017), ‘Single-item dynamic lot-sizing problems: An updated survey’, *European Journal of Operational Research* **263**(3), 838–863.

Cunha, J. O. and Melo, R. A. (2016), ‘On reformulations for the one-warehouse multi-retailer problem’, *Annals of Operations Research* **238**, 99–122.

Eppen, G. D. and Martin, R. K. (1987), ‘Solving multi-item capacitated lot-sizing problems using variable redefinition’, *Operations Research* **35**(6), 832–848.

Federgruen, A. and Tzur, M. (1999), ‘Time-partitioning heuristics: Application to one warehouse, multiitem, multiretailer lot-sizing problems’, *Naval Research Logistics (NRL)* **46**(5), 463–486.

Gayon, J.-P., Massonnet, G., Rapine, C. and Stauffer, G. (2017), ‘Fast approximation algorithms for the one-warehouse multi-retailer problem under general cost structures and capacity constraints’, *Mathematics of Operations Research* **42**(3), 854–875.

Krarup, J. and Bilde, O. (1977), ‘Plant location, set covering and economic lot size: An 0 (mn)-algorithm for structured problems’, *Numerische Methoden bei Optimierungsaufgaben Band 3: Optimierung bei graphentheoretischen und ganzzahligen Problemen* pp. 155–180.

Levi, R., Roundy, R., Shmoys, D. and Sviridenko, M. (2008), ‘A constant approximation algorithm for the one-warehouse multiretailer problem’, *Management Science* **54**(4), 763–776.

Melo, R. A. and Wolsey, L. A. (2010), ‘Uncapacitated two-level lot-sizing’, *Operations Research Letters* **38**(4), 241–245.

Pochet, Y. and Wolsey, L. A. (2006), *Production planning by mixed integer programming*, Vol. 149, Springer.

Schwarz, L. B. (1973), ‘A simple continuous review deterministic one-warehouse n-retailer inventory problem’, *Management science* **19**(5), 555–566.

Solyalı, O. and Süral, H. (2012), ‘The one-warehouse multi-retailer problem: Reformulation, classification, and computational results’, *Annals of Operations Research* **196**, 517–541.

Wagelmans, A., Van Hoesel, S. and Kolen, A. (1992), ‘Economic lot sizing: An o (n log n) algorithm that runs in linear time in the wagner-whitin case’, *Operations Research* **40**(1-supplement-1), S145–S156.

Wagner, H. M. and Whitin, T. M. (1958), ‘Dynamic version of the economic lot size model’, *Management science* **5**(1), 89–96.

Yang, W., Chan, F. T. and Kumar, V. (2012), ‘Optimizing replenishment polices using genetic algorithm for single-warehouse multi-retailer system’, *Expert systems with applications* **39**(3), 3081–3086.

7 APPENDICES

7.1 Results - no additional constraints

$\|T\|$	$\|R\|$	d_t^c	f_t^0	ES Time	Nodes	MC Time	Nodes	OWMR Time	Nodes	SES Time	Nodes	SP Time	Nodes	SPC Time	Nodes	TP Time	Nodes	TPC Time	Nodes	Mean Time	Mean Nodes
20	50	D	D	6.7	1029.10	0.4	0.00	76.0	3722.90	5.2	342.30	0.8	0.00	0.8	0.00	0.6	0.00	0.5	0.00	11.4	636.79
			S	12.3	1527.80	0.4	0.00	89.3	3105.20	5.1	312.40	0.8	0.00	0.7	0.00	0.5	0.00	0.5	0.00	13.7	618.18
		S	D	23.5	1804.20	0.5	0.30	133.7	3956.80	5.8	503.50	0.9	0.30	0.9	0.30	0.6	0.30	0.5	0.30	20.8	783.25
			S	20.1	1263.90	0.4	0.00	99.7	3192.60	5.3	394.90	0.8	0.00	0.8	0.00	0.5	0.00	0.5	0.00	16.0	606.43
	75	D	D	117.8	2421.90	0.9	0.30	198.3	3495.90	13.5	927.90	1.5	0.30	1.3	0.30	1.1	0.30	1.0	0.30	41.9	855.90
			S	156.0	2669.90	1.0	0.30	295.1	3714.60	13.7	962.20	1.5	0.00	1.4	0.60	1.2	0.30	0.9	0.00	58.8	918.49
		S	D	129.4	2393.20	0.9	0.30	213.6	3097.50	13.4	940.20	1.8	0.30	1.4	0.30	1.1	0.30	1.0	0.30	45.3	804.05
			S	194.1	2739.20	1.1	0.60	274.2	4232.30	19.4	1481.00	1.9	0.60	1.5	0.60	1.3	0.90	1.2	0.60	61.8	1056.98
	100	D	D	333.7	2212.20	1.6	0.60	289.3	3380.70	29.2	1509.40	2.2	0.60	1.9	0.60	1.8	0.60	1.6	0.60	82.6	888.16
			S	491.1	2700.00	1.6	0.60	393.4	3878.20	37.4	2138.10	2.1	0.60	1.8	0.60	1.8	0.60	1.6	0.60	116.4	1089.91
		S	D	479.7	2886.00	1.6	0.30	284.7	3516.20	31.3	1734.40	2.2	0.30	1.9	0.30	1.8	0.30	1.6	0.30	100.6	1017.26
			S	297.2	2470.10	1.5	0.00	274.3	3125.70	25.3	1411.50	1.9	0.00	1.6	0.00	1.4	0.00	1.3	0.00	75.6	875.91
25	50	D	D	115.9	2679.30	0.9	0.00	831.8	12546.10	23.4	1753.90	1.7	0.00	1.7	0.00	1.1	0.00	1.0	0.00	122.2	2122.41
			S	112.6	2263.10	0.8	0.00	726.2	14507.50	21.1	1622.90	1.6	0.00	1.6	0.00	1.1	0.00	1.0	0.00	108.2	2299.19
		S	D	88.1	2022.60	0.8	0.20	853.7	12418.50	15.1	1196.70	2.0	0.00	2.0	0.30	1.3	0.00	1.0	0.10	120.5	1954.80
			S	125.6	2639.80	0.8	0.00	1072.8	17033.80	20.7	1480.50	1.7	0.00	1.7	0.00	1.0	0.00	0.9	0.00	153.1	2644.26
	75	D	D	578.1	3298.70	1.6	0.00	1261.5	12117.10	81.9	4493.10	2.9	0.30	2.8	0.50	1.8	0.00	1.7	0.30	241.5	2488.75
			S	735.6	3478.30	1.7	0.00	1489.2	14965.00	86.6	4339.10	2.9	0.00	2.7	0.00	2.0	0.00	1.6	0.00	290.3	2847.80
		S	D	687.9	3838.50	1.8	0.90	1693.3	14711.20	85.4	4444.80	3.4	0.90	3.3	0.90	2.4	0.90	2.1	0.90	309.9	2874.88
			S	595.9	3119.40	1.4	0.00	1292.5	12245.90	78.5	3887.60	2.9	0.00	2.6	0.00	1.9	0.00	1.8	0.00	247.2	2406.61
	100	D	D	2273.8	5573.00	2.6	0.00	1545.6	13463.50	197.5	7720.40	3.8	0.00	3.4	0.00	2.8	0.00	2.5	0.00	504.0	3344.61
			S	2399.0	5993.90	3.2	0.90	1494.8	13101.70	240.7	9750.50	6.2	1.10	4.4	1.10	4.0	0.90	3.5	0.90	519.5	3606.38
		S	D	1617.1	4895.80	3.5	1.10	1591.7	13083.70	196.7	7493.00	6.5	1.10	5.4	1.10	4.3	1.10	3.8	1.10	428.6	3184.75
			S	2187.4	5806.40	3.6	0.90	1643.5	15179.20	235.2	9056.70	5.6	0.90	4.6	1.10	3.8	0.90	3.8	0.90	510.9	3755.88

Table 17: Solving time in seconds and number of nodes

7.2 Results - fixed production periods

		R	50				75				100			
		demand	D		S		D		S		D		S	
		setup costs	D	S	D	S	D	S	D	S	D	S	D	S
ES	M	Time	1.04	1.01	1.05	1.14	1.92	1.87	1.87	1.96	2.74	2.61	2.68	2.59
		Nodes	61.17	70.17	61.87	77.53	94.70	81.53	90.13	105.57	107.87	92.70	110.97	82.90
	+ ESC	Time	0.93	0.99	0.94	1.01	1.73	1.73	1.77	1.79	2.52	2.36	2.51	2.30
		Nodes	59.77	63.90	56.97	62.43	77.87	78.30	88.00	84.17	98.70	83.87	104.27	76.13
	fix y	Time	0.75	0.76	0.72	0.77	1.26	1.24	1.29	1.36	1.90	1.77	1.93	1.80
		Nodes	76.40	74.07	67.37	76.37	93.43	90.20	92.17	109.47	109.33	85.07	118.67	92.23
	+ ESC	Time	0.93	0.99	0.94	1.02	1.74	1.74	1.78	1.81	2.55	2.39	2.53	2.34
		Nodes	59.77	63.90	56.97	62.43	77.87	78.30	88.00	84.17	98.70	83.87	104.27	76.13
	infinite	Time	1.06	1.06	1.07	1.14	2.02	1.91	1.92	1.97	2.84	2.70	2.73	2.65
		Nodes	75.23	83.47	74.47	89.50	118.10	102.43	112.20	110.27	132.60	116.27	127.67	110.60
	+ ESC	Time	0.93	0.99	0.94	1.02	1.73	1.74	1.78	1.80	2.53	2.38	2.52	2.34
		Nodes	59.77	63.90	56.97	62.43	77.87	78.30	88.00	84.17	98.70	83.87	104.27	76.13
	ESC	Time	0.94	1.00	0.95	1.02	1.74	1.74	1.78	1.80	2.56	2.39	2.54	2.34
		Nodes	59.77	63.90	56.97	62.43	77.87	78.30	88.00	84.17	98.70	83.87	104.27	76.13
MC	M	Time	0.49	0.50	0.49	0.50	0.81	0.81	0.81	0.82	1.17	1.15	1.13	1.15
		Nodes	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	+ ESC	Time	2.11	1.96	1.87	1.95	4.53	4.57	4.65	4.65	8.26	9.09	7.75	8.20
		Nodes	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	fix y	Time	0.25	0.25	0.25	0.25	0.40	0.40	0.40	0.40	0.55	0.53	0.54	0.54
		Nodes	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	+ ESC	Time	1.90	2.00	1.86	1.93	4.51	4.33	4.56	4.17	7.72	8.64	6.84	7.95
		Nodes	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	infinite	Time	0.71	0.71	0.72	0.73	5.36	4.86	4.69	4.56	1.78	1.79	1.67	1.68
		Nodes	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	+ ESC	Time	1.98	2.02	1.94	1.98	4.88	4.53	5.07	4.60	8.25	8.70	7.59	8.51
		Nodes	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	ESC	Time	2.01	1.83	1.83	1.88	4.32	4.23	4.35	4.10	7.49	8.42	6.90	7.34
		Nodes	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
OWMR	M	Time	2.96	3.09	2.84	3.17	5.36	5.21	4.81	5.86	7.30	7.51	7.65	7.91
		Nodes	848.27	1081.43	941.87	975.43	1331.63	1283.13	1128.43	1496.53	1264.53	1474.37	1371.47	1574.30
	+ ESC	Time	0.76	0.76	0.84	0.80	1.26	1.33	1.28	1.37	2.19	1.84	1.92	1.99
		Nodes	88.37	92.10	128.23	102.03	97.87	122.63	104.93	141.00	187.17	110.77	120.97	132.77
	fix y	Time	0.99	1.03	0.88	1.08	1.51	1.59	1.63	1.58	2.38	2.35	2.41	2.28
		Nodes	117.00	161.40	93.83	165.20	94.33	119.20	148.33	129.07	136.23	119.73	130.50	103.37
	+ ESC	Time	0.76	0.77	0.85	0.80	1.25	1.32	1.26	1.36	2.14	1.80	1.87	1.94
		Nodes	88.37	92.10	128.23	102.03	97.87	122.63	104.93	141.00	187.17	110.77	120.97	132.77
	infinite	Time	0.82	0.84	0.88	0.92	1.51	1.40	1.38	1.64	2.59	2.08	2.10	2.14
		Nodes	113.10	118.67	149.80	145.27	178.00	146.07	142.73	229.97	292.20	178.33	172.13	183.17
	+ ESC	Time	0.75	0.76	0.84	0.80	1.26	1.32	1.26	1.36	2.16	1.80	1.87	1.97
		Nodes	88.37	92.10	128.23	102.03	97.87	122.63	104.93	141.00	187.17	110.77	120.97	132.77
	ESC	Time	0.79	0.78	0.83	0.79	1.23	1.30	1.23	1.33	2.09	1.76	1.84	1.90
		Nodes	88.37	92.10	128.23	102.03	97.87	122.63	104.93	141.00	187.17	110.77	120.97	132.77
SES	M	Time	2.35	2.34	2.37	2.36	5.06	4.86	4.98	4.96	9.04	8.52	8.90	8.75
		Nodes	7.90	8.40	6.17	11.67	6.70	5.60	6.33	6.47	5.17	5.17	5.43	5.43
	+ ESC	Time	2.21	2.19	2.18	2.23	4.31	4.29	4.40	4.45	7.17	7.13	7.76	7.29
		Nodes	13.57	8.40	5.67	13.53	6.63	5.53	5.63	6.30	5.60	5.63	5.67	5.37
	fix y	Time	2.06	2.06	2.02	2.07	4.18	4.10	4.16	4.19	6.82	7.04	6.67	6.93
		Nodes	7.13	7.57	5.77	12.37	6.63	5.33	5.73	6.23	5.63	5.63	5.77	5.50
	+ ESC	Time	2.20	2.19	2.14	2.23	4.35	4.29	4.49	4.43	7.10	7.19	7.22	7.31
		Nodes	13.57	8.40	5.67	13.53	6.63	5.53	5.63	6.30	5.60	5.63	5.67	5.37
	infinite	Time	2.49	2.21	2.25	2.37	4.82	4.74	4.90	4.90	9.08	8.67	8.20	8.54
		Nodes	12.73	5.73	3.93	20.77	4.53	3.67	12.90	17.17	3.93	2.50	3.93	15.33
	+ ESC	Time	2.22	2.24	2.15	2.25	4.52	4.46	4.53	4.57	7.54	7.28	7.60	7.56
		Nodes	13.57	8.40	5.67	13.53	6.63	5.53	5.63	6.30	5.60	5.63	5.67	5.37
	ESC	Time	2.16	2.18	2.18	2.21	4.44	4.35	4.46	4.42	7.45	7.44	7.56	7.72
		Nodes	13.57	8.40	5.67	13.53	6.63	5.53	5.63	6.30	5.60	5.63	5.67	5.37

		R	50				75				100			
		demand	D		S		D		S		D		S	
		setup costs	D	S	D	S	D	S	D	S	D	S	D	S
SP	M	Time	1.55	1.51	1.52	1.53	2.36	2.32	2.44	2.33	3.33	3.35	3.32	3.30
		Nodes	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	+ ESC	Time	3.58	3.66	3.59	3.44	6.61	6.54	6.66	6.60	10.65	10.53	10.45	10.70
		Nodes	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	fix y	Time	0.77	0.75	0.79	0.81	1.18	1.18	1.17	1.17	1.58	1.58	1.60	1.60
		Nodes	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	+ ESC	Time	3.53	3.58	3.58	3.47	6.41	6.39	6.44	6.53	10.37	10.39	10.35	10.57
		Nodes	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	infinite	Time	1.71	1.71	1.80	1.67	2.55	2.54	2.58	2.59	4.03	3.82	4.31	3.52
		Nodes	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	+ ESC	Time	3.50	3.53	3.56	3.35	6.40	6.66	6.55	6.37	10.37	10.84	10.34	10.94
		Nodes	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	ESC	Time	3.51	3.55	3.59	3.39	6.44	6.40	6.41	6.39	10.42	10.59	10.34	10.37
		Nodes	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
SPC	M	Time	1.56	1.53	1.50	1.54	2.22	2.27	2.26	2.29	3.09	3.17	3.09	3.19
		Nodes	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	+ ESC	Time	1.98	2.01	2.00	1.99	3.06	3.08	3.13	3.17	4.45	4.43	4.44	4.44
		Nodes	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	fix y	Time	0.65	0.65	0.64	0.64	0.97	0.95	0.97	0.96	1.33	1.35	1.31	1.33
		Nodes	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	+ ESC	Time	2.00	1.99	1.98	2.00	3.09	3.11	3.15	3.14	4.49	4.46	4.47	4.44
		Nodes	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	infinite	Time	2.03	2.04	2.06	2.07	3.11	3.09	3.23	3.20	4.37	4.38	4.49	4.40
		Nodes	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	+ ESC	Time	1.98	2.00	1.99	1.98	3.07	3.08	3.10	3.08	4.40	4.36	4.43	4.42
		Nodes	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	ESC	Time	2.00	1.99	2.00	2.00	3.12	3.10	3.15	3.13	4.45	4.45	4.51	4.52
		Nodes	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
TP	M	Time	0.84	0.86	0.84	0.87	1.31	1.29	1.31	1.30	1.82	1.82	1.81	1.83
		Nodes	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	+ ESC	Time	0.68	0.66	0.66	0.67	1.11	1.09	1.12	1.12	1.59	1.58	1.61	1.58
		Nodes	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	fix y	Time	0.33	0.33	0.34	0.34	0.51	0.56	0.51	0.52	0.72	0.71	0.70	0.71
		Nodes	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	+ ESC	Time	0.69	0.70	0.69	0.70	1.14	1.15	1.12	1.12	1.62	1.70	1.65	1.63
		Nodes	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	infinite	Time	3.41	3.63	3.63	3.42	5.35	5.50	5.27	5.73	7.74	7.81	7.59	7.88
		Nodes	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	+ ESC	Time	0.68	0.69	0.68	0.68	1.12	1.11	1.11	1.12	1.65	1.64	1.61	1.61
		Nodes	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	ESC	Time	0.67	0.67	0.67	0.68	1.08	1.07	1.09	1.10	1.62	1.63	1.59	1.62
		Nodes	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

		R	50				75				100			
		demand	D		S		D		S		D		S	
		setup costs	D	S	D	S	D	S	D	S	D	S	D	S
TPC	M	Time	0.76	0.74	0.76	0.74	1.15	1.17	1.15	1.15	1.60	1.62	1.55	1.56
		Nodes	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	+ ESC	Time	0.41	0.42	0.41	0.42	0.66	0.67	0.67	0.65	0.95	0.93	0.92	0.93
		Nodes	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	fix y	Time	0.29	0.29	0.30	0.29	0.45	0.45	0.46	0.45	0.63	0.61	0.63	0.62
		Nodes	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	+ ESC	Time	0.42	0.42	0.41	0.42	0.66	0.67	0.68	0.66	0.97	0.98	0.93	0.97
		Nodes	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	infinite	Time	3.51	3.57	3.48	3.60	5.10	5.50	5.31	5.63	7.72	7.61	7.80	7.58
		Nodes	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	+ ESC	Time	0.42	0.41	0.41	0.42	0.65	0.68	0.65	0.67	0.91	0.95	0.93	0.93
		Nodes	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	ESC	Time	0.42	0.42	0.41	0.41	0.64	0.65	0.65	0.65	0.92	0.93	0.92	0.93
		Nodes	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table 18: Fixed production periods, solving time in seconds and number of nodes

7.3 Results - total max

$\|T\|$	$\|R\|$	Removed periods	MC Consecutive sum	MC Total sum	SP Consecutive sum	SP Total sum	SPC Consecutive sum	SPC Total sum	TP Consecutive sum	TP Total sum	TPC Consecutive sum	TPC Total sum
20	50	1	0.64	0.64	0.93	0.92	0.85	0.83	0.77	0.76	0.69	0.66
		2	1.02	1.06	1.08	1.07	0.93	0.91	1.07	1.09	0.97	0.95
		3	1.49	1.47	1.28	1.26	1.03	1.01	1.48	1.46	1.38	1.27
		4	1.75	1.76	1.57	1.57	1.18	1.15	1.83	1.81	1.61	1.59
		5	2.17	2.23	2.07	2.06	1.38	1.38	2.52	2.43	2.23	2.09
		6	2.37	2.37	2.58	2.56	1.60	1.59	2.88	2.83	2.53	2.45
	75	1	1.28	1.28	1.53	1.53	1.33	1.32	1.28	1.30	1.21	1.11
		2	2.09	1.85	1.69	1.66	1.43	1.36	1.77	1.84	1.63	1.52
		3	2.96	2.87	2.16	2.10	1.61	1.56	2.40	2.50	2.17	2.18
		4	4.08	3.97	2.72	2.64	1.81	1.77	3.66	3.61	3.15	3.08
		5	5.55	5.75	3.68	3.53	2.16	2.11	5.15	5.19	4.44	4.30
		6	6.75	6.59	4.62	4.74	2.55	2.54	6.79	6.71	5.76	5.55
	100	1	2.46	2.72	2.35	2.26	1.95	1.86	2.18	2.31	2.06	2.00
		2	4.45	4.55	2.57	2.55	2.02	2.08	3.35	3.20	2.88	2.76
		3	5.91	6.02	3.08	3.04	2.18	2.17	4.32	4.37	3.79	3.59
		4	7.11	7.40	3.93	3.99	2.60	2.55	5.35	5.72	5.09	4.75
		5	10.73	10.47	5.67	5.44	3.29	3.20	9.06	9.33	7.94	8.09
		6	13.45	13.39	7.53	7.27	3.96	3.82	12.86	12.94	10.69	10.69
25	50	1	1.24	1.14	1.75	1.73	1.72	1.72	1.43	1.42	1.35	1.25
		2	1.91	1.80	2.01	1.96	1.88	1.87	1.88	1.93	1.59	1.59
		3	2.51	2.71	2.39	2.35	2.05	2.10	2.73	2.86	2.27	2.34
		4	3.36	3.69	2.81	2.84	2.38	2.34	3.58	3.65	3.12	3.03
		5	4.61	4.52	3.58	3.44	2.55	2.52	4.88	4.74	3.87	3.77
		6	5.78	5.95	4.54	4.51	2.94	2.90	6.42	6.33	5.37	5.15
		7	6.66	6.41	5.85	5.72	3.37	3.36	7.90	7.67	6.21	6.00
		8	7.18	6.64	6.57	6.68	3.80	3.83	8.91	8.50	6.90	6.58
	75	1	2.61	2.57	3.06	3.22	2.84	3.05	2.57	2.72	2.48	2.32
		2	3.97	3.98	3.22	3.27	2.85	2.87	3.73	3.62	2.92	3.01
		3	6.23	6.15	3.94	3.97	3.19	3.14	5.65	5.57	4.44	4.42
		4	8.13	7.97	4.87	4.65	3.56	3.50	7.73	7.60	6.34	6.03
		5	10.78	10.32	6.02	5.90	4.02	4.10	10.22	10.06	8.62	8.26
		6	13.07	12.58	7.79	7.51	4.69	4.66	13.40	13.57	11.10	11.09
		7	17.04	16.70	10.50	9.97	5.64	5.48	19.99	19.05	15.33	15.64
		8	18.25	18.40	13.13	13.02	6.60	6.39	23.24	23.26	18.12	18.06
	100	1	5.16	5.20	4.84	5.18	4.31	4.43	4.80	4.92	4.08	4.28
		2	6.85	7.13	5.53	5.68	4.44	4.56	6.22	6.24	5.04	5.03
		3	8.71	9.33	6.29	6.50	4.74	4.76	7.55	7.61	6.11	5.94
		4	12.98	13.46	7.98	7.76	4.99	5.32	10.94	10.71	8.74	8.53
		5	16.67	17.69	9.96	9.91	5.77	5.67	14.62	14.31	12.14	11.66
		6	21.31	21.33	12.12	12.49	6.77	6.74	21.40	20.41	16.63	16.78
		7	27.31	26.98	16.26	16.27	8.32	8.00	30.69	29.88	23.62	23.46
		8	32.62	33.39	23.91	22.30	9.83	9.71	41.13	41.67	33.32	33.20

Table 19: Total max solving time in seconds

7.4 Results - Max gap

$\|T\|$	$\|R\|$	n	MC Consecutive sum	MC Total sum	SP Consecutive sum	SP Total sum	SPC Consecutive sum	SPC Total sum	TP Consecutive sum	TP Total sum	TPC Consecutive sum	TPC Total sum
15	50	1	0.07	0.07	0.31	0.3	0.32	0.31	0.1	0.1	0.07	0.06
		2	0.25	0.24	0.62	0.59	0.32	0.3	0.31	0.29	0.25	0.22
	75	1	0.11	0.11	0.47	0.47	0.49	0.48	0.16	0.16	0.1	0.09
		2	0.37	0.36	0.89	0.86	0.45	0.42	0.45	0.43	0.36	0.33
	100	1	0.16	0.16	0.66	0.64	0.68	0.67	0.22	0.22	0.14	0.13
		2	0.69	0.68	1.35	1.3	0.69	0.65	0.76	0.73	0.64	0.6
20	50	1	0.16	0.16	0.85	0.83	1.06	1.05	0.28	0.28	0.13	0.12
		2	0.42	0.4	1.49	1.41	0.72	0.63	0.6	0.57	0.5	0.44
		3	0.64	0.52	1.69	1.51	0.83	0.7	0.75	0.66	0.64	0.52
	75	1	0.25	0.25	1.31	1.3	1.63	1.61	0.44	0.44	0.2	0.19
		2	0.79	0.76	2.32	2.27	1.1	1	1.02	0.98	0.83	0.73
		3	1.41	1.12	2.92	2.67	1.39	1.24	1.4	1.27	1.15	1.02
	100	1	0.35	0.35	1.82	1.77	2.23	2.17	0.6	0.61	0.28	0.26
		2	1.42	1.4	3.48	3.31	1.67	1.55	1.69	1.59	1.38	1.22
		3	2.06	1.66	4.1	3.68	1.9	1.7	1.78	1.77	1.5	1.36
25	50	1	0.27	0.26	1.87	1.82	2.7	2.65	0.55	0.55	0.23	0.23
		2	0.71	0.68	2.96	2.83	1.42	1.25	1.09	1.03	0.88	0.76
		3	1.31	0.94	3.43	3.02	1.7	1.43	1.51	1.28	1.29	0.96
	75	1	0.41	0.41	2.88	2.82	4.19	4.12	0.87	0.82	0.35	0.34
		2	1.58	1.51	4.96	4.75	2.54	2.32	2.19	2.11	1.8	1.64
		3	2.13	1.86	5.53	4.88	2.68	2.38	2.45	2.16	1.98	1.71
	100	1	0.59	0.57	3.89	3.88	5.62	5.55	1.16	1.11	0.49	0.47
		2	2.89	2.87	7.21	7.04	3.69	3.37	3.4	3.25	2.77	2.5
		3	3.65	3.3	8.46	7.89	4.23	3.79	3.58	3.58	2.85	2.84

Table 20: Max gap solving time in seconds and number of nodes

7.5 Results - Min gap

$\|T\|$	$\|R\|$	n	MC Consecutive sum	MC Total sum	SP Consecutive sum	SP Total sum	SPC Consecutive sum	SPC Total sum	TP Consecutive sum	TP Total sum	TPC Consecutive sum	TPC Total sum
15	50	2	0.27	0.26	0.37	0.39	0.35	0.34	0.32	0.27	0.28	0.24
		3	0.41	0.25	0.39	0.33	0.36	0.30	0.39	0.26	0.37	0.23
	75	2	0.44	0.39	0.54	0.55	0.54	0.50	0.54	0.43	0.48	0.39
		3	0.83	0.48	0.70	0.53	0.62	0.47	0.77	0.48	0.72	0.43
	100	2	1.13	0.87	0.93	0.88	0.85	0.74	0.95	0.78	0.89	0.68
		3	1.69	0.89	1.18	0.77	0.98	0.67	1.42	0.78	1.32	0.72
20	50	2	0.47	0.43	0.89	0.91	0.86	0.82	0.68	0.56	0.60	0.52
		3	1.07	0.58	1.25	0.90	1.47	0.79	1.20	0.69	1.10	0.62
		4	1.44	0.68	1.46	0.90	1.79	0.79	1.61	0.75	1.45	0.67
	75	2	1.13	0.87	1.45	1.44	1.44	1.32	1.34	1.02	1.17	0.93
		3	2.41	1.25	1.98	1.37	2.26	1.23	2.36	1.19	2.10	1.09
		4	3.52	1.67	2.87	1.55	3.76	1.29	3.71	1.63	3.36	1.45
	100	2	2.17	1.63	2.15	2.10	2.00	1.81	2.15	1.65	1.95	1.42
		3	5.38	2.42	3.34	2.03	3.12	1.78	4.49	1.96	3.85	1.69
		4	7.40	3.27	4.87	2.35	6.15	1.88	7.27	2.87	6.49	2.55
25	50	2	0.87	0.75	1.71	1.79	1.85	1.66	1.28	1.04	1.20	0.94
		3	2.19	1.08	2.69	1.77	3.71	1.73	2.66	1.30	2.42	1.18
		4	3.59	1.61	3.74	1.97	7.30	1.84	5.01	1.84	4.33	1.63
	75	2	2.18	1.80	3.25	3.13	3.01	2.89	2.58	1.97	2.36	1.81
		3	7.19	3.13	6.17	3.26	8.53	2.77	8.26	3.16	6.63	2.73
		4	9.64	4.29	7.92	3.68	12.31	3.02	12.61	4.31	10.17	3.60
	100	2	7.21	4.26	6.17	5.23	5.45	4.49	6.92	4.21	6.01	3.44
		3	16.72	7.24	10.55	4.74	12.92	3.97	18.60	5.95	14.21	4.87
		4	20.55	9.00	14.46	5.91	18.00	4.54	27.37	8.52	20.58	6.76

Table 21: Min gap solving time in seconds and number of nodes

7.6 Echelon stock constraints

We give the way to rewrite constraints (21) using the variable used in each of the formulations introduced (see 4.1.2).

OWMR:

$$I_t^0 = s_t^0 + \sum_{c\in R} s_t^c, \quad \forall t \in T \tag{54}$$

SES:

$$I_t^0 = \sum_{j=1}^{t} \sum_{k=t+1}^{|T|} X_{jk}^0, \quad \forall t \in T \tag{55}$$

TP and TPC:

$$I_t^0 = \sum_{c \in R} \sum_{q=1}^{t} \sum_{k=t+1}^{|T|} \sum_{j=q}^{k} W_{qjk}^c, \quad \forall t \in T \tag{56}$$

SP and SPC:

$$I_t^0 = \sum_{c \in R} \sum_{q=1}^{t} \sum_{k=t+1}^{|T|} (\sum_{j=q}^{t} U_{qjk}^c d_{t+1,k}^c + \sum_{j=t+1}^{k} U_{qjk}^c d_{j,k}^c), \quad \forall t \in T \tag{57}$$

MC:

$$I_t^0 = \sum_{c \in R} \sum_{j=t+1}^{|T|} (\sigma_{tj}^{0c} + \sigma_{tj}^{1c}), \quad \forall t \in T \tag{58}$$

7.7 Fixed production periods, exact methods

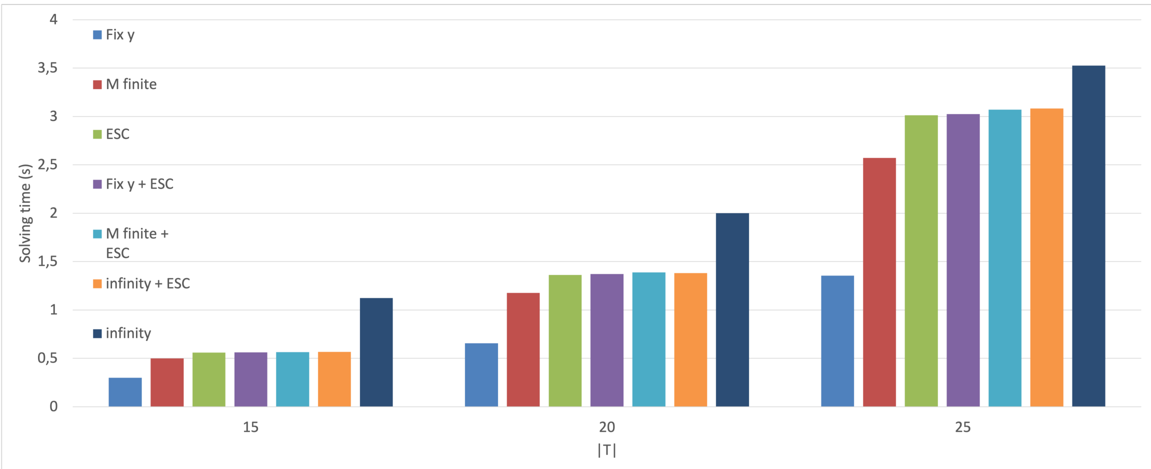

Figure 4: Average solving time in seconds

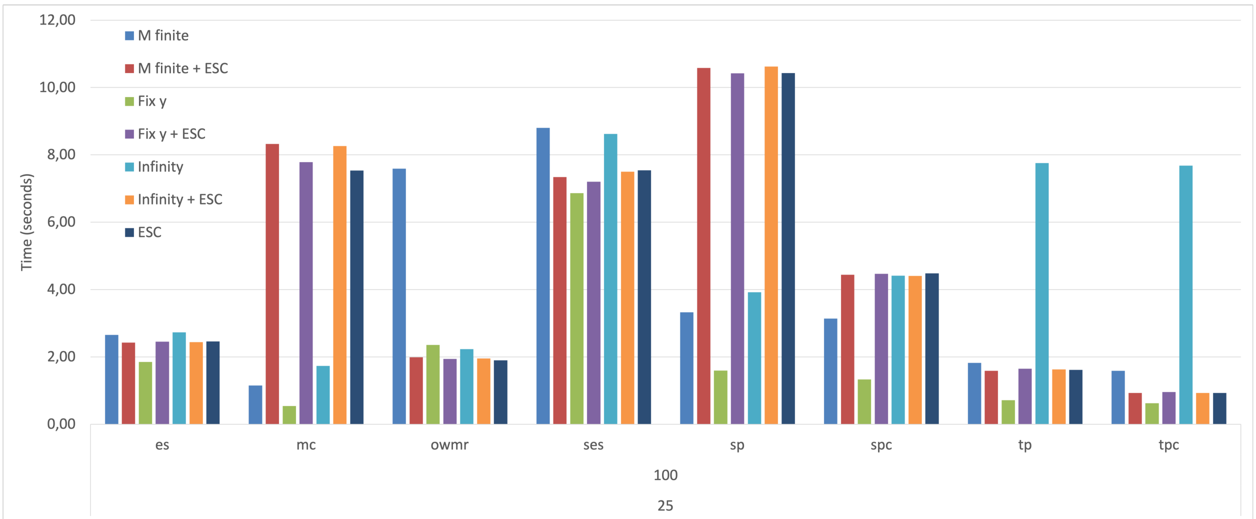

Figure 5: Average solving time in seconds ($|T| = 25, |R| = 100$)