Joint optimization of dynamic pricing and lot-sizing decisions with nonlinear demands: Theoretical and computational analysis

Paulin Couzon, Yassine Ouazene, Farouk Yalaoui

To cite this version:

Paulin Couzon, Yassine Ouazene, Farouk Yalaoui. Joint optimization of dynamic pricing and lot-sizing decisions with nonlinear demands: Theoretical and computational analysis. Computers and Operations Research, 2019, pp.104862. 10.1016/j.cor.2019.104862 . hal-02391554

HAL Id: hal-02391554
https://utt.hal.science/hal-02391554

Submitted on 21 Jul 2022

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

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

Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

Version of Record: https://www.sciencedirect.com/science/article/pii/S0305054819303041
Manuscript_6acdd615fe0bc56e13c7b6746c9432fc

Joint optimization of dynamic pricing and lot-sizing decisions with nonlinear demands: Theoretical and computational analysis

Paulin Couzon[a,b,*], Yassine Ouazene[a,b], Farouk Yalaoui[a,b]

[a] *Chaire Connected Innovation, Université de Technologie de Troyes, 12 rue Marie Curie, CS 42060 10004 Troyes CEDEX, France*
[b] *ICD-LOSI, Université de Technologie de Troyes, CNRS, 12 rue Marie Curie, CS 42060 10004 Troyes CEDEX, France*

Abstract

This paper addresses a capacitated lot-sizing problem with pricing decisions. The considered problem consists of planning the production of different products during several time periods with setup costs. Unlike the classical version of the capacitated lot sizing problem, the demand for the products is not fixed but price-sensitive in this problem. The demand function is assumed to be nonlinear. The decisions consist of establishing the best strategy for production and inventory, and the best price policy. We propose an improved mathematical formulation by extending some previous works using new lower and upper bounds to reduce the solution space. We also introduce new heuristic methods to provide near-optimal solutions. These methods are tested on several instances from previous studies. The obtained results illustrate the efficiency of these methods.

Keywords: Lot-sizing, Dynamic pricing, Mathematical programming, Heuristic method optimization, Isoelastic demand

---

*Corresponding author
Email address: paulin.couzon@utt.fr (Paulin Couzon)

© 2019 published by Elsevier. This manuscript is made available under the CC BY NC user license
https://creativecommons.org/licenses/by-nc/4.0/

1. Introduction

Dynamic pricing can be considered one of the most powerful features for companies to increase their profit and the adaptability of factories (Kienzler (2018)). The use of a pricing strategy associated with demand knowledge helps companies manage their production and optimize their inventory policy.

Dynamic pricing has been a subject of research for economists for more than 50 years. It became popular with companies after the success of yield management in the airline industry, increasing company profits in the sector (for instance, it increased the profits of American Airlines by 5% (Smith et al. (1992))). Yield management has been successfully applied to the hotel industry (Bitran and Mondschein (1995), Bandalouski et al. (2018)) and passenger trains (Bharill and Rangaraj (2008)). It is also currently used by e-retailers (for example Amazon (Chen et al. (2016))), but has yet to be applied to complex production and inventory systems.

Different surveys on dynamic pricing and inventory decisions were conducted: Bitran and Caldentey (2003), Elmaghraby and Keskinocak (2003), and Chen and Chen (2015). In addition, Chan et al. (2004) and Chen and Simchi-Levi (2012) considered problems dealing with inventory and dynamic pricing decisions with production considerations.

Differences between monopolistic and competitive markets must also be considered. A competitive market is generally modelized using game theory. In Adida and Perakis (2010) and Lamas and Chevalier (2018), problems with a company and multiple competitors are modelized. The main purpose of the studies was to find the Nash equilibrium maximizing the total profitability of their systems.

In studies on dynamic pricing with monopolistic companies, two main types of modelization have arisen. The first considers a Markov or Poisson decision process. The demand for a product is then represented by an arrival rate of the customers, depending on the price. For instance, Chen et al. (2015) studied a make-to-order system with pricing and modelized the demand using a Poisson process. In Yu et al. (2017), a production and inventory system, in which certain products could be substituted by higher-quality products to fulfill a demand, was studied. In the cited papers, the authors determined an optimal price and production policy based on the state of the system.

The second main type of modelization for monopolistic problems is mathematical programming. The majority of these mathematical formulations are in fact based on lot-sizing formulations. The main difference with lot-sizing problems concerns a non-fixed product demand. The demand is modelized by a closed-form function depending on the price. The study in Thomas (1970) was one of the first to deal with pricing and lot-size decisions, and the author derived an optimal price policy based on discrete values of the production and demand. Haugen et al. (2007) expanded the capacitated lot-sizing problem by introducing pricing considerations. This model may also be viewed as an expansion of Thomas' study. In Mahmoudzadeh et al. (2013), a problem of the production of new products and a remanufacturing of old products was considered. They also proposed a deterministic and robust mathematical formulation. In Bajwa et al. (2016a), a model similar to that by Haugen et al. (2007) was proposed, although it allowed lost sales and applied an exact algorithm. The authors also compared the results obtained by the non-coordinated and the coordinated approach. Ouazene et al. (2017) studied a problem with multiple selling channels for products. The authors compared the impact of dynamic and static pricing on the profits.

In Huang et al. (2013), a survey of different methods for modelizing customer demand was presented. The two most widely used type of demand functions are linear and isoelastic functions. All the studies on lot-sizing cited here considered a linear demand. The survey also provided interesting variants for incorporating advertising, rebates, quality, and lead-time into the demand function.

Few problems are modelized through differential equations. In Maihami and Karimi (2014), a generic, stochastic, and price-dependent demand was proposed. The authors proved the uniqueness of their solution and developed an algorithm to determine it. Feng et al. (2018) proposed a model with an inventory level based on a differential equation, where the production is controlled based on the production rate. The resolution approach used Lagrange multipliers, and was aimed at finding the optimal price and production rate for each model state. In Wu et al. (2017), a lot-sizing and pricing problem was presented, which used the Bass diffusion model for the demand of new products. The problem is modelized using a difference equation (discretized differential equation), and every state of the system is fully determined by its initial state.

Studies dealing with lead-time decisions and dynamic pricing have also been previously published. In Li et al. (2012), a model with lot-sizing decisions was developed, where the customer may accept a delayed delivery to earn a discount. In Öner-Közen and Minner (2017), a Markov decision process was designed based on price and lead-time dependent consumers, and considering a guaranteed lead-time decision. The authors derived optimal strategies for variable decisions. Finally, in Albana (2018) a model for a supply chain with pricing and lead-time decisions was proposed. The author considered a decrease in production cost based on the lead-time.

Regarding the variety of models studied in the literature, a comparison of the different methods is not easy, and few benchmarks or data exist to test these methods. In addition, there has been a lack of real-world application studies in the literature, despite all advantages a company may earn by using dynamic pricing. For a company, achieving an accurate representation of product demand is complex, requiring numerous price changes to estimate the demand parameters. In den Boer (2015), a survey on demand learning was provided, as well as a method for using it in a dynamic pricing problem. The results illustrate the trade-off companies face between learning to have a reliable demand representation and using the results to create a profit. In Fisher et al. (2017), an example of successful demand learning is described. The methods applied to obtain real data are presented in their study. The authors worked with an e-retailer, experimented with random price changes over a one month timeframe and recorded daily changes in their competitors. They succeeded in obtaining an accurate demand function, and the use of this function in a dedicated algorithm increases the e-retailer revenue by 11% for the tested products. Herbon (2014) also considered an inventory problem with dynamic pricing decisions, and the author aimed to find the optimal choice for a retailer between the development of a dynamic pricing model and the collection of customer purchase information.

In this study, we focus on a lot-sizing problem with pricing decisions. The problem considers multiple products and setup cost. The demand function is assumed to be isoelastic.

The main contributions of this paper can be summarized as follows.

- A new mathematical model is proposed.
- A heuristic method is developed, which provides high-quality solutions.

The remainder of this paper is organized as follows. Section 2 is dedicated to the mathematical formulation and its theoretical properties. In section 3, the resolution method based on two constructive heuristics is detailed. Section 4 compares the results of the heuristics on instances from the literature with results obtained using the LINGO solver. Finally, some concluding remarks and insights into future study are provided.

2. Problem formulation

The initial problem considered in this study is the same as the most general model addressed in Bajwa et al. (2016a). Their mathematical model is denoted in our study as (P_0). The problem represents a company producing and selling products. The company is also able to stock its products to sell them later. The seller decides during each period the quantity of products produced, the quantity stocked, and the quantity sold to maximize its profits.

The customer demand for each product is represented by a price-dependent function. In our study, the considered demand function is an isoelastic one: $D(P_{jt}) = \gamma_{jt}\alpha_j P_{jt}^{-\beta_j}$, where γ is the seasonality parameter, α is the demand for a price equal to 1, and β is the price elasticity of the demand. The price of each product can be changed during each period (we assume here that the cost of changing the price is negligible).

2.1. Notations

The decision variables used in the mathematical model are as follows:

- P_{jt}: Price for product j during period t;
- S_{jt}: Quantity of product j sold during period t;
- X_{jt}: Quantity of product j produced during period t;
- I_{jt}: Inventory of product j at the end of period t;
- Y_{jt}: Binary, equal to 1 if there is a setup to produce j during period t; 0, otherwise.

The first four variables are non-integer and greater than or equal to zero. The last one is a binary variable $\{0,1\}$.

The following sets and notations are used for the mathematical model:

- J: Number of products;
- T: Number of time periods;
- c_{jt}: Production cost for product j during period t;
- h_{jt}: Inventory cost for product j during period t;
- a_{jt}: Setup cost to produce j during period t;
- C_t: Production capacity during period t;
- v_j: Production capacity used for one unit of product j;
- α_j: Demand of product j for a price equal to 1;
- β_j: Price elasticity of the demand for product j;
- γ_{jt}: Demand seasonality of product j during period t .

2.2. Initial model

The initial model (P_0), is detailed below:

$$\max z = \sum_{j=1}^{J}\sum_{t=1}^{T}(P_{jt}S_{jt} - c_{jt}X_{jt} - h_{jt}I_{jt} - a_{jt}Y_{jt}) \tag{1}$$

such that

$$S_{jt} - \gamma_{jt}\alpha_j P_{jt}^{-\beta_j} \leq 0, \forall j, t \tag{2}$$

$$\sum_{j=1}^{J} v_j X_{jt} \leq C_t, \forall t \tag{3}$$

$$S_{jt} + I_{jt} - I_{jt-1} - X_{jt} = 0, \forall j, \forall t \in \{2, ..., T-1\} \tag{4}$$

$$S_{j1} + I_{j1} - X_{j1} = 0, \forall j \tag{5}$$

$$S_{jT} - I_{jT-1} - X_{jT} = 0, \forall j \tag{6}$$

$$v_j X_{jt} - C_t Y_{jt} \leq 0, \forall j, t \tag{7}$$

$$X_{jt}, S_{jt}, P_{jt}, I_{jt} \geq 0, Y_{jt} \in \{0, 1\}\ , \forall j, t \tag{8}$$

Equation (1) represents the objective function defined as a profit maximization.

Constraints (2) ensure the sales for the products to be less than or equal to the demand for these products, where an inequality means that lost sales are allowed. Constraints (3) are production capacity constraints. Constraints (4), (5), and (6) represent the stock conservation constraints, with the initial and final inventory equal to zero. Finally, constraints (7) add a setup cost for each production period.

2.3. Discussion on the nonconvexity of the problem

This model (P_0) is a classical model already proposed in the literature. Owing to the constraints (2) and objective function, the model is nonlinear. In addition, the same constraints make the model nonconvex. The outer approximation algorithm was introduced by Duran and Grossmann (1986) and consists of decomposing a model into a master problem with relaxed nonlinear constraints and several primal models. The results of the primal model are injected into the master problem to linearize the removed constraints. This algorithm was used by Bajwa et al. (2016a) to solve the problem with a linear demand, but is not usable for a model with an isoelastic demand function. This algorithm requires constraints expressed by $g(x) \leq 0$, where g is a convex function, and x is a real variable. In the studied model, constraints (2) do not satisfy those conditions. For instance, the same conditions are also necessary to use the Benders decomposition of Geoffrion (1972). Attempts at relaxing the convexity conditions have been proposed by Kesavan et al. (2004) and Bergamini et al. (2008), although these methods were ineffective at solving this problem.

The consequence of this nonconvexity is the inability to replicate the results obtained by Bajwa et al. (2016a) for the isoelastic case, which is why we chose to develop specific methods and compare them only with a commercial solver.

Prior to the use of this software to solve the model, it had to be modified to enhance its computational tractability. In its current state, the model cannot be solved efficiently by a commercial solver.

The modified version of the model is constructed by adding new lower and upper bounds to limit the solution search space. A comparison of the efficiency of LINGO software for the resolution of the model P_0 and the modified model P_1 is presented in section 4.

2.4. Proposed model

The results described in this section are based on the mathematical analysis of (P_0) and more specifically on a reformulation with fixed setup variables. The results consist of new bounds for the decision variables. These bounds cut off non-optimal values of the variables.

The first bound developed limits the value of the sales variables S_{jt}.

Proposition 1. *:*

The optimal quantity of each product j sold during period t is bounded as follows.

$$S^*_{jt} \leq \frac{\alpha_j \gamma_{jt}(1 - \frac{1}{\beta_j})^{\beta_j}}{(\min_{t_0 \in \{1,..,t\}}(c_{jt_0} + \sum_{k=t_0}^{t-1} h_{jk}))^{\beta_j}} \text{ , } \forall j, \forall t \tag{9}$$

Proof. The proof of this proposition is based on a reformulation of the model (P_0) with a fixed setup configuration (fixed values of the variables Y_{jt}). The reformulation replaces four 2-index variables by one 3-index variable. This is a reformulation of the initial model (P_0) with only one variable type.

This new model is based on the notations introduced in the previous section and on the following notations:

- $N = \{(j,m,n), j \in \{1,..,J\}, m \in \{1,...,T\}, n \in \{m,...,T\}, Y_{jm} = 1\}$
- $A_{jmn} = c_{jm} + \sum_{t=m}^{n-1} h_{jt}, \forall (j,m,n) \in N$
- $B_{jn} = \alpha_j \gamma_{jn}, \forall j, n$

Here, X_{jmn} represents the quantity of product j produced during period m and sold during period n.

The initial variables X_{jt}, S_{jt}, and I_{jt} from model (P_0) are replaced by the variables X_{jmn}. The relationships between these variables are given as follows:

- $X_{jt} = \sum_{n=t}^{T} X_{jtn}, \forall j, t$
- $S_{jt} = \sum_{m=1}^{t} X_{jmt}, \forall j, t$

- $I_{jt} = \sum_{m=1}^{t} \sum_{n=t+1}^{T} X_{jmn}, \forall j, t$

The mathematical model, denoted as (RP_1), is as follows:

$$\text{maximize } z_P = \sum_{(j,m,n)\in N} (X_{jmn}((B_{jn})^{\frac{1}{\beta_j}}(\sum_{l=1}^{n} X_{jln})^{-\frac{1}{\beta_j}} - A_{jmn})) - \sum_{j=1}^{J}\sum_{m=1}^{T} a_{jm}Y_{jm} \quad (10)$$

such that

$$\sum_{j=1}^{J}\sum_{n=m}^{T} v_j X_{jmn} \leq C_m \ , \forall m \in \{1, ..., T\} \quad (11)$$

$$X_{jmn} \geq 0 \ , \forall (j,m,n) \in N \quad (12)$$

The formulation of equation (10) is based on theorem 1 from Bajwa et al. (2016a) assuming that there are no lost sales for the optimal solution. This means that inequalities (2) are replaced by equalities. Then, $P_{jt} = (\frac{S_{jt}}{\alpha_j \gamma_{jt}})^{-\frac{1}{\beta_j}} = (\frac{\sum_{m=1}^{t} X_{jmt}}{\alpha_j \gamma_{jt}})^{-\frac{1}{\beta_j}}$. By replacing P_{jt} with this expression in the objective function (1), we are able to formulate the entire equation as a function depending only on X_{jmn}.

In the case of a non-limiting capacity (i.e., constraint (11) is always verified), the optimal value of X_{jmn} is obtained by differentiating the objective function with respect to X_{jmn}, setting this function to 0, and solving for X_{jmn}. The value obtained is as follows:

$$X^*_{jmn} = \alpha_j \gamma_{jn} \left(\frac{1 - \frac{1}{\beta_j}}{c_{jm} + \sum_{t=m}^{n-1} h_{jt}} \right)^{\beta_j} - \sum_{l=1, l\neq m}^{n} X_{jln}. \quad (13)$$

Without a loss of generality, we may suppose that all of the products sold during a period are made during only one previous period. Then, because the relationship between the S_{jt} and X_{jmn} variables is given by $S_{jt} = \sum_{m=1}^{t} X_{jmt}$, the optimal quantity sold during a period n depends only on one value over all m periods of X_{jmn}.

This production period is denoted by m^*. It follows then that $S^*_{jt}(m^*) = \alpha_j \gamma_{jt} \left(\frac{1 - \frac{1}{\beta_j}}{c_{jm^*} + \sum_{n=m^*}^{t-1} h_{jn}} \right)^{\beta_j}$.

Finally, its maximal value over all possible values of m^* is given as follows:

$$\max_{m^*} S_{jt}^*(m^*) = \alpha_j \gamma_{jt} \left(\frac{1 - \frac{1}{\beta_j}}{\min_{t_0 \in \{1,..,t\}} (c_{jt_0} + \sum_{k=t_0}^{t-1} h_{jk})} \right)^{\beta_j}. \tag{14}$$

$\square$

The following properties are derived from this inequality.

Corollary 1. *For each product, the total optimal quantity produced is bounded as follows.*

$$\sum_{t=1}^{T} X_{jt}^* \leq \sum_{t=1}^{T} \frac{\alpha_j \gamma_{jt} (1 - \frac{1}{\beta_j})^{\beta_j}}{(\min_{t_0 \in \{1,..,t\}} (c_{jt_0} + \sum_{k=t_0}^{t-1} h_{jk}))^{\beta_j}} \ , \forall j \tag{15}$$

A lower bound for the optimal price of each product at each period is also provided.

$$P_{jt}^* \geq \frac{(\min_{t_0 \in \{1,..,t\}} (c_{jt_0} + \sum_{k=t_0}^{t-1} h_{jk}))}{1 - \frac{1}{\beta_j}} \ , \forall j, \forall t \tag{16}$$

Proof. For the first part of the corollary, we start by adding equations (4), (5), and (6) $\forall t \in T$, and the result of this sum is $\sum_{t=1}^{T} X_{jt} = \sum_{t=1}^{T} S_{jt}$, $\forall j$.

Finally, with the inequality from proposition 1, the following result is obtained.

$$\sum_{t=1}^{T} X_{jt}^* = \sum_{t=1}^{T} S_{jt}^* \leq \sum_{t=1}^{T} \frac{\alpha_j \gamma_{jt} (1 - \frac{1}{\beta_j})^{\beta_j}}{(\min_{t_0 \in \{1,..,t\}} (c_{jt_0} + \sum_{k=t_0}^{t-1} h_{jk}))^{\beta_j}} \tag{17}$$

For the second part, the proof starts from $S_{jt}^* = \gamma_{jt} \alpha_j (P_{jt}^*)^{-\beta_j}$ (the optimal solution does not include any lost sales).

By replacing S_{jt} by its closed-form into (9), we obtain the following:

$$\gamma_{jt} \alpha_j (P_{jt}^*)^{-\beta_j} \leq \frac{\alpha_j \gamma_{jt} (1 - \frac{1}{\beta_j})^{\beta_j}}{(\min_{t_0 \in \{1,..,t\}} (c_{jt_0} + \sum_{k=t_0}^{t-1} h_{jk}))^{\beta_j}} \tag{18}$$

Finally, we have the following:

$$P_{jt}^* \geq \frac{(\min_{t_0 \in \{1,..,t\}}(c_{jt_0} + \sum_{k=t_0}^{t-1} h_{jk}))}{1 - \frac{1}{\beta_j}} \tag{19}$$

$\square$

Remark. *Within equations (9), (15), and (16), the term* $(\min_{t_0 \in \{1,..,t\}}(c_{jt_0} + \sum_{k=t_0}^{t-1} h_{jk}))$ *may be replaced by* c_{jt} *if parameters* c_{jt} *are constant over time or more generally when equation* $c_{jt} \leq c_{jt_0} + \sum_{k=t_0}^{t-1} h_{jk}$ *,* $\forall t, \forall t_0 \in \{1, ..., t\}$ *is verified.*

In the following section, the model considered is denoted by (P_1) and is made up of model (P_0) with the addition of equations (9), (15), and (16).

3. Optimization approaches

Using LINGO software to obtain optimal or near-optimal solutions is appealing. However, it is unsustainable as the solver quickly reaches its limits with an increase in the size of the instance. In addition, a proof of the optimal solution is unavailable for this problem when using LINGO to solve it. Therefore, an alternative to LINGO was developed in this study.

Our approach relies on the fact that, for a given setup (fixed values of Y_{jt}), an optimal solution can be found by using the model reference adaptive (MRA) search algorithm developed in Bajwa et al. (2016a).

3.1. MRA algorithm

This algorithm was introduced by Bajwa et al. (2016a), and we present its main steps in this section, referring to the original study for further details and a theoretical proof. MRA starts with an initial feasible solution, generates an increasing sequence of objective values, and terminates at an optimal solution.

The notations for the algorithm are the same as those used in the model (RP_1) presented during the proof of Proposition 1.

The algorithm is based on two different cases that may occur during a given period:

Case 1: The capacity constraint is not binding for the optimal solution. Then, for a given m period, the optimal solution is found by differentiating z_P with respect to X_{jmn}, setting it to 0, and solving for X_{jmn} as follows:

$$X^*_{jmn} = \frac{B_{jn}(1 - \frac{1}{\beta_j})^{\beta_j}}{A^{\beta_j}_{jmn}} - \sum_{l=0,l\neq m}^{n} X_{jln} \tag{20}$$

Case 2: The capacity constraint is binding for the optimal solution. In this case, the Lagrangian operator can be written as follows: $L = -z_P + \sum_{m=1}^{T} \lambda_m (\sum_{j=1}^{J} \sum_{n=m}^{T} v_j X_{jmn} - C_m)$.

Then, for a given m period, the optimal solution may be found as a function of λ_m:

$$X^*_{jmn} = \begin{cases} B_{jn} \frac{(1-\frac{1}{\beta_j})^{\beta_j}}{(A_{jmn}+\lambda_m v_j)^{\beta_j}} - \sum_{l=0,l\neq m}^{n} X_{jln} & \text{if } X^*_{jmn} \in N^+_m \\ 0 & \text{if } X^*_{jmn} \in N^0_m \end{cases} \tag{21}$$

where N^+_m is the set of variables taking a positive value for the optimal solution (the most valuable), and N^0_m is the set of variables equal to zero for the optimal solution (the least valuable).

It is not possible to obtain a closed-form value for λ_m here by replacing (21) into the capacity constraint. Therefore, λ_m is determined using a bisection method, with the value in $[0, \max_{(j,n)}(\frac{1}{v_j} \frac{\partial z_P}{\partial X_{jmn}}(0))]$.

The purpose of this algorithm is to find the variables belonging to the set N^+_m and to find the values of λ_m associated with this set.

This algorithm was proved to be optimal for a given setup, then the main issue here is to find the optimal configuration, or at least a near-optimal one. Two different constructive methods have been developed to build this setup configuration.

These methods use the assumption that the production during a particular period will occur only if the inventory level is equal to zero at the end of the previous period. This

Algorithm 1: MRA pseudocode Bajwa et al. (2016a)

$X_{jmn} = \epsilon$, $\forall j \in \{1, ..., J\}, \forall m \in \{1, ..., T\}, \forall n \in \{m, ..., T\}$;
$z_P^0 = 0$, $s = 0$, $\Delta z = 1$;
while $\Delta z > 0$ **do**
  $s = s + 1$, $m = 1$;
  while $m \leq T$ **do**
    if *Case 1* **then**
      $X_{jmn} = \max(\frac{B_{jn}(1-\frac{1}{\beta_j})^{\beta_j}}{A_{jmn}^{\beta_j}} - \sum_{l=0,l\neq m}^{n} X_{jln}, 0)$, $\forall j \in \{1, ..., J\}, \forall n \in \{m, ..., T\}$;
    else
      Sort $\frac{1}{v_j}\frac{\partial z_P}{\partial X_{jmn}}(0)$ in non-increasing order, $\forall j \in \{1, ..., J\}, \forall n \in \{m, ..., T\}$;
      Denote X_q as the variable associated with the q^{th} position in the sorting based on the $\frac{1}{v_j}\frac{\partial z_P}{\partial X_{jmn}}(0)$ values;
      $q_m = 0$, $q_m^* = J * (T - m)$;
      while $q_m^* \neq q_m$ **do**
        $q_m = q_m + 1$;
        Determine $\lambda_m(q_m)$ using the bisection method;
        if $\frac{1}{v_{q_m+1}}\frac{\partial z_P}{\partial X_{q_m+1}}(0) \leq \lambda_m(q_m)$ **then**
          $q_m^* = q_m$;
      $\lambda_m^* = \lambda_m(q_m^*)$;
      Compute X_{jmn} from equation (21) $\forall j \in \{1, ..., J\}, \forall n \in \{m, ..., T\}$;
    m=m+1;
  Compute z_P^s from X_{jmn};
  $\Delta z = z_P^s - z_P^{s-1}$;

assumption was proved to be optimal for a class of incapacitated lot-sizing problems by Wagner and Whitin (1958). For model (P_1), this assumption will not provide an optimal solution, but it may give a good solution.

To improve the results from the constructive heuristics, several local search moves have also been implemented.

3.2. Heuristic 1

This heuristic starts with an initial sorting of all products. Then, the entire setup is set for each product. The configuration values are set by assigning a setup to a product and a period if the production capacity associated with this period is able to cover the optimal sales of several periods, or if the capacity production is able to cover, at least partially, its own period. Finally, the setup configuration obtained is evaluated using the MRA algorithm presented in the previous section.

Four different decision rules are used to sort the products:

- S1: Sort the products according to decreasing values of $\frac{\beta_j}{\log(\alpha_j)}$;
- S2: Sort the products according to decreasing values of $\frac{\alpha_j}{\beta_j}(\frac{1-\frac{1}{\beta_j}}{c_{jt}})^{\beta_j-1}$;
- S3: Sort the products according to decreasing values of $\alpha_j(\frac{1-\frac{1}{\beta_j}}{c_{jt}})^{\beta_j}$;
- S4: Sort the products according to decreasing values of $\frac{a_{jt}}{h_{jt}}$.

The first decision rule is based on a simple evaluation that differentiates the products by their demand parameters. The second decision rule prioritizes the products with the highest potential revenue. The value used to sort the products is the optimal revenue if the product is processed and sold during the same period and the production capacity is sufficient to produce it. This value is determined from model (RP_1). For the third decision rule, the value is the optimal quantity associated with the optimal revenue used for the second decision rule. Finally, the fourth decision rule sorts the products by using the ratio between the setup cost and the inventory cost. This last sorting rule prioritizes the products with the highest setup cost and the lowest inventory cost, and aims at minimizing the impact on the total setup cost.

Algorithm 2 summarizes the different steps of the proposed heuristic.

Algorithm 2: Pseudocode of Heuristic 1

Apply one of the four sorting rules;
Initialize all setup values to 0;
$C_{t,remaining} = C_t, \forall t$;
foreach $j \in J$ **do**
    t=0;
    while $t < T$ **do**
        if $C_{t,remaining} > \alpha_j (\frac{1-\frac{1}{\beta_j}}{c_j})^{\beta_j}$ **then**
            $Y_{jt} = 1$ and $Y_{jt_1} = 0$, $\forall t_1 > t$ such that $C_{t,remaining} \geq \sum_{k=t}^{t_1} v_j X^*_{jk}$, with X^*_{jk} being the optimal production quantity;
            Let t_2 be the last period covered by the production;
            Update($C_{t,remaining}$);
            $t = t + t_2$;
        else if $C_{t,remaining} > 0$ **then**
            $Y_{jt} = 1$;
            $C_{t,remaining} = 0$;
            t=t+1;
        else
            $Y_{jt} = 0$;
            t=t+1;
Use the MRA algorithm to evaluate the solution;

3.3. Heuristic 2

During each period, the algorithm decides which products to produce, and how many periods are covered by the production. To determine the best assignment, the algorithm evaluates the partial solution value. For the evaluation, the remaining unassigned periods are filled with the setups (all remaining Y_{jt} are fixed at 1) and the evaluation of the setup configuration by MRA algorithm provides the objective value. The setup values for the remaining periods are fixed at 1 to easily compare the solutions obtained. Owing to this partial evaluation, the algorithm is able to choose locally the best solution by trying different setup configurations and evaluating them.

The pseudocode of this heuristic is detailed in Algorithm 3.

Algorithm 3: Pseudocode of Heuristic 2

```
Initialize all setup values to 0;
t_available(j) = 1 , ∀j ∈ {1, ..., J};
foreach t ∈ T do
    j=1;
    while j<J do
        if t_available(j) = t then
            Assign the setup for period t of the current best partial solution by
              prioritizing product j;
            Complete the partial setup configuration by adding 1 to each
              non-assigned setup;
            Use the MRA algorithm to evaluate the current configuration;
        j=j+1;
    Keep the best solution for all values of j for period t;
    Update the best partial solution;
    Update t_available from the partial solution;
```

Within heuristic H1, the setup variables are product-dependent, whereas for heuristic H2, the setup variables are time-dependent. In addition, the solutions for heuristic H1 are evaluated at the end, whereas for heuristic H2, the solutions are evaluated after each step.

To improve the efficiency of the heuristics, the instances have been solved consecutively from high to low capacity. The algorithm stores the previous solution. It evaluates this solution with the new capacity alongside the new one, to compare the results and keep the best one. Because the computational time of the heuristic remains at less than 1 s, it is possible to solve the instances with all different capacity values consecutively without too much computational effort. In addition to this, it may be wise for a company to determine the solutions for different production capacities to evaluate the benefits of a change in such capacity. For such a company, the modification of the algorithm does not involve any change in the total computation time.

A solution evaluated with a decreased capacity will nevertheless remain of good quality because the MRA algorithm is able to re-assign the partial production between products. This solution may also be better than that obtained using the heuristics again because the heuristics tend to increase the total number of setups when the production capacity is decreased, and the addition of a setup may sometimes be worse than a partial production.

3.4. Local search procedure

A local search is widely used after the constructive heuristics or metaheuristics are applied. This helps the algorithms avoid being stuck in a local optimum to obtain a better solution. Better solutions can sometimes be found by exploring the immediate neighborhood of the previous solution. Here, the neighborhood of a solution is represented only by the moves used to modify the solution. This neighborhood representation prevents an enumeration of all possible feasible solutions, therefore limiting the total computational cost of this representation.

Three local search moves are proposed. They are all used with a "first-improvement" policy to limit the computational load of the local search.

- Move 1:
  - Find two periods for which two products have a setup assigned to both periods;
  - Unassign a setup period for each product.
- Move 2:
  - Find two periods for which two products have a setup assigned to one period (not the same for both products);
  - Swap their setup assignments for these two periods.
- Move 3: Shift one setup assignment of one product to another available period.

The first move is particularly useful for instances with a low capacity, where the heuristics have trouble reducing the total number of setups. By deleting two setup assignments at two different periods, the production will be reported to other periods, and the profit impact from the total production decrease will be balanced by the cost decrease owing to the setup removal.

The second move aims at correcting some sub-optimal assignments due to the initial sorting for H1 or on-time sorting for H2. These sorting rules generate priority for the products, but the global impact of these priorities cannot be evaluated at the moment of a decision, and may be easily modified afterwards using this local search move. The third move has a lower impact than the second one as it affects only one product, although it can be helpful to improve a setup assignment. More specifically, the H1 heuristic may provide an optimal setup assignment for one single product, becoming sub-optimal in the cases of multiple products, and the third move can rectify this.

4. Computational results

The two heuristics and the local search moves are tested on the instances proposed by Bajwa et al. (2016a) and Bajwa et al. (2016b). These instances are based on real-world data. For the second instance, setup costs have been added to adapt the data to our problem.

The data related to each instance is detailed in Table 1.

The differences between the two instances lie mostly in the demand parameters α_j and β_j affecting the price range of the solutions. In addition to these parameters, several production capacities ranging between 40 and 110 are tested.

The two instances are tested on four demand scenarios, given by Table 2. These scenarios impact the demand seasonality γ_{jt} for each product. The first scenario represents the case without any seasonality. The second scenario has a low seasonality at the beginning of the horizon, and a high seasonality at the end. The third scenario is the opposite of the second one, with a decreasing seasonality. Finally, the fourth scenario is a mix between the second and third scenarios, with the seasonality depending on the product.

Table 3 illustrates the effect of the improvement of the model when using LINGO

	α_j	β_j	v_j	c_{jt}	h_{jt}	a_{jt}
Instance 1	500	1.9	1.0	1.6	0.02	8.5
	400	1.6	1.0	1.3	0.05	4.5
	600	2.5	1.0	1.5	0.04	7.5
Instance 2	20000	3.5	1.0	3.0	0.035	10.5
	18000	4.0	1.0	3.0	0.035	4.5
	800	5.5	1.0	1.1	0.013	3.5

Table 1: Data for each instance (Bajwa et al. (2016a), Bajwa et al. (2016b))

t	1	2	3	4	5	6
Scenario 1	0.1667	0.1667	0.1667	0.1667	0.1667	0.1667
	0.1667	0.1667	0.1667	0.1667	0.1667	0.1667
	0.1667	0.1667	0.1667	0.1667	0.1667	0.1667
Scenario 2	0.1	0.1	0.1	0.2	0.2	0.3
	0.1	0.1	0.1	0.2	0.2	0.3
	0.1	0.1	0.1	0.2	0.2	0.3
Scenario 3	0.3	0.2	0.2	0.1	0.1	0.1
	0.3	0.2	0.2	0.1	0.1	0.1
	0.3	0.2	0.2	0.1	0.1	0.1
Scenario 4	0.3	0.2	0.2	0.1	0.1	0.1
	0.3	0.2	0.2	0.1	0.1	0.1
	0.1	0.1	0.1	0.2	0.2	0.3

Table 2: Demand scenarios Bajwa et al. (2016a)

		Model P_0	Model P_1
Instances without a feasible solution		10/64	0/64
Average objective value	Instance 1	166.69	224.70
	Instance 2	69.80	211.46

Table 3: Comparison of the efficiency of LINGO solver on models P_0 and P_1

software to solve it. The first row represents the number of instances for which LINGO software found no feasible solution. The second and third rows show the average objective function value obtained when solving the instances of type 1 Bajwa et al. (2016a) or type 2 Bajwa et al. (2016b).

The raw results obtained with decreasing capacity are presented in Tables 4 and 5. Within the LINGO column, the solutions presented are those obtained by the solver for model P_1 with a 3,600 s time limit. LINGO is used along with its "global solver" setting. On the tested instances, the solver is not able to guarantee the optimality of its solutions.

The following columns in Tables 4 and 5 represent the results obtained by the two constructive heuristics. The first eight methods (M1 to M8) use the H1 heuristic with four sorting rules. In addition, methods M5 to M8 use three local search moves to improve the results obtained by the H1 heuristic. Finally, methods M9 and M10 correspond to the H2 heuristic, with M10 using a local search. A bold number in the tables represents the best result obtained with the methods developed. Finally, the lines "Number" and "Rank" represent the number of best solutions obtained with each method, and the rank based on these numbers.

Within Table 6, the minimal, average, and maximal gaps (over all scenarios and production capacities) for each heuristic and each instance are presented. These gaps are given by the formula $gap = \frac{z_{LIN} - z_M}{z_{LIN}}$, with z_{LIN} corresponding to the solution obtained by LINGO software and z_M corresponding to the solution obtained by one of the ten methods. The gap represents the percentage of deviation of the heuristic solutions compared to the solutions by LINGO.

In Figure 1, the curves represent the gaps between the heuristic results and the results of the LINGO software. These gaps are averaged over the different demand scenarios, and are plotted for each production capacity.

The top of the figure corresponds to the results for instance 1, and the bottom of the figure corresponds to the results for instance 2. On the left, heuristics are represented alone, and on the right, the results are obtained with the use of heuristics and local search moves.

Table 7 provides the percentage of deviation of the heuristic solutions from the LINGO software solutions for each instance and scenario (averaged over the capacity values).

As the first analysis shown in the tables and figure, there is no method or sorting rule that outperforms all other methods on all instances. Methods M4 and M8 are those performing the best on instance 1. For the second instance, methods M9 and M10 provide the best results. The gaps obtained for the second instances are also greater (not only

	Capacity	LINGO	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
Scenario 1	40	218.01	214.93	210.96	215.15	208.50	215.60	**216.88**	216.84	211.98	208.13	212.40
	50	221.80	217.4	215.91	219.51	217.39	219.46	221.53	**221.75**	217.39	217.39	217.39
	60	224.50	220.08	218.60	222.15	223.01	220.08	224.21	**224.45**	223.01	223.01	223.01
	70	225.81	224.32	219.72	223.48	**225.76**	224.32	225.43	225.62	**225.76**	**225.76**	**225.76**
	80	227.30	**227.25**	223.20	224.01	**227.25**	226.42	226.42	226.09	**227.25**	226.48	226.48
	90	229.17	**229.12**	225.80	223.91	**229.12**	228.96	228.96	**229.12**	**229.12**	226.36	226.36
	100	230.61	230.09	227.36	224.89	230.09	230.09	**230.56**	230.09	230.09	225.39	227.32
	110	231.41	230.32	228.05	225.12	230.32	230.32	**231.36**	230.32	230.32	223.78	228.11
Scenario 2	40	221.24	215.00	214.13	217.97	217.02	215.02	214.90	**218.02**	217.07	217.07	217.07
	50	224.26	215.48	216.69	217.19	221.51	217.19	218.74	217.19	**221.70**	**221.70**	**221.70**
	60	225.28	217.72	216.61	219.35	224.03	217.72	223.06	222.15	**224.23**	**224.23**	**224.23**
	70	225.84	222.38	218.77	221.36	224.48	222.38	222.38	224.24	224.53	**225.24**	**225.24**
	80	226.38	225.67	222.23	222.44	225.67	225.67	225.67	**225.84**	225.67	225.51	225.51
	90	227.85	**227.85**	224.44	221.40	**227.85**	**227.85**	**227.85**	224.91	**227.85**	227.30	227.30
	100	229.10	**229.10**	225.62	222.20	**229.10**	**229.10**	**229.10**	**229.10**	**229.10**	228.18	228.18
	110	229.55	**228.97**	225.46	222.19	**228.97**	**228.97**	**228.97**	**228.97**	**228.97**	228.33	228.33
Scenario 3	40	213.95	210.37	207.54	207.69	**212.19**	210.37	210.67	208.80	**212.19**	208.80	208.80
	50	218.58	216.99	208.64	216.99	**218.40**	218.39	212.76	218.39	**218.40**	218.39	218.39
	60	224.15	222.42	216.06	222.42	222.11	**224.15**	219.58	**224.15**	222.11	**224.15**	**224.15**
	70	227.03	224.93	221.52	224.93	224.68	**227.03**	224.68	**227.03**	224.68	**227.03**	**227.03**
	80	228.41	**228.41**	225.44	225.88	**228.41**	**228.41**	**228.41**	227.83	**228.41**	227.83	227.83
	90	231.04	**231.04**	228.13	226.59	**231.04**	**231.04**	**231.04**	228.54	**231.04**	228.43	228.43
	100	232.73	**232.73**	229.78	227.63	**232.73**	**232.73**	**232.73**	**232.73**	**232.73**	228.24	228.24
	110	233.64	**233.64**	230.55	228.57	**233.64**	**233.64**	**233.64**	**233.64**	**233.64**	228.44	230.00
Scenario 4	40	218.07	213.80	203.97	208.97	**216.44**	213.79	212.15	213.79	**216.44**	212.30	215.89
	50	223.32	213.55	210.58	214.91	**221.89**	214.93	210.58	220.05	**221.89**	217.49	217.49
	60	226.11	218.23	217.46	219.08	**225.29**	218.23	217.46	223.53	**225.29**	223.23	223.23
	70	227.29	223.06	222.41	221.77	**227.14**	223.06	222.41	225.38	**227.14**	226.08	226.08
	80	228.06	226.53	225.83	223.27	227.03	226.53	225.83	225.97	227.03	**227.23**	**227.23**
	90	229.25	**228.87**	228.02	223.74	**228.87**	**228.87**	228.02	**228.87**	**228.87**	228.02	228.02
	100	230.29	**230.29**	229.18	225.20	**230.29**	**230.29**	229.70	**230.29**	**230.29**	229.18	229.70
	110	231.02	230.90	229.49	225.85	230.90	230.90	**231.02**	230.90	230.90	229.49	**231.02**
Number			$\frac{11}{32}$	$\frac{0}{32}$	$\frac{0}{32}$	$\frac{18}{32}$	$\frac{11}{32}$	$\frac{11}{32}$	$\frac{13}{32}$	$\frac{20}{32}$	$\frac{7}{32}$	$\frac{8}{32}$
Rank			4	-	-	2	4	4	3	1	6	5

Table 4: Instance 1

	Capacity	LINGO	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
Scenario 1	40	194.11	180.22	187.92	170.94	178.57	183.98	**188.42**	181.07	179.45	185.34	187.15
	50	204.85	191.92	198.93	185.19	192.11	196.26	198.93	194.14	192.81	198.75	**200.14**
	60	210.73	200.25	205.50	195.65	201.33	203.94	205.92	202.51	201.79	207.41	**208.01**
	70	213.56	206.42	209.40	203.28	207.60	210.52	210.67	207.80	208.20	**212.67**	**212.67**
	80	216.48	210.94	212.00	208.73	211.63	214.65	212.00	211.55	212.62	**215.18**	**215.18**
	90	218.12	214.04	212.75	212.46	209.78	**216.87**	215.81	214.12	211.12	213.31	214.57
	100	218.91	215.77	214.74	214.75	211.60	**218.18**	216.71	214.75	211.60	214.97	215.92
	110	219.50	216.60	212.00	213.05	209.07	**218.74**	215.23	215.80	213.57	215.74	216.49
Scenario 2	40	199.51	191.38	189.37	169.72	191.54	192.45	189.37	181.18	191.54	192.10	**197.00**
	50	205.79	199.73	200.31	184.11	**202.44**	200.25	200.31	194.27	**202.44**	199.86	201.95
	60	211.70	204.89	206.70	194.70	**208.82**	204.89	206.93	202.78	**208.82**	207.47	207.47
	70	214.61	207.70	210.47	202.47	212.32	207.70	211.43	208.13	**212.94**	211.37	211.37
	80	216.01	208.75	209.16	208.07	214.50	210.12	214.06	211.88	**215.82**	213.27	213.68
	90	218.30	207.90	211.64	212.15	214.62	209.01	215.23	214.47	**216.94**	210.94	213.22
	100	218.90	207.84	213.17	214.86	212.36	213.17	**216.27**	214.86	215.79	211.68	212.87
	110	219.03	208.37	212.80	211.31	213.73	214.00	215.91	212.90	**216.12**	212.08	213.29
Scenario 3	40	181.44	174.23	177.68	169.89	**181.36**	180.30	177.68	172.66	**181.36**	181.18	181.18
	50	194.69	188.56	190.95	184.96	193.66	191.15	190.95	187.12	193.66	**194.32**	**194.32**
	60	202.64	198.85	200.12	196.01	201.61	198.85	200.12	197.64	201.61	**202.94**	**202.94**
	70	208.50	206.04	206.33	204.09	206.62	206.04	206.33	205.28	206.62	**208.50**	**208.50**
	80	213.09	210.81	207.48	209.88	209.34	210.81	208.96	210.72	209.85	**212.52**	**212.52**
	90	216.16	213.80	210.69	213.66	212.28	213.80	211.04	214.41	212.64	**215.37**	**215.37**
	100	218.51	215.99	212.72	215.71	209.15	215.99	215.51	216.67	214.13	**217.25**	**217.25**
	110	219.89	209.67	212.39	214.39	211.44	212.39	214.59	215.40	218.20	**218.34**	**218.34**
Scenario 4	40	192.57	174.46	185.87	176.45	**190.17**	174.72	185.87	177.45	**190.17**	187.39	187.39
	50	201.73	188.81	198.23	189.75	**200.98**	189.25	198.23	190.74	**200.98**	200.80	200.80
	60	211.43	199.11	205.47	199.34	207.40	200.25	205.47	200.27	207.40	**209.07**	**209.07**
	70	214.9	206.30	209.27	206.14	211.23	207.99	209.27	207.00	212.07	**213.76**	**213.76**
	80	217.26	211.00	212.16	210.83	214.05	213.11	215.25	211.60	215.45	**216.51**	**216.51**
	90	218.31	213.90	214.99	213.87	216.11	216.11	216.92	214.69	217.12	**217.93**	**217.93**
	100	220.37	215.99	213.90	215.49	214.76	218.01	217.92	216.60	218.03	**218.60**	**218.60**
	110	221.10	209.10	215.79	213.83	211.45	215.84	**220.76**	217.41	217.85	217.51	217.51
Number			$\frac{0}{32}$	$\frac{0}{32}$	$\frac{0}{32}$	$\frac{5}{32}$	$\frac{3}{32}$	$\frac{3}{32}$	$\frac{0}{32}$	$\frac{9}{32}$	$\frac{14}{32}$	$\frac{17}{32}$
Rank			-	-	-	5	6	6	-	3	2	1

Table 5: Instance 2

	Gaps	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
Instance 1	min	0.00%	0.48%	0.72%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
	average	1.11%	2.41%	2.14%	0.56%	0.96%	1.04%	0.65%	0.50%	1.15%	0.90%
	max	4.38%	6.47%	4.17%	4.36%	3.76%	5.71%	3.15%	2.77%	4.53%	2.61%
Instance 2	min	1.07%	1.04%	1.15%	0.04%	0.33%	0.15%	0.81%	0.04%	-0.15%	-0.15%
	average	3.64%	2.55%	4.79%	2.49%	2.60%	1.87%	3.25%	1.85%	1.46%	1.18%
	max	9.40%	5.09%	14.93%	8.00%	9.27%	5.09%	9.19%	7.55%	4.52%	3.58%

Table 6: Average, min, and max deviation from LINGO solver solutions

the average gaps but also the minimal and maximal gaps) than those obtained for the first instance. In addition, the difference between the sorting rules on instance 1 are not the same as that for the second instance. The local search slightly increases the results obtained by both heuristics, but performs better for far-from-optimum solutions than for close-to-optimum solutions.

Sorting rule S4 is the most efficient rule of the four. This efficiency may be explained by its synergy with the way the heuristic assigns the setup values. As the algorithm assigns less setups to the first products in the sorted list, the rule S4 helps the algorithm decrease the total inventory and setup cost.

For some parameter sets, the best solutions are all found using the same sorting rule, which is due to the resolution by decreasing the production capacity. This means that an optimal configuration for a given production capacity may remain optimal for smaller capacity values.

From Figure 1, it should be noted that the heuristics are particularly efficient for medium and high production capacity values, but less for low capacity values. The low capacities force the solutions to have a number of assigned setups. It is then difficult for the algorithms to decrease the total number of setups. For high production capacity values, the solutions tend to have few assigned setups, and the removal or addition of a setup usually has unexpected consequences on the production of other products.

Overall, by taking the best results between the H1 and H2 heuristics, and by using a local search, the worst solution obtained over every instance and parameter having a percentage of deviation of only 2.93%, and on average, for each instance and parameter set, the best solution obtained by either the H1 or H2 heuristic has a percentage of deviation of 0.49% compared to the LINGO solver.

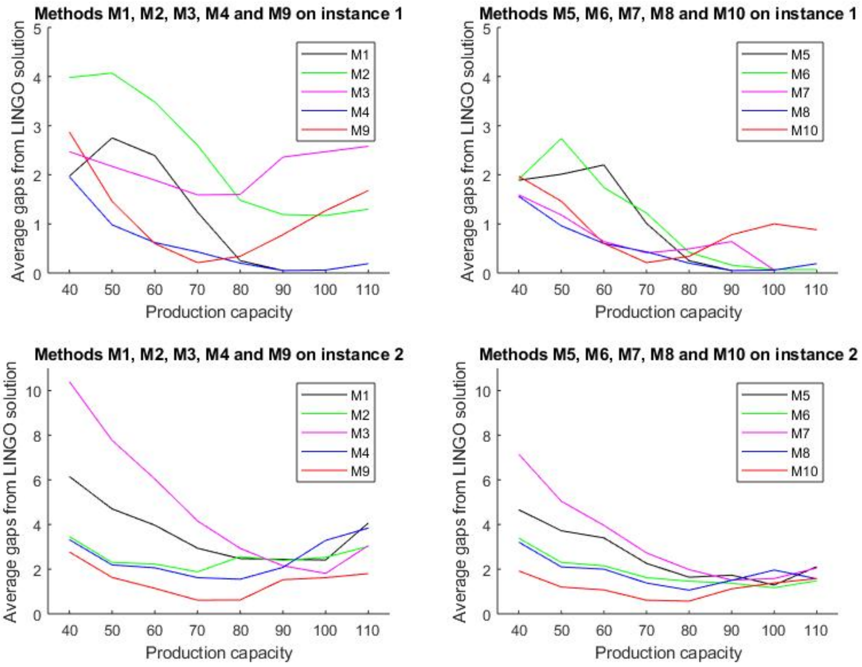

Figure 1: Average deviation for the ten heuristic methods compared to LINGO

		M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
Instance 1	Scenario 1	0.85%	2.17%	1.67%	0.97%	0.69%	0.18%	0.24%	0.77%	1.79%	1.21%
	Scenario 2	1.52%	2.53%	2.50%	0.61%	1.43%	1.05%	1.06%	0.58%	0.66%	0.66%
	Scenario 3	0.51%	2.34%	1.59%	0.36%	0.22%	0.91%	0.48%	0.36%	1.00%	0.92%
	Scenario 4	1.57%	2.60%	2.80%	0.31%	1.49%	2.02%	0.82%	0.31%	1.14%	0.82%
Instance 2	Scenario 1	3.62%	2.55%	5.56%	4.45%	2.02%	1.94%	3.28%	3.90%	1.98%	1.57%
	Scenario 2	3.94%	2.97%	6.38%	1.98%	3.07%	2.05%	3.79%	1.40%	2.65%	1.93%
	Scenario 3	2.27%	2.19%	2.90%	1.71%	1.53%	1.79%	2.19%	0.99%	0.26%	0.26%
	Scenario 4	4.74%	2.48%	4.32%	1.83%	3.79%	1.68%	3.74%	1.09%	0.97%	0.97%

Table 7: Gaps for each instance and scenario (averaged over the production capacity values)

5. Conclusion and future research

In this study, we considered the problem of optimizing, simultaneously, production and pricing decisions while considering a capacity constraint limitation, multi-period time horizon, and multiple products to maximize profits.

As the first contribution, we propose a non-linear mathematical formulation solved using LINGO software. On average, this mathematical formulation improves the quality of the solutions obtained with LINGO software by 84.4%.

The second contribution consists of new constructive methods to solve this problem. These methods have been proven to be efficient in providing high-quality results on literature-based instances, reaching a worst deviation of 2.93% from the solver's solutions.

The results presented in this study have found for an isoelastic demand function and can easily be adapted to a linear demand function.

As a future extension of this study, an exact method can be developed to replace the use of LINGO software, and provide a proof of optimality for the obtained solutions.

Finally, the price of a product is not the only parameter impacting the customer's choice when buying. The guaranteed lead-time can also significantly influence their choices, and for some product categories, customers may have a reference price in mind from their previous purchases. These parameters may be included in the studied model to achieve a more accurate representation of industrial problems.

Acknowledgment

This research was supported by the Industrial Chair Connected-Innovation (https://chaire-connected-innovation.fr)

References

M. Kienzler, Value-based pricing and cognitive biases: An overview for business markets, Industrial Marketing Management 68 (2018) 86–94, doi:10.1016/j.indmarman.2017.09.028.

B. C. Smith, J. F. Leimkuhler, R. M. Darrow, Yield management at American airlines, interfaces 22 (1) (1992) 8–31, doi:10.1287/inte.22.1.8.

G. Bitran, S. V. Mondschein, An application of yield management to the hotel industry considering multiple day stays, Operations research 43 (3) (1995) 427–443, doi:10.1287/opre.43.3.427.

A. M. Bandalouski, M. Y. Kovalyov, E. Pesch, S. A. Tarim, An overview of revenue management and dynamic pricing models in hotel business, RAIRO-Operations Research 52 (1) (2018) 119–141, doi: 10.1051/ro/2018001.

R. Bharill, N. Rangaraj, Revenue management in railway operations: A study of the Rajdhani Express, Indian Railways, Transportation Research Part A: Policy and Practice 42 (9) (2008) 1195–1207, doi: 10.1016/j.tra.2008.03.007.

L. Chen, A. Mislove, C. Wilson, An empirical analysis of algorithmic pricing on amazon marketplace, in: Proceedings of the 25th International Conference on World Wide Web, International World Wide Web Conferences Steering Committee, 1339–1349, doi:10.1145/2872427.2883089, 2016.

G. Bitran, R. Caldentey, An overview of pricing models for revenue management, Manufacturing & Service Operations Management 5 (3) (2003) 203–229, doi:10.1287/msom.5.3.203.16031.

W. Elmaghraby, P. Keskinocak, Dynamic pricing in the presence of inventory considerations: Research overview, current practices, and future directions, Management science 49 (10) (2003) 1287–1309, doi:10.1287/mnsc.49.10.1287.17315.

M. Chen, Z.-L. Chen, Recent developments in dynamic pricing research: multiple products, competition, and limited demand information, Production and Operations Management 24 (5) (2015) 704–731, doi: 10.1111/poms.12295.

L. M. Chan, Z. M. Shen, D. Simchi-Levi, J. L. Swann, Coordination of pricing and inventory decisions: A survey and classification, in: Handbook of quantitative supply chain analysis, Springer, 335–392, doi:10.1007/978-1-4020-7953-5_9, 2004.

X. Chen, D. Simchi-Levi, Pricing and inventory management, The Oxford handbook of pricing management (2012) 784–824.

E. Adida, G. Perakis, Dynamic pricing and inventory control: Uncertainty and competition, Operations Research 58 (2) (2010) 289–302, doi:10.1287/opre.1090.0718.

A. Lamas, P. Chevalier, Joint dynamic pricing and lot-sizing under competition, European Journal of Operational Research 266 (3) (2018) 864–876, doi:10.1016/j.ejor.2017.10.026.

L. Chen, Y. Feng, G. Hao, M. F. Keblis, Optimal pricing and sequencing in a make-to-order system with batch demand, IEEE Transactions on Automatic Control 60 (6) (2015) 1455–1470, doi:10.1109/TAC. 2014.2372910.

Y. Yu, B. Shou, Y. Ni, L. Chen, Optimal production, pricing, and substitution policies in continuous review production-inventory systems, European Journal of Operational Research 260 (2) (2017) 631–649, doi:10.1016/j.ejor.2017.01.012.

J. Thomas, Price-production decisions with deterministic demand, Management Science 16 (11) (1970) 747–750, doi:10.1287/mnsc.16.11.747.

K. K. Haugen, A. Olstad, B. I. Pettersen, The profit maximizing capacitated lot-size (PCLSP) problem, European Journal of Operational Research 176 (1) (2007) 165–176, doi:10.1016/j.ejor.2005.08.001.

M. Mahmoudzadeh, S. J. Sadjadi, S. Mansour, Robust optimal dynamic production/pricing policies in a closed-loop system, Applied Mathematical Modelling 37 (16) (2013) 8141–8161, doi:10.1016/j.apm.

2013.03.008.

N. Bajwa, C. R. Sox, R. Ishfaq, Coordinating pricing and production decisions for multiple products, Omega 64 (2016a) 86–101, doi:10.1016/j.omega.2015.11.006.

Y. Ouazene, F. Yalaoui, R. Kelly, T. Idjeraoui, Coordination and optimization of dynamic pricing and production decisions, in: Computational Intelligence (SSCI), 2017 IEEE Symposium Series on, IEEE, 1–6, doi:10.1109/SSCI.2017.8285252, 2017.

J. Huang, M. Leng, M. Parlar, Demand functions in decision modeling: A comprehensive survey and research directions, Decision Sciences 44 (3) (2013) 557–609, doi:10.1111/deci.12021.

R. Maihami, B. Karimi, Optimizing the pricing and replenishment policy for non-instantaneous deteriorating items with stochastic demand and promotional efforts, Computers & Operations Research 51 (2014) 302–312, doi:10.1016/j.cor.2014.05.022.

L. Feng, J. Zhang, W. Tang, Dynamic joint pricing and production policy for perishable products, International Transactions in Operational Research 25 (6) (2018) 2031–2051, doi:10.1111/itor.12239.

X. Wu, H. Xu, C. Chu, J. Zhang, et al., Dynamic lot-sizing models with pricing for new products, European Journal of Operational Research 260 (1) (2017) 81–92, doi:10.1016/j.ejor.2016.12.008.

Y. Li, G. Nemhauser, M. Savelsbergh, Pricing for production and delivery flexibility in single-item lot-sizing, Computers & Operations Research 39 (12) (2012) 3408–3419, doi:10.1016/j.cor.2012.05.003.

M. Öner-Közen, S. Minner, Dynamic pricing, leadtime quotation and due date based priority dispatching, International Journal of Production Research (2017) 1–13doi:10.1080/00207543.2017.1397791.

A.-S. Albana, Pricing decision and lead time quotation in supply chains with an endogenous demand sensitive to lead time and price, Ph.D. thesis, Université Grenoble Alpes, 2018.

A. V. den Boer, Dynamic pricing and learning: historical origins, current research, and new directions, Surveys in operations research and management science 20 (1) (2015) 1–18, doi:10.1016/j.sorms.2015.03.001.

M. Fisher, S. Gallino, J. Li, Competition-based dynamic pricing in online retailing: A methodology validated with field experiments, Management Science 64 (6) (2017) 2496–2514, doi:10.1287/mnsc.2017.2753.

A. Herbon, Dynamic pricing vs. acquiring information on consumers' heterogeneous sensitivity to product freshness, International Journal of Production Research 52 (3) (2014) 918–933, doi:10.1080/00207543.2013.843800.

M. A. Duran, I. E. Grossmann, An outer-approximation algorithm for a class of mixed-integer nonlinear programs, Mathematical programming 36 (3) (1986) 307–339, doi:10.1007/BF02592064.

A. M. Geoffrion, Generalized benders decomposition, Journal of optimization theory and applications 10 (4) (1972) 237–260.

P. Kesavan, R. J. Allgor, E. P. Gatzke, P. I. Barton, Outer approximation algorithms for separable nonconvex mixed-integer nonlinear programs, Mathematical Programming 100 (3) (2004) 517–535.

M. L. Bergamini, I. Grossmann, N. Scenna, P. Aguirre, An improved piecewise outer-approximation algorithm for the global optimization of MINLP models involving concave and bilinear terms, Computers & Chemical Engineering 32 (3) (2008) 477–493.

H. M. Wagner, T. M. Whitin, Dynamic version of the economic lot size model, Management science 5 (1) (1958) 89–96, doi:10.1287/mnsc.5.1.89.

N. Bajwa, B. Fontem, C. R. Sox, Optimal product pricing and lot sizing decisions for multiple products with nonlinear demands, Journal of Management Analytics 3 (1) (2016b) 43–58, doi:10.1080/23270012.2015.1121118.