Solving a Production-Routing Problem with Price-Dependent Demand Using an Outer Approximation Method

Citation for published version (APA):

Torkaman, S., Akbari Jokar, M. R., Mutlu, N. R., & van Woensel, T. (2020). Solving a Production-Routing Problem with Price-Dependent Demand Using an Outer Approximation Method. *Computers & Operations Research*, 123, [105019]. <https://doi.org/10.1016/j.cor.2020.105019>

Document license:
TAVERNE

DOI:
[10.1016/j.cor.2020.105019](https://doi.org/10.1016/j.cor.2020.105019)

Document status and date:
Published: 01/11/2020

Document Version:
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Solving a production-routing problem with price-dependent demand using an outer approximation method

Somayeh Torkaman^a, Mohammad Reza Akbari Jokar^{a,*}, Nevin Mutlu^b, Tom Van Woensel^b

^a Industrial Engineering Department, Sharif University of Technology, Tehran, Iran

^b School of Industrial Engineering, Eindhoven University of Technology, Eindhoven, The Netherlands

ARTICLE INFO

Article history:

Received 30 September 2019

Revised 23 March 2020

Accepted 1 June 2020

Available online 8 June 2020

Keywords:

Nonlinear programming

Production-routing

Pricing

Outer approximation

ABSTRACT

A production-routing problem with price-dependent demand (PRP-PD) is studied in this paper. Demand follows a general convex, differentiable, continuous and strictly decreasing function in price. The problem is modeled as a mixed integer nonlinear program (MINLP). Two Outer Approximation (OA) based algorithms are developed to solve the PRP-PD. The efficiency of the proposed algorithms in comparison with commercial MINLP solvers is demonstrated. The computational results show that our basic OA algorithm outperforms the commercial solvers both in solution quality and in computational time aspects. On the other hand, our extended (two-phase) OA algorithm provides near-optimal solutions very efficiently, especially for large problem instances. These findings prevail both for linear and for nonlinear demand functions. Additional sensitivity analyses are conducted to investigate the impact of different problem parameters on the optimal solution. The results show that the manufacturer should give higher priority to the retailer who has lower price sensitivity and who is closer to the manufacturer. Another takeaway is that a larger market size and a lower price sensitivity lead to more profit.

© 2020 Published by Elsevier Ltd.

1. Introduction

One of the common challenges that manufacturers face in today's business environment is matching demand with operational capabilities (Lapide, 2006). Due to, for example, fluctuations in raw material costs (which is common in commodity markets), volatile exchange rates, and labor conditions such as strikes, manufacturers can experience fluctuations in their production and logistics operations with respect to cost and capacity (Bizfluent, 2017). In addition to the economic oscillations, other factors, such as fixed costs for production and transportation, capacity restrictions due to varying production mix and excess capacity in low-peak demand periods, can exacerbate the problem (Insight2profit, 2019). In such settings, manufacturers would need to use effective marketing mechanisms (such as pricing) to align demand with their operational capabilities and costs, as maintaining profitability hinges on an integrated sales and operations planning (Forbes, 2015).

In this paper, we develop a framework to study a manufacturer's integrated decision-making with respect to marketing (i.e., pricing) and operations decisions. The setting we are considering is an extension of the traditional production routing problem (PRP), which aims to optimize the production, inventory, and routing

decisions of a manufacturer who serves multiple retailers with exogenous demand in a multi-period setting. Similar to the PRP, we consider a manufacturer that produces a single type of product in a capacitated plant and distributes it to multiple retailers via a homogenous fleet of vehicles in a centralized setting. Contrary to the existing works on PRP, which consider exogenous demand with a linear, cost-minimizing objective function, the demand in our work is endogenous and price-dependent. As a result, by leveraging price to control demand at each retailer, the manufacturer can effectively create the order volume that can be supported by the operations team when it comes to production, inventory, and delivery. This results in a profit-maximizing objective function for the PRP with price-dependent demand (we call this new problem PRP-PD in the remainder of this paper).

In our model, we assume a general demand function, which is convex, continuous, differentiable and strictly decreasing in price (see Bajwa et al., 2016). This aspect leads to a nonlinear objective function and a number of nonlinear constraints. As a result, the solution techniques that are used to solve the PRP do not directly apply to the proposed PRP-PD, which is a more general form of PRP. Hence, we adopt an Outer Approximation (OA) method to solve this NP-hard problem. The OA method is an iterative method in which a restricted version and a relaxed version of the main problem, called the primal problem and the master problem, respectively, are solved in each iteration to generate lower and

* Corresponding author.

upper bounds for the main problem. These bounds converge to an optimal solution in a finite number of iterations (Floudas, 1995).

Our contributions to the literature are as follows. Firstly, we provide a general formulation for the PRP-PD. This formulation is a mixed integer nonlinear program (MINLP) with a profit-maximizing objective and a number of nonlinear constraints. Secondly, we develop efficient algorithms to provide exact as well as near-optimal solutions for the PRP-PD. The computational results show that our basic OA algorithm outperforms commercial solvers in both solution quality and computational time aspects. On the other hand, our two-phase OA algorithm provides near-optimal solutions very efficiently, especially for large problem instances that commercial solvers cannot solve. These results prevail both for linear and nonlinear demand functions. Lastly, we examine the impact of various problem parameters on the optimal solution, and provide managerial insights on how different manufacturer and retailer characteristics affect the optimal decisions.

The paper is organized as follows. In Section 2, we provide a brief review of the related literature. In Section 3, we introduce the PRP-PD formulation and notation. In Section 4, we introduce the proposed OA algorithms to solve the PRP-PD. In Section 5, we present the computational results of the proposed OA methods with a sensitivity analysis; and finally, in Section 6 we conclude the paper and present potential future research directions.

2. Literature review

PRP-PD builds onto the well-known PRP by incorporating price-dependent demand into the problem formulation through a MINLP. In addition to the PRP literature, the research on the production pricing problem is also a relevant research stream for the PRP-PD, due to the integrated lot-sizing and pricing decisions. In the following, we provide an overview of the most relevant works on the PRP and the production pricing problem, along with a brief discussion on the solution methodologies for MINLP problems.

The PRP aims to optimize the production, inventory, and routing decisions of a manufacturer who serves multiple retailers with exogenous demand. PRP applies to centralized settings as well as decentralized settings where vendor managed inventory is implemented. Since its introduction by Chandra (1993), the PRP has gained a lot of attention in the field. Due to the complexity of the problem, which is NP-hard, the research on this topic mainly focuses on heuristic methods such as branch and price based heuristics (e.g. Bard and Nananukul, 2009, 2010; Qiu et al., 2017), mixed integer programming (MIP) based heuristics (e.g. Absi et al., 2014; Brahimi and Aouam, 2016; Zhang et al., 2017; Solyali and Sural, 2017; Russell, 2017; Miranda et al., 2018; Li et al., 2019), Lagrangian relaxation based heuristics (Fumero and Vercellis, 1999; Solyali and Sural, 2009), and decomposition heuristics (Chandra, 1993; Chandra and Fisher, 1994; Lei et al., 2006; Boudia et al., 2008; Çetinkaya et al., 2009; Vahdani et al., 2017; Chitsaz et al., 2019; Avci and Yildiz, 2019). More recently, there has been interest in developing exact algorithms for the PRP. Ruokokoski et al. (2010) developed a branch and cut (B&C) method to solve the PRP with a single uncapacitated vehicle and without production capacity. Archetti et al. (2011) applied B&C methods to solve the PRP under order-up-to level (OU) and maximum level (ML) inventory replenishment policies. Adulyasak et al. (2014) used the B&C method of Archetti et al. (2011) for the PRP with multiple vehicles under ML and OU policies. They concluded that incorporating the vehicle index in the formulation provides a model that performs better in finding an optimal solution. Adulyasak et al. (2015a) studied the PRP with uncertain demand. A Benders decomposition method was developed to solve for the set up and routing decisions in the first stage, and production and delivery quantities in the subsequent stages of the proposed

stochastic optimization model. Dayarian and Desaulniers (2019) proposed an exact branch-price-and cut algorithm to solve the PRP with time windows. They also avoided the complete enumeration of the search tree to develop a heuristic in order to tackle the large instances and then proposed two more matheuristics based on this heuristic. We refer the reader to Díaz-Madroñero et al. (2015) and Adulyasak et al. (2015b) for a comprehensive overview of the PRP characteristics, formulations, and solution methods. Note, however, that in all of the aforementioned works, demand is exogenous. We contribute to this stream of research by providing a mathematical formulation for the PRP with price-dependent demand. We also develop two efficient algorithms that provide optimal and near-optimal solutions for this new problem.

A seminal paper in the production pricing problem literature includes Whitin (1955) which defined price as a decision variable in an economic order quantity problem. Thomas (1970) extended the problem as a multi-period production pricing problem and showed that it is possible to optimize the prices in periods between the setups, independently. Kunreuther and Richard (1971) studied the same problem and illustrated the benefits of coordination of pricing and lot-sizing decisions. We refer the reader to Eliashberg and Steinberg (1993) and Yano and Gilbert (2005) for a detailed review of models and solution algorithms used in this literature. For our research, the most relevant works in this domain pertain to multi-period models with deterministic demand functions considering a single product. In this area, Gilbert (1999) and Van and Wagelmans (2006) developed polynomial time algorithms to solve the uncapacitated production pricing problem. Geunes et al. (2006) developed a polynomial time approach to solve the capacitated problem with constant production capacities and piecewise-linear concave revenue functions. Geunes et al. (2009) further extended the previous work for a general class of concave-revenue functions. Other works in this area include Biller et al. (2005), Deng and Yano (2006), Ahn et al. (2007) and Wu et al. (2017). Similar to the aforementioned papers, the PRP-PD formulation incorporates a concave objective function. Such a formulation enables a tractable solution algorithm for a production problem with an ML inventory policy.

The PRP-PD has a nonlinear, concave profit-maximizing objective function. In addition, due to the general form of the demand function, a subset of the constraints may also be nonlinear. Duran and Grossmann (1986) study a special class of MINLP problems and compare the efficiency of different solution algorithms. They illustrate that for problems with special characteristics, the OA algorithm performs better than the Generalized Benders Decomposition (GBD) method. The PRP-PD satisfies these special characteristics (see Section 4), except for the linearity property of the equality constraints. Since nonlinear demand functions result in a number of nonlinear equality constraints in the PRP-PD, we develop a modified OA based algorithm with equality relaxation. We refer the reader to Floudas (1995), Grossmann (2002) and Bonami et al. (2008) for excellent reviews on solution algorithms for MINLP problems.

3. The production routing problem with price-dependent demand (PRP-PD)

The setup of our problem is similar to the general PRP setting that is presented in Boudia et al. (2007, 2008), Bard and Nananukul (2009, 2010) and Adulyasak et al. (2012, 2014). We consider a set $N_c = \{1, \dots, n\}$ of retailers (i.e. nodes) that are being served by a single production plant, identified with node 0 throughout the discrete planning horizon $T = \{1, \dots, |T|\}$. There is a single product, delivered to the retailers via a set of homogenous vehicles $K = \{1, \dots, |K|\}$, each with capacity Q . The travel cost c_{ij_t} is incurred for the transportation of products between any node pair

(i, j) connected by arcs $A = \{(i, j) : i, j \in N = N_C \cup \{0\}, i \neq j\}$ in period $t \in T$. Inventory is stocked both in the production plant and in the retail stores at a unit holding cost of h_{it} in period $t \in T$. The maximum level of inventory that can be held at node $i \in N$ is denoted by L_i . The inventory replenishment of retailers takes place under the ML policy, which dictates that the manufacturer can deliver any quantity to the retailer i as long as the quantity of the delivered items plus the remaining inventory from the previous period does not exceed L_i before the demand is met. We assume that backorders are not allowed, and the initial inventory at each node is zero. Additionally, the manufacturer incurs a fixed setup cost of f_t and variable unit cost of u_t for production in period $t \in T$, which is limited by the production capacity C .

We endogenize the demand in this paper through price. To that end, $d_{it}(p_{it})$ denotes the demand at retailer $i \in N_C$ in period $t \in T$ as a function of the retailer's product price in the same period, p_{it} . We assume that $d_{it}(p_{it})$ follows a general convex, continuous, differentiable and strictly decreasing function in price, p_{it} , leading to a concave revenue function. For technical reasons, we also assume that the demand of each retailer is finite and it approaches to zero when price is sufficiently high. These assumptions are not restrictive and satisfied by some of the most commonly used demand functions in the literature such as linear, constant elasticity, and power demand functions (In the Appendix A.2, we provide the proofs of concavity for revenue functions that include linear and power demand functions, which are used in our computational experiments). In what follows, we present the decision variables and the problem formulation. This is largely inspired by the vehicle index based formulation of the PRP presented in [Adulyasak et al. \(2014\)](#).

Decision variables:

W_t	Production quantity in period $t \in T$;
I_{it}	Inventory amount at node $i \in N$ at the end of period $t \in T$;
y_t	Binary variable that equals 1 if the production set up occurs in period $t \in T$, and equals 0 otherwise;
z_{ikt}	Binary variable that equals 1, if node $i \in N$ is visited by vehicle $k \in K$ in period $t \in T$; and equals 0 otherwise;
x_{ijkt}	Binary variable that equals 1, if node $j \in N$ is visited immediately after node $i \in N$ by vehicle $k \in K$ in period $t \in T$, and equals 0 otherwise;
q_{ikt}	Quantity delivered to retailer $i \in N_C$ by vehicle $k \in K$ in period $t \in T$;
p_{it}	Product price at retailer $i \in N_C$ in period $t \in T$

Formulation:

$$(P) \max Z_p = \sum_{t \in T} \left(\sum_{i \in N_C} p_{it} d_{it}(p_{it}) - \left(u_t W_t + f_t y_t + \sum_{i \in N} h_{it} I_{it} + \sum_{(i,j) \in A} \sum_{k \in K} c_{ijt} x_{ijkt} \right) \right) \quad (1)$$

s.t.

$$I_{0,t-1} + W_t = \sum_{i \in N_C} \sum_{k \in K} q_{ikt} + I_{0t} \quad \forall t \in T \quad (2)$$

$$I_{i,t-1} + \sum_{k \in K} q_{ikt} = d_{it}(p_{it}) + I_{it} \quad \forall i \in N_C, \forall t \in T \quad (3)$$

$$W_t \leq C y_t \quad \forall t \in T \quad (4)$$

$$I_{0t} \leq L_0 \quad \forall t \in T \quad (5)$$

$$I_{i,t-1} + \sum_{k \in K} q_{ikt} \leq L_i \quad \forall i \in N_C, \forall t \in T \quad (6)$$

$$\sum_{i \in N_C} q_{ikt} \leq Q z_{0kt} \quad \forall k \in K, \forall t \in T \quad (7)$$

$$\sum_{k \in K} z_{ikt} \leq 1 \quad \forall i \in N_C, \forall t \in T \quad (8)$$

$$q_{ikt} \leq \min\{L_i, Q\} z_{ikt} \quad \forall i \in N_C, \forall k \in K, \forall t \in T \quad (9)$$

$$\sum_{j \in N} x_{ijkt} + \sum_{j \in N} x_{jik} = 2 z_{ikt} \quad \forall i \in N, \forall t \in T, \forall k \in K \quad (10)$$

$$\sum_{i \in S} \sum_{j \in S} x_{ijkt} \leq \sum_{i \in S} z_{ikt} - z_{ekt} \quad \forall S \subseteq N_C, |S| \geq 2, \forall e \in S, \forall t \in T, \forall k \in K \quad (11)$$

$$W_t, I_{it}, q_{ikt}, p_{it} \geq 0 \quad \forall i \in N, \forall t \in T, \forall k \in K \quad (12)$$

$$y_t, z_{ikt}, x_{ijkt} \in \{0, 1\} \quad \forall i, j \in N, \forall k \in K, \forall t \in T \quad (13)$$

The objective function (1) maximizes the total profit, which equals total revenue less the summation of the production, setup, inventory holding and routing costs across the planning horizon. Constraints (2)–(6) represent the lot-sizing constraints of the problem. Specifically, Constraints (2) and (3) are the inventory balance equations at the production plant and retailers, respectively. Constraints (4) ensure that production can only be performed if production setup occurs, and the production quantity cannot exceed the production capacity. Constraints (5) and (6) limit the delivered amount to the retailers by the inventory capacity of the plant and the retailers, respectively, based on the ML policy. Constraints (7)–(11) are the vehicle routing constraints. Specifically, Constraints (7) set the capacity of each vehicle, and Constraints (8) ensure that each retailer is visited by at most one vehicle (i.e. split deliveries are not allowed). Constraints (9) ensure that the delivery to the retailer can be performed only if the retailer is visited in that time period. Constraints (10) indicate the flow conservation of each vehicle. Constraints (11) represent the subtour elimination constraints. Constraints (12) and (13) introduce the continuous and binary variables of the problem, respectively.

Valid inequalities

To strengthen the formulation of the routing part, we introduce the following valid inequalities into the formulation (see [Archetti et al., 2007, 2011](#)):

$$z_{ikt} \leq z_{0kt} \quad \forall i \in N_C, \forall k \in K, \forall t \in T \quad (14)$$

$$x_{ijkt} \leq z_{ikt} \quad \forall i \in N_C, \forall j \in N, \forall k \in K, \forall t \in T \quad (15)$$

$$x_{ijkt} \leq z_{jkt} \quad \forall i \in N, \forall j \in N_C, \forall k \in K, \forall t \in T \quad (16)$$

Constraints (14) ensure that a vehicle is allowed to visit retailers only in the case of dispatching from the plant. Constraints (15) and (16) state that including a retailer on an edge is allowed only if that retailer is visited. Following [Adulyasak et al. \(2014, 2015a\)](#), we also introduce the following valid symmetry breaking inequalities to further improve the formulation of a multi-vehicle problem:

$$z_{0kt} \geq z_{0,k+1,t} \quad \forall k \in \{1, \dots, |K| - 1\}, \forall t \in T \quad (17)$$

$$\sum_{i=1}^j 2^{(j-i)} z_{ikt} \geq \sum_{i=1}^j 2^{(j-i)} z_{i,k+1,t} \quad \forall j \in N_C, \forall k \in \{1, \dots, |K| - 1\}, \forall t \in T \quad (18)$$

Constraints (17) ensure that vehicle $k + 1$ is allowed to be dispatched only if vehicle k is already dispatched. Constraints (18) uses the lexicographic ordering constraints in order to assign a unique number to each possible route and order the vehicles according to their assigned number.

4. Solution approach

The PRP-PD formulation incorporates a nonlinear, concave profit-maximizing objective function. In addition, due to the general form of the demand function, a subset of the constraints may also be nonlinear. According to [Duran and Grossmann \(1986\)](#), OA algorithm performs better than the GBD method for

problems that satisfy these special characteristics: (1) the continuous and binary variables can be separated in both the objective function and in the constraints, (2) the terms with the binary variables are linear, (3) the revenue follows a concave and continuously differentiable function, (4) the demand follows a convex, continuous, differentiable, and strictly decreasing function in price, and (5) all the equality constraints are linear. Characteristics (1)–(4) are satisfied by the PRP-PD. However, characteristic (5) is not satisfied for nonlinear demand functions. Therefore, we use a modified OA algorithm with equality relaxation, proposed by Kocis and Grossmann (1987). In what follows, we present two algorithms: the basic OA algorithm, and the two-phase OA algorithm.

4.1. The basic OA algorithm

The basic OA algorithm works as follows (see also Kocis and Grossmann, 1987): a restricted version (primal problem) and a relaxed version (master problem) of the main problem are solved iteratively to produce lower and upper bounds, respectively, on the optimal solution of the main problem. These bounds converge to the optimal solution in a finite number of iterations (Duran and Grossmann, 1986). Below, we provide a description of our algorithm and present the pseudocode (in our description, we make references to the respective lines in the pseudocode).

The algorithm starts at iteration $l = 0$ by 1) fixing all the binary variables such that the constraints including only the binary variables remain feasible (when executing our algorithm, we fix all binary variables at zero, see line 1, since doing so always satisfies the aforementioned feasibility requirement in Constraints (8), (10), (11) and (13)–(18)); 2) setting the upper bound (Z_U^0) of the

main problem to $\pi^{\max}$, which can simply be determined by maximizing the revenue function without considering the constraints (line 1); 3) setting the convergence tolerance at ε (line 1); and 4) solving the primal problem (P_p^0), which is a restricted version of the main problem due to the fixed binary variables, to obtain the optimal price p_{it}^0 (line 2) and an initial lower bound Z_L^0 (lines 3 and 4) on the objective value of the main problem. If the primal problem is infeasible (line 5), then the feasibility problem (P_F^0) is solved (line 6) to make the algorithm able to identify a feasible point in next iterations. After these steps, the algorithm enters the cycle in which the master problem and the primal problem are solved iteratively (line 7). In the proposed algorithm a nonlinear program (NLP) solver is used to solve the primal and feasibility problems which are nonlinear.

Iteration $l(l \neq 0)$ (line 8) starts with formulating the outer approximation cuts at the price solution of P_p^{l-1} or P_F^{l-1} (line 9), and adding them to the master problem (P_M^l). Then the master problem, which is a relaxation of the main problem, is solved (line 10) to obtain an upper bound on the objective value of the main problem (line 11). The binary variables obtained from the solution of the master problem are then fed to the primal problem (P_p^l) (line 12). The primal problem is solved with these fixed binary variables (line 13), and a lower bound to the main problem is obtained (lines 14 and 15). If the primal problem is infeasible (line 16), the feasibility problem (P_F^l) is solved (line 17). Then the iteration counter is raised by one (line 8), and this procedure is repeated until the lower and upper bounds converge at the tolerance ε (line 7).

Below, we present the pseudocode of the basic OA algorithm.

```

1 initialize  $l \leftarrow 0$ ,  $Z_U^0 \leftarrow \pi^{\max}$ ,  $Z_L^0 \leftarrow -\infty$ , set  $\varepsilon$ ,  $\bar{x}_{ijkt}^0$ ,  $\bar{z}_{ikst}^0$ ,  $\bar{y}_t^0 = 0$ 
2 solve  $P_p^0$  and obtain optimal price  $p_{it}^0$ 
3 if  $P_p^0$  is feasible then
4    $Z_L^0 \leftarrow Z_{P_p}^0$ 
5 else
6   solve  $P_F^0$  and obtain optimal price  $p_{it}^0$ 
7 while  $(Z_U^l - Z_L^l)/Z_U^l > \varepsilon$  do
8    $l \leftarrow l + 1$ 
9   use  $p_{it}^{l-1}$  to add linear approximation cuts to  $P_M^l$ 
10  solve  $P_M^l$  optimally and obtain optimal binary variables  $x_{ijkt}^l$ ,  $z_{ikst}^l$ ,  $y_t^l$ 
11   $Z_U^l \leftarrow \min(Z_U^{l-1}, Z_{P_M}^l)$ 
12   $\bar{x}_{ijkt}^l = x_{ijkt}^l$ ,  $\bar{z}_{ikst}^l = z_{ikst}^l$ ,  $\bar{y}_t^l = y_t^l$ 
13  solve  $P_p^l$  and obtain optimal price  $p_{it}^l$ 
14  if  $P_p^l$  is feasible then
15     $Z_L^l \leftarrow \max(Z_L^{l-1}, Z_{P_p}^l)$ 
16  else
17    solve  $P_F^l$  and obtain optimal price  $p_{it}^l$ 
18 Output ( $W_t, I_{it}, y_t, z_{ikst}, x_{ijkt}, q_{ikst}, p_{it}$ ),  $Z_P, Z_U$ 

```

Below, we describe the primal problem, the feasibility problem, and the master problem in detail.

4.2. Primal problem

As explained above, the primal problem that is solved in each iteration l , is a restricted version of the main problem (P) with a fixed set of values for the binary variables, i.e. $\bar{y}_{it}^l$, $\bar{x}_{ijkt}^l$ and $\bar{z}_{ikt}^l$ (the bar notation denotes that the corresponding variables are fixed). The primal problem is a NLP and is formulated as follows:

$$(P_P^l) \max Z_{P_P}^l = \sum_{t \in T} \sum_{i \in N_C} p_{it} d_{it}(p_{it}) - \sum_{t \in T} \left(u_t W_t + f_t \bar{y}_{it}^l + \sum_{i \in N} h_{it} I_{it} + \sum_{(i,j) \in A} \sum_{k \in K} c_{ijt} \bar{x}_{ijkt}^l \right) \quad (19)$$

s.t.

$$I_{0,t-1} + W_t = \sum_{i \in N_C} \sum_{k \in K} q_{ikt} + I_{0t} \quad \forall t \in T \quad (20)$$

$$I_{i,t-1} + \sum_{k \in K} q_{ikt} = d_{it}(p_{it}) + I_{it} \quad \forall i \in N_C, \forall t \in T \quad (21)$$

$$W_t \leq C \bar{y}_t^l \quad \forall t \in T \quad (22)$$

$$I_{0t} \leq L_0 \quad \forall t \in T \quad (23)$$

$$I_{i,t-1} + \sum_{k \in K} q_{ikt} \leq L_i \quad \forall i \in N_C, \forall t \in T \quad (24)$$

$$\sum_{i \in N_C} q_{ikt} \leq Q \bar{z}_{0kt}^l \quad \forall k \in K, \forall t \in T \quad (25)$$

$$q_{ikt} \leq \min \{L_i, Q\} \bar{z}_{ikt}^l \quad \forall i \in N_C, \forall k \in K, \forall t \in T \quad (26)$$

$$W_t, I_{it}, q_{ikt}, p_{it} \geq 0 \quad \forall i \in N, \forall t \in T, \forall k \in K \quad (27)$$

Based on the sign of Lagrange multipliers $\lambda_{i,t}^l$ associated with the equality Constraints (21), nonlinear equalities are relaxed into inequalities in the master problem. These multipliers are the dual (shadow) prices associated with the equality constraints, and the values of these multipliers are provided by the NLP solver that we use to solve the primal problem. Using these multipliers, the elements $\tau_{i,t}^l$ are defined as:

$$\tau_{i,t}^l = \begin{cases} -1 & \lambda_{i,t}^l > 0 \\ 1 & \lambda_{i,t}^l < 0 \\ 0 & \lambda_{i,t}^l = 0 \end{cases} \quad \forall i \in N_C, \forall t \in T \quad (28)$$

Let $H(I_{i,t-1}, \bar{q}_{it}, p_{it}, I_{i,t}) \equiv I_{i,t-1} + \sum_{k \in K} q_{ikt} - d_{it}(p_{it}) - I_{i,t}$, where $\bar{q}_{it} \equiv (q_{ikt})_{k \in K}$. Then, Constraint (21) can be reformulated as $H(I_{i,t-1}, \bar{q}_{it}, p_{it}, I_{i,t}) = 0$. In order to be able to use the OA algorithm with equality relaxation, it is necessary that $\tau_{i,t}^l \cdot H(I_{i,t-1}, \bar{q}_{it}, p_{it}, I_{i,t})$ is a quasiconvex function (Floudas, 1995). In the Appendix A.1, we mathematically show that this requirement holds. Then, based on Constraints (28), if the multiplier $\lambda_{i,t}^l$ is positive, the Constraint (21) can be reformulated as $I_{i,t-1} + \sum_{k \in K} q_{ikt} - I_{it} - d_{it}(p_{it}) \geq 0$. If the sign of multiplier $\lambda_{i,t}^l$ is negative, the equality is reformulated as $I_{i,t-1} + \sum_{k \in K} q_{ikt} - I_{it} - d_{it}(p_{it}) \leq 0$. Lastly, if the multiplier $\lambda_{i,t}^l$ equals zero, the nonlinear equality (21) can be eliminated from the constraint set.

4.3. Feasibility problem

In the case of a nonlinear demand function, where demand approaches to zero asymptotically, or when there is an upper bound for price which does not allow the demand to be zero, the primal problem may be infeasible for certain combinations of bin-

ary variables. If the primal problem is infeasible at iteration l , the following feasibility problem (an NLP) needs to be solved:

$$(P_F^l) \min Z_{P_F}^l = \sum_{t \in T} \left(\alpha_t + \sum_{k \in K} \left(\alpha'_{kt} + \sum_{i \in N_C} \alpha''_{ikt} \right) \right) \quad (29)$$

s.t.

$$I_{0,t-1} + W_t = \sum_{i \in N_C} \sum_{k \in K} q_{ikt} + I_{0t} \quad \forall t \in T \quad (30)$$

$$I_{i,t-1} + \sum_{k \in K} q_{ikt} = d_{it}(p_{it}) + I_{it} \quad \forall i \in N_C, \forall t \in T \quad (31)$$

$$W_t \leq C \bar{y}_t^l + \alpha_t \quad \forall t \in T \quad (32)$$

$$I_{0t} \leq L_0 \quad \forall t \in T \quad (33)$$

$$I_{i,t-1} + \sum_{k \in K} q_{ikt} \leq L_i \quad \forall i \in N_C, \forall t \in T \quad (34)$$

$$\sum_{i \in N_C} q_{ikt} \leq Q \bar{z}_{0kt}^l + \alpha'_{kt} \quad \forall k \in K, \forall t \in T \quad (35)$$

$$q_{ikt} \leq \min \{L_i, Q\} \bar{z}_{ikt}^l + \alpha''_{ikt} \quad \forall i \in N_C, \forall k \in K, \forall t \in T \quad (36)$$

$$W_t, I_{it}, q_{ikt}, p_{it}, \alpha_t, \alpha'_{kt}, \alpha''_{ikt} \geq 0 \quad \forall i \in N, \forall t \in T, \forall k \in K \quad (37)$$

The variables α_t , α'_{kt} and α''_{ikt} determine the constraint violations. The solution provided by the feasibility problem is used to linearize the solution space (note that this problem always finds a feasible point). Similar to the primal problem, the Lagrange multipliers $\lambda_{i,t}^l$ (28) associated with the equality Constraints (31) are used to transform the nonlinear equalities into inequalities in the master problem.

4.4. Master problem

The primal problem (P_P^l) is solved for a fixed set of binary variables at each iteration l . If P_P^l is infeasible, then the feasibility problem (P_F^l) is solved. The solutions of both problems are used for an outer approximation of the solution space, and the feasible solutions to P_P^l are utilized to approximate the objective function. The cuts related to the outer approximation of objective function and the solution space are progressively added to the master problem (P_M^l) at each iteration (see Constraints (39), (41) and (42)). According to Duran and Grossmann (1986), the optimal solution to the main problem (P) and the master problem is the same if all linear supports for nonlinear functions are included in the master problem. The master problem is a mixed integer linear program (MILP) and is formulated as follows in each iteration l :

$$(P_M^l) \max Z_{P_M}^l = \pi - \sum_{t \in T} \left(u_t W_t + f_t y_t + \sum_{i \in N} h_{it} I_{it} + \sum_{(i,j) \in A} \sum_{k \in K} c_{ijt} x_{ijkt} \right) \quad (38)$$

s.t.

$$\sum_{t \in T} \sum_{i \in N_C} (r_{it}(p_{it}^v) + \nabla r_{it}(p_{it}^v)(p_{it}^v - p_{it})) \geq \pi \quad \forall v \in F^l \quad (39)$$

$$\pi \leq \pi^{\max} \quad (40)$$

$$\tau_{i,t}^v \left(I_{i,t-1} + \sum_{k \in K} q_{ikt} - I_{it} - d_{it}(p_{it}^v) - \nabla d_{it}(p_{it}^v)(p_{it} - p_{it}^v) \right) \leq 0 \quad \forall v \in F^l, \forall i \in N_C, \forall t \in T \quad (41)$$

$$\bar{\tau}_{i,t}^v \left(I_{i,t-1} + \sum_{k \in K} q_{ikt} - I_{it} - d_{it}(p_{it}^{\bar{v}}) - \nabla d_{it}(p_{it}^{\bar{v}})(p_{it} - p_{it}^{\bar{v}}) \right) \leq 0 \quad \forall \bar{v} \in \bar{F}^l, \forall i \in N_C, \forall t \in T \quad (42)$$

$$I_{0,t-1} + W_t = \sum_{i \in N_C} \sum_{k \in K} q_{ikt} + I_{0t} \quad \forall t \in T \quad (43)$$

$$W_t \leq C y_t \quad \forall t \in T \quad (44)$$

$$I_{0t} \leq L_0 \quad \forall t \in T \quad (45)$$

$$I_{i,t-1} + \sum_{k \in K} q_{ikt} \leq L_i \quad \forall i \in N_C, \forall t \in T \quad (46)$$

$$\sum_{i \in N_C} q_{ikt} \leq Q z_{0kt} \quad \forall k \in K, \forall t \in T \quad (47)$$

$$\sum_{k \in K} z_{ikt} \leq 1 \quad \forall i \in N_C, \forall t \in T \quad (48)$$

$$q_{ikt} \leq \min\{L_i, Q\} z_{ikt} \quad \forall i \in N_C, \forall k \in K, \forall t \in T \quad (49)$$

$$\sum_{j \in N} x_{ijkt} + \sum_{j \in N} x_{jik} = 2 z_{ikt} \quad \forall i \in N, \forall t \in T, \forall k \in K \quad (50)$$

$$\sum_{i \in S} \sum_{j \in S} x_{ijkt} \leq \sum_{i \in S} z_{ikt} - z_{ekt} \quad \forall S \subseteq N_C |S| \geq 2, \quad (51)$$

$$\forall e \in S, \forall t \in T, \forall k \in K \quad (51)$$

$$W_t, I_{it}, q_{ikt}, p_{it}, \pi \geq 0 \quad \forall i \in N, \forall t \in T, \forall k \in K \quad (52)$$

$$y_t, z_{ikt}, x_{ijkt} \in \{0, 1\} \quad \forall i, j \in N, \forall k \in K, \forall t \in T \quad (53)$$

where $r_{it}(p_{it}) = p_{it} d_{it}(p_{it})$ denotes the revenue function, F^l denotes the set of iterations before l in which the primal problem is feasible, and $\bar{F}^l$ denotes the set of previous iterations where a feasibility problem is solved due to infeasibility of the primal problem. p_{it}^v is the optimal price of the primal problem at iteration v , and $p_{it}^{\bar{v}}$ is the optimal price of the feasibility problem at iteration $\bar{v}$. The variable π in Constraints (39) is considered as an upper bound for the total revenue $r_{it}(p_{it})$. The left hand-side of Constraints (39) denote the valid linear supports for the nonlinear revenue function $r_{it}(p_{it})$ at $p_{it} = p_{it}^v, \forall v \in F^l$. The variable π is bounded by an upper bound $\pi^{\max}$ in Constraints (40). Constraints (41) and (42) are the linear approximations of the demand function and are replaced with Constraints (3) for a linear demand function.

Integer cuts

A specific type of integer cuts ensures that the sets of integer combinations previously considered in the primal problem are not considered again in the subsequent iterations. For the problem whose integer variables are binary variables, such as our problem, the following integer cut is used. This cut performs similar to the well-known integer cut proposed by Duran and Grossmann (1986), but its application is simpler.

$$\begin{aligned} & \sum_{i \in N} \sum_{j \in N} \sum_{k \in K} \sum_{t \in T} (-1)^{x_{ijkt}^v + 1} x_{ijkt} + \sum_{i \in N} \sum_{k \in K} \sum_{t \in T} (-1)^{z_{ikt}^{\bar{v}} + 1} z_{ikt} \\ & + \sum_{t \in T} (-1)^{y_t^{\bar{v}} + 1} y_t \\ & \leq \sum_{i \in N} \sum_{j \in N} \sum_{k \in K} \sum_{t \in T} x_{ijkt}^v + \sum_{i \in N} \sum_{k \in K} \sum_{t \in T} z_{ikt}^v + \sum_{t \in T} y_t^v - 1 \quad \forall i, j \\ & \in N, \forall k \in K, \forall t \in T, \forall v \in \{0, \dots, l-1\} \end{aligned} \quad (54)$$

Constraint (54) is violated by the combinations of binary variables obtained in previous iterations. Hence, this constraint helps expediting the solution procedure of the master problem by limiting the search space of the binary variables. In Appendix B, we explain in more detail why this constraint holds.

4.5. The two-phase OA algorithm

Recall that the objective value of the optimal solution to the master problem provides a valid upper bound on the optimal objective value of the main problem. However, solving the master problem to optimality in each iteration is computationally expensive. A possible idea to tackle this challenge may include solving the master problem with an optimality gap of $\gamma > 0$ in each iteration, and using the objective value of the solution to this problem as an upper bound on the optimal solution of the main problem. Although this approach can be easily implemented in commercial solvers via tweaking solver settings, it does not guarantee that the algorithm converges to an optimal solution at the end. This is because if the master problem is not solved to optimality, then the objective value of the obtained solution is not necessarily a valid upper bound on the optimal solution of the main problem (e.g., the obtained objective value may well be lower than the optimal objective value of the main problem). Hence, while the γ -optimal solution itself can be used as an input for the primal problem, its objective value cannot be used as a valid upper bound on the optimal solution of the main problem P within the OA framework. However, note that if the MIP solver for master problem is interrupted at an optimality gap of $\gamma > 0$, while the objective value of the incumbent (γ -optimal) solution is not necessarily an upper bound to the main problem, the objective value corresponding to the fractional solution, which we denote by Z_{object}^l , is a valid upper bound to the main problem. Therefore, by solving the master problem with an optimality gap of $\gamma > 0$, we can obtain faster valid upper bounds, i.e., Z_{object}^l on the optimal objective value of the main problem (regardless of the objective values of actual γ -optimal solutions that are obtained). In our two-phase algorithm, we exploit this property. Below, we provide a description of our algorithm and present the pseudocode (in our description, we make references to the respective lines in the pseudocode, see Algorithm 2).

Specifically, in our two-phase OA algorithm, we implement the following (after a similar initialization procedure as in the basic OA algorithm, see lines 1–9 in the pseudocode of Algorithm 2). In the first phase of the algorithm (line 10), instead of solving the master problem to optimality, we set the optimality gap of the master problem equal to $\gamma > 0$ (lines 11 and 12) in each iteration l , and use the upper bound Z_{object}^l provided by the solver as a valid upper bound on the objective value of the main problem P (line 13). Then, we progressively reduce γ in each iteration (line 11) such that after $|N| * |T| * |K|$ iterations, the algorithm enters Phase 2 (line 14) and the master problem is solved to optimality ($\gamma = 0$) in each iteration (lines 15 and 16). Such a setup reduces the computation time by reducing the iteration durations earlier in the algorithm. In Phase 2, since the master problem is solved to optimality, the objective value of the master problem can be used as a valid upper bound for the optimal objective value of the main problem P (line 17). This upper bound is tighter than the bounds provided in Phase 1. The current values of the binary variables (line 18) are used to solve the problem P_p^l (line 19). If the primal problem is feasible (line 20), its objective value is compared with the current lower bound, and the lower bound is updated (line 21); otherwise (line 22), the feasibility problem (P_f^l) is solved (line 23). The algorithm continues until the upper and lower bounds converge within ε (line 7).

The pseudocode of two-phase algorithm is presented below.

```

1 Initialize  $l \leftarrow 0$ ,  $Z_U^0 \leftarrow \pi^{\max}$ ,  $Z_L^0 \leftarrow -\infty$ , set  $\varepsilon$  and  $\gamma$ ,  $\bar{x}_{ijkt}^0$ ,  $\bar{z}_{ikst}^0$ ,  $\bar{y}_t^0 = 0$ 
2 solve  $P_p^0$  and obtain optimal price  $p_{it}^0$ 
3 if  $P_p^0$  is feasible then
4    $Z_L^0 \leftarrow Z_{P_p}^0$ 
5 else
6   solve  $P_F^0$  and obtain optimal price  $p_{it}^0$ 
7 while  $(Z_U^l - Z_L^l)/Z_U^l > \varepsilon$  do
8    $l \leftarrow l + 1$ 
9   use  $p_{it}^{l-1}$  to add linear approximation cuts to  $P_M^l$ 
10  if  $l < |N| * |T| * |K|$  then
11     $\gamma \leftarrow \gamma * (1 - \frac{1}{|N|*|T|*|K|})$ 
12    solve  $P_M^l$  with optimality gap  $\gamma$  and obtain binary variables  $x_{ijkt}^l$ ,  $z_{ikst}^l$ ,  $y_t^l$ 
13     $Z_U^l \leftarrow \min(Z_U^{l-1}, Z_{objest}^l)$ 
14  else
15     $\gamma \leftarrow 0$ 
16    solve  $P_M^l$  optimally and obtain optimal binary variables  $x_{ijkt}^l$ ,  $z_{ikst}^l$ ,  $y_t^l$ 
17     $Z_U^l \leftarrow \min(Z_U^{l-1}, Z_{P_M}^l)$ 
18     $\bar{x}_{ijkt}^l = x_{ijkt}^l$ ,  $\bar{z}_{ikst}^l = z_{ikst}^l$ ,  $\bar{y}_t^l = y_t^l$ 
19    solve  $P_p^l$  and obtain optimal solution  $p_{it}^l$ 
20    if  $P_p^l$  is feasible then
21       $Z_L^l \leftarrow \max(Z_L^{l-1}, Z_{P_p}^l)$ 
22    else
23      solve  $P_F^l$  and obtain optimal solution  $p_{it}^l$ 
24 Output  $(W_t, I_{it}, y_t, z_{ikst}, x_{ijkt}, q_{ikst}, p_{it})$ ,  $Z_P$ ,  $Z_U$ 

```

5. Computational results

Numerical experiments are performed to evaluate the performance of the proposed OA algorithms. GAMS IDE (ver. 24.7.4) software is used for the presented MINLP model and the OA algorithms. The problem parameters are generated according to the instances available in the PRP literature. To create our own instances, the data set of Archetti et al. (2011) and Adulyasak et al. (2014) are utilized. The generated parameters are listed in Table 1. A computer with a Core i5 2.53 GHz CPU and 4 GB RAM is used.

Random parameter selections are performed according to a (discrete) uniform distribution. Note that A_{it} defines the maximum possible amount of demand for retailer $i \in N_C$ in period $t \in T$. For example, for linear demand function $d_{it}(p_{it}) = a_{it} - b_{it}p_{it}$, it holds that $A_{it} = a_{it}$. A sensitivity analysis on the parameter assumed as

Table 1
The problem parameters.

L_i	Is equal to $\bar{d}_i g_i$, where g_i is randomly selected from the set $\{2, 3, 6\}$ and $\bar{d}_i = \sum_t A_{it} / (100 T)$
L_0	Is equal to $\rho \sum_i L_i$, where ρ is randomly selected from the set $\{2, 3, 6\}$
h_{it}	Is randomly generated as an integer number in the interval $[1, 10]$
h_{0t}	Is randomly selected from the set $\{3, 8\}$
u_t	Is equal to $10h_{0t}$
f_t	Is equal to $100u_t$
c_{ijt}	Is equal to $\varphi_t \left\lceil \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2} + 0.5 \right\rceil$, where the points (X_i, Y_i) and (X_j, Y_j) are randomly generated as an integer number in the interval $[0, 1000]$. As such, $\left\lceil \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2} + 0.5 \right\rceil$ indicates the distance between two points, and φ_t is a random number in the interval $[0.5, 1]$, which states the travel cost per unit distance (i.e., the travel cost is considered as a linear function of distance.)
Q	Is randomly selected from the set $\{\bar{L}, 3/2\bar{L}, 2\bar{L}\}$, where $\bar{L} = \max_{i \in N_C} \{L_i + \bar{d}_i\}$
C	Is equal to $2Q$

Table 2
Numerical results for the linear demand function.

Size $ N * T * K $	BONMIN (B-BB)			BONMIN (B-OA)			DICOPT			Basic OA			Two-phase OA			$\epsilon = 10\%$			$\epsilon = 5\%$			$\epsilon = 3\%$				
	time		obj	time		obj	time		obj	time		obj	time		obj	time		obj	time		obj	time		obj		
	time	obj	time	obj	time	obj	time	obj	time	obj	time	obj	time	obj	time	obj	time	obj	time	obj	time	obj	time	obj		
1	5'3"2	123.5	224,188	23.6	224,188	13.7	224,188	11.9	224,188	11.5	217,859	2.82%	4.22%	12.0	221,636	1.14%	1.92%	12.1	221,636	1.14%	1.86%	12.1	221,636	1.14%	1.86%	
2	5'3"3	207.2	224,928	26.2	224,928	15.8	224,928	12.1	224,928	12.0	217,650	3.24%	4.34%	12.9	221,945	1.33%	2.30%	13.5	222,382	1.13%	1.70%	13.5	222,382	1.13%	1.70%	
3	5'6"2	161.9	280,476	2313.5	280,476 (4)	14.2	280,476	12.8	280,476	12.5	272,494	2.85%	4.75%	13.9	274,641	2.08%	3.28%	18.1	276,396	1.45%	2.27%	18.1	276,396	1.45%	2.27%	
4	5'6"3	378.9	324,459	2903.7	324,459 (3)	17.5	324,459	13.0	324,459	12.6	314,306	3.13%	4.77%	14.0	317,168	2.25%	3.08%	18.4	319,491	1.53%	2.36%	18.4	319,491	1.53%	2.36%	
5	5'9"2	4082.4	603,093	4294.6	603,093 (3)	17.9	603,093	16.0	603,093	14.5	586,827	2.70%	3.98%	14.6	586,827	2.70%	3.98%	22.7	591,369	1.94%	2.84%	22.7	591,369	1.94%	2.84%	
6	5'9"3	4508.7	604,427	4408.7	604,427 (3)	18.0	604,427	16.2	604,427	14.8	589,303	2.50%	3.14%	14.7	589,303	2.50%	3.14%	26.8	592,132	2.03%	2.68%	26.8	592,132	2.03%	2.68%	
7	7'3"2	1399.3	285,840	2921.8	285,840	278,193	2.68%	3.16%	12.0	279,781	2.12%	2.57%	13.3	280,268	1.95%	2.37%										
(3)	20.3	285,840	13.1	285,840	11.7	285,897	13.7	285,897	13.7	285,897	13.2	181,569	1.51%	1.84%	17.5	412,801	1.75%	2.66%	20.0	415,974	1.00%	1.80%	20.0	415,974	1.00%	1.80%
8	7'3"3	2660.0	285,897	3066.8	285,897 (3)	25.0	7'3"3	285,897	52.4	420,167	13.0	399,735	4.86%	5.97%	17.5	412,801	1.75%	2.66%	15.3	283,379	0.88%	1.12%	15.3	283,379	0.88%	1.12%
9	7'6"2	5778.8	419,419	7200	-	61.6	420,167	52.4	420,167	13.0	399,735	4.86%	5.97%	17.5	412,801	1.75%	2.66%	20.0	415,974	1.00%	1.80%	20.0	415,974	1.00%	1.80%	
10	7'6"3	7200	420,153	7200	-	104.9	420,715	70.5	420,715	13.3	394,477	6.24%	7.38%	19.0	409,735	2.61%	3.35%	20.8	413,107	1.81%	2.47%	20.8	413,107	1.81%	2.47%	
1	7'9"2	7200	586,501	7200	-	141.2	655,256	91.4	655,256	14.7	632,741	3.44%	5.04%	18.6	638,194	2.60%	3.98%	62.6	644,988	1.57%	2.40%	62.6	644,988	1.57%	2.40%	
12	7'9"3	7200	535,023	7200	-	268.5	655,911	193.7	655,911	15.3	626,178	4.53%	5.87%	20.5	642,864	1.99%	3.26%	111.1	646,988	1.36%	2.14%	111.1	646,988	1.36%	2.14%	
13	10'3"2	7200	332,164 (4)	7200	-	289.1	423,445	128.2	423,445	19.9	409,296	3.34%	3.97%	22.0	413,495	2.35%	2.98%	29.6	416,339	1.68%	2.21%	29.6	416,339	1.68%	2.21%	
14	10'3"3	7200	-	7200	-	374.6	411,146	179.6	411,146	21.9	399,432	2.85%	3.69%	25.0	406,138	1.22%	1.99%	32.3	406,443	1.14%	1.81%	32.3	406,443	1.14%	1.81%	
15	10'6"2	7200	-	7200	-	4163.4	755,645	3629.9	755,645	31.2	729,975	3.40%	4.46%	36.7	743,175	1.65%	2.46%	42.9	748,188	0.99%	1.58%	42.9	748,188	0.99%	1.58%	
16	10'6"3	7200	-	7200	-	7200	692,588 (2)	4145.7	692,591	41.0	684,219	1.21%	2.02%	42.2	684,219	1.21%	2.02%	52.0	685,157	1.07%	1.55%	52.0	685,157	1.07%	1.55%	
17	10'9"2	7200	-	7200	-	5581.1	723,133 (3)	4519.4	723,539	51.7	706,756	2.32%	3.20%	53.9	706,756	2.32%	3.20%	62.8	713,322	1.41%	2.07%	62.8	713,322	1.41%	2.07%	
18	10'9"3	7200	-	7200	-	7200	713,805 (2)	4854.6	714,413	70.0	704,668	1.36%	2.41%	72.4	704,668	1.36%	2.41%	98.1	704,668	1.36%	2.05%	98.1	704,668	1.36%	2.05%	
19	12'3"2	7200	-	7200	-	7200	-	2015.6	435,444	112.6	429,799	1.30%	2.05%	114.0	429,799	1.30%	2.05%	126.0	430,209	1.20%	1.95%	126.0	430,209	1.20%	1.95%	
20	12'3"3	7200	-	7200	-	7200	-	3064.7	435,645	162.8	418,377	3.96%	4.72%	182.9	429,820	1.34%	2.08%	183.6	429,820	1.34%	2.08%	183.6	429,820	1.34%	2.08%	
21	12'6"2	7200	-	7200	-	7200	-	7200	964,010	328.1	946,564	1.81%	2.47%	327.5	946,564	1.81%	2.47%	350.8	955,847	0.85%	1.40%	350.8	955,847	0.85%	1.40%	
22	12'6"3	7200	-	7200	-	7200	-	7200	966,760	559.5	961,424	0.55%	0.99%	560.4	961,424	0.55%	0.99%	559.8	961,424	0.55%	0.99%	559.8	961,424	0.55%	0.99%	
23	12'9"2	7200	-	7200	-	7200	-	7200	985,790	957.4	974,257	1.17%	2.04%	961.8	974,257	1.17%	2.04%	1178.5	981,254	0.46%	1.24%	1178.5	981,254	0.46%	1.24%	
24	12'9"3	7200	-	7200	-	7200	-	7200	986,738	1627.4	973,616	1.33%	2.19%	1632.6	973,616	1.33%	2.19%	1833.6	979,570	0.73%	1.49%	1833.6	979,570	0.73%	1.49%	

100 in calculating $\bar{d}_i = \sum_t A_{it} / (100|T|)$ is provided in the Appendix C.1.

To compare the objective value and the solution time of our proposed OA algorithms and MINLP solvers, 24 problem instances with various sizes are considered. We set the number of retailers as $|N| = 5, 7, 10$ and 12 , the number of periods as $|T| = 3, 6$ and 9 and the number of vehicles as $|K| = 2$ and 3 which forms $4 \times 3 \times 2 = 24$ problems in total. Problem instances with the same number of retailers and periods but with different number of vehicles are generated with the same values for other parameters. To obtain reliable results, for each problem size, five different instances are generated using the parameters in Table 1 and the average results are reported. To evaluate the efficiency of the proposed OA algorithms, three MINLP algorithms are considered: 1) the default B-BB algorithm of the BONMIN solver, 2) B-OA algorithm of the BONMIN solver, and 3) DICOPT solver. B-BB algorithm is selected since it is the default algorithm of the BONMIN solver, and it performs based on a traditional branch and bound algorithm. B-OA and DICOPT are selected since they perform based on an OA algorithm similar to our proposed algorithms.

For the DICOPT solver, there are several options that are tuned to have a fair comparison. First of all, we used the option “Stop on crossover” which provides the global optimum in convex problems as the stopping criterion. We also changed the default value of “maxcycles” to 10,000 to prevent the termination of the algorithm before finding the optimal solution. We also tested the settings “convex = 1” and “feaspump = 1,” and since the results became worse, we disabled these settings. Note that by using the “optcr” option, different optimality gaps, i.e., γ , can be set for the master problem in the DICOPT solver. This means that in each iteration, the objective function value of a γ -optimal solution is used as an upper bound on the main problem P within the OA framework. However, as we discuss in Section 4, the objective function value of a γ -optimal solution cannot be considered as valid upper bound on the main problem P within the OA framework, as the algorithm may stop at a solution with unknown quality. That is why our two-phase algorithm is distinctive and cannot be realized in DICOPT solver.

5.1. Results for the linear demand function

The performance comparison between the GAMS MINLP solvers and the proposed OA algorithms in terms of objective value (“obj” column) and computational time (“time” column) for a linear demand function, formulated as $d_{it}(p_{it}) = a_{it} - b_{it}p_{it}$, is presented in Table 2. The parameter of the market size is defined as $a_{it} = \bar{a}_{it}\eta_{it}$, where $\bar{a}_{it}$ is randomly generated as an integer number in the interval $[50, 500]$ and η_{it} is randomly selected from the set $\{3, 8\}$. The price sensitivity parameter b_{it} is an integer number which is randomly generated from the interval $[3, 8]$. The initial γ for the two-phase algorithm is set to 0.1. The MIP and NLP solvers which are used in the GAMS MINLP solvers and the proposed OA algorithms are CPLEX and IPOPT, respectively.

Note that the numbers in parentheses in the “obj” column demonstrate the number of instances with a feasible solution (although not necessarily the optimal solution) within 7200 s for the associated problem size. For those problem sizes with some instances where no feasible solution has been obtained within 7200 s using one or more solvers, the average solution time and objective value are calculated by considering the solution time and objective value across the instances with feasible solutions only. For the instances which are not solved to optimality, the objective value that is obtained in 7200 s is used to calculate the average objective value. The “Gap-opt” column for the two-phase OA algorithm shows the deviation between the solution provided

by the two-phase OA algorithm and the solution provided by the basic OA algorithm. The “Gap-up” column for the two-phase OA algorithm shows the deviation between the objective value of the solution provided by the two-phase OA algorithm and the upper bound provided in the last iteration of the two-phase OA algorithm. “Gap-up” is calculated as (upper bound-obj)/upper bound. Therefore, this column is indicative of the performance of the two-phase OA algorithm in terms of the quality of the near-optimal solutions provided by it.

Table 2 shows that the B-OA algorithm of BONMIN solver is not able to provide feasible solutions for the problem instances with sizes more than $7 \times 3 \times 3$ in 7200 s. In addition, the computational time of B-OA algorithm of BONMIN solver is significantly higher than the other compared GAMS MINLP algorithms and our proposed OA algorithms. An exception to this applies to very small problem sizes for which B-OA algorithm performs better than the B-BB algorithm in terms of computation time. The efficiency of the default algorithm of BONMIN solver (B-BB) is better compared to the B-OA algorithm in general, the B-BB algorithm cannot provide feasible solutions for problem instances that are larger than $10 \times 3 \times 2$. Our results show that the DICOPT solver performs best among the three algorithms that are compared. However, DICOPT is also not able to provide feasible solutions for the problem instances that are larger than the size of $10 \times 9 \times 3$. In addition, the quality of obtained solution for large instances is not acceptable using this solver – DICOPT starts performing bad for problems that are larger than $10 \times 6 \times 2$ in terms of providing the optimal solution. The basic OA algorithm is capable to solve all the problem instances. Table 2 shows that with the proposed basic OA algorithm, we can solve problem instances up to $12 \times 3 \times 3$ optimally. In addition, we can provide high quality solutions for large problems that none of the compared algorithms can handle (e.g. problem with sizes larger than $12 \times 3 \times 3$). Table 2 shows that the reduction in the average computation time per problem size can exceed 50%. For larger size problems, this is a big improvement in computation time.

The basic OA algorithm has a much better performance than the commercial MINLP solvers implemented in GAMS, but it still suffers from the long computational time needed to solve the master problem to optimality, especially for large problem instances. Our proposed two-phase OA algorithm overcomes this challenge and provides near-optimal solutions very fast. By avoiding solving the master problem optimally in each iteration, the two-phase OA algorithm can provide the solution in any desired gap with the upper bound in a very efficient manner.

The results for the performance of the two-phase OA algorithm with tolerance (between the lower and upper bounds on the optimal solution) $\varepsilon = 10\%$, 5% and 3% are reported in Table 2. The two-phase OA algorithm is also capable of providing the optimal solution by setting $\varepsilon = 0\%$. However, our experiments show that the basic OA algorithm provides the optimal solution faster than the two-phase OA algorithm with $\varepsilon = 0\%$ for small size problem instances which can be solved optimally in less than 7200 s. Since the upper bound in the two-phase OA algorithm is looser than the basic OA algorithm in the first phase, reaching the zero gap requires many iterations in two-phase OA algorithm. On the other hand, the basic OA algorithm can provide a feasible solution with an objective value that lies within a specific gap ε by adjusting the stopping criterion of the algorithm. However, since the master problem has to be solved optimally in each iteration, it cannot provide these solutions in a reasonable time for large size problem instances (We refer the reader to the Appendix C.2 for a comparison between the performance of two proposed OA algorithms with $\varepsilon > 0\%$). Therefore, based on our observations, it is suggested to apply the basic OA algorithm for solving the problem optimally (i.e., $\varepsilon = 0\%$) for small problem instances, but to use the two-phase

Table 3
Numerical results for the nonlinear demand function.

Size $ N * T * K $		DICOPT		Basic OA		Two-phase OA							
						$\varepsilon = 10\%$				$\varepsilon = 5\%$			
		time	obj	time	obj	time	obj	Gap-opt	Gap-up	time	obj	Gap-opt	Gap-up
1	5*3*2	19.7	284,578	17.0	284,578	12.8	283,912	0.23%	4.82%	13.6	284,481	0.03%	4.16%
2	5*3*3	20.5	284,578	17.4	284,578	13.0	283,853	0.25%	4.77%	14.8	284,422	0.05%	4.14%
3	5*6*2	29.5	582,042	18.4	582,042	13.3	577,001	0.87%	6.62%	16.4	582,034	0.00%	4.86%
4	5*6*3	35.4	582,042	21.0	582,042	13.5	574,227	1.34%	7.17%	19.1	582,034	0.00%	4.85%
5	5*9*2	41.3	724,526	38.1	724,526	14.2	712,587	1.65%	6.57%	23.6	722,959	0.22%	4.51%
6	5*9*3	47.2	724,526	45.6	724,526	14.4	720,860	0.51%	5.30%	26.8	722,036	0.34%	4.75%
7	7*3*2	35.1	417,694	20.1	417,694	13.1	414,218	0.83%	6.14%	15.4	417,367	0.08%	4.76%
8	7*3*3	53.1	417,694	20.8	417,694	13.4	416,102	0.38%	5.49%	16.0	417,415	0.07%	4.76%
9	7*6*2	985.0	872,881	84.5	872,881	14.6	864,299	0.98%	6.44%	28.2	871,918	0.11%	4.66%
10	7*6*3	1189.7	883,619	123.9	883,619	15.0	861,881	2.46%	6.18%	32.7	871,743	1.34%	4.78%
11	7*9*2	1898.8	1,184,332	314.0	1,184,332	15.4	1,160,840	1.98%	6.87%	35.3	1,178,331	0.51%	4.80%
12	7*9*3	1999.1	1,184,332	457.5	1,184,332	17.4	1,175,492	0.75%	5.38%	37.1	1,178,428	0.50%	4.73%
13	10*3*2	2158.8	563,939	486.9	563,939	20.8	556,232	1.37%	6.76%	63.4	563,234	0.13%	4.92%
14	10*3*3	2864.3	563,939	690.2	563,939	25.2	559,733	0.75%	6.00%	140.2	563,369	0.10%	4.87%
15	10*6*2	7200	1,129,194 (3)	5448.3	1,130,022	57.1	1,118,152	1.05%	4.39%	82.8	1,125,623	0.39%	3.63%
16	10*6*3	7200	1,154,093 (1)	5943.3	1,156,974	79.1	1,115,927	3.55%	7.82%	157.9	1,153,887	0.27%	3.57%
17	10*9*2	7200	1,638,904 (1)	6727.5	1,639,264	146.9	1,615,533	1.45%	6.46%	153.0	1,635,697	0.22%	4.07%
18	10*9*3	7200	–	7200	1,945,272	152.6	1,941,433	0.20%	4.88%	182.6	1,943,606	0.09%	4.60%
19	12*3*2	7200	–	7200	697,194	129.4	693,049	0.59%	3.25%	143.7	696,215	0.14%	2.34%
20	12*3*3	7200	–	7200	697,194	204.0	692,482	0.68%	3.35%	227.9	695,512	0.24%	2.53%
21	12*6*2	7200	–	7200	1,452,402	366.5	1,441,736	0.73%	4.65%	439.6	1,448,399	0.28%	4.12%
22	12*6*3	7200	–	7200	1,452,402	641.7	1,445,625	0.47%	4.51%	721.0	1,450,236	0.15%	3.97%
23	12*9*2	7200	–	7200	2,392,793	1461.8	2,388,788	0.17%	3.77%	1690.9	2,389,780	0.13%	3.66%
24	12*9*3	7200	–	7200	2,390,378	1770.0	2,384,829	0.23%	3.69%	1848.0	2,386,135	0.18%	3.54%

OA algorithm to provide fast, good quality solutions (i.e., $\varepsilon > 0\%$) for medium and large problem instances. Table 2 shows that by setting $\varepsilon = 3\%$ and using the two-phase OA, the computational time to solve the large problem instances can be reduced by more than 90% compared to the basic OA algorithm. Furthermore, the objective value provided by the two-phase OA algorithm deviates only around 1% from the solution provided by the basic OA algorithm.

5.2. Results for the nonlinear demand function

The PRP-PD for the nonlinear demand function is considered in this section. It is assumed that the demand follows a power function which is modeled as $d_{it}(p_{it}) = (\delta_{it}p_{it} + \beta_{it})^{-\xi}$, where $\delta_{it}, \beta_{it} > 0$ and $\xi > 1$ (The graph of the power demand function is presented in the Appendix C.3). To keep the revenue function concave, the

price is bounded from above by the equation $p_{it} \leq 2\beta_{it}/(\delta_{it}(\xi - 1))$. To conduct the experiments, we assume that $\xi = 2$, $\beta_{it} = 1/\sqrt[3]{a_{it}}$, where $a_{it} = \bar{a}_{it}\eta_{it}$ is the market size such that $\bar{a}_{it}$ is randomly generated as an integer number in the interval $[50, 500]$ and η_{it} is randomly selected from the set $\{3, 8\}$. We define the multiplier of price as $\delta_{it} = (1/100\sqrt[3]{a'_{it}})$, where a'_{it} is calculated in the same way as a_{it} . Due to considering the upper bound for price, the demand would not be zero. Therefore, to make the main problem feasible for this demand function, we increase the production and vehicles capacities as well as the maximum inventory that can be held at the production plant and the retailers by a factor of 10, in comparison to those for the linear demand function. The other parameters are the same as Table 1. To demonstrate the efficiency of the proposed algorithms, the objective value and solution

Table 4

Parameters to analyze the price sensitivity.

	a_i	b_i	L_i	h_i	c_{0i}	L_0	h_0	u	f	Q
Retailer I	1000	6	60	5	500	360	8	80	8000	105
Retailer II	1000	3	60	5	500

(a) Optimal price for each retailer

(b) Optimal demand for each retailer

(c) Optimal profit for the manufacturer

Fig. 1. Optimal solution as a function of manufacturer's capacity for the case with two retailers who have different price sensitivities.

Table 5

Parameters to analyze the distance of retailers and the production plant.

	a_i	b_i	L_i	h_i	c_{0i}	L_0	h_0	u	f	Q
Retailer I	1000	6	60	5	1500	360	8	80	8000	105
Retailer II	1000	6	60	5	500

Fig. 2. Optimal solution as a function of manufacturer's capacity for the case with two retailers who have different distances to the manufacturing plant.

Table 6
Parameters to analyze the market size and the price sensitivity.

L_i	h_i	c_{0i}	L_0	h_0	u	f	Q	C
60	5	500	360	8	80	8000	105	210

Fig. 3. Investigating market characteristics.

time of the best GAMS solver in Section 5.1, i.e. DICOPT, and the OA algorithms are compared in Table 3.

As demonstrated in Table 3, the proposed OA algorithms perform much better than the DICOPT solver in the case of nonlinear demand function as well. DICOPT solver is not able to provide feasible solutions for the problem instances that are larger than

$10 \times 9 \times 2$ and its efficiency in finding optimal solution reduces for problem instances larger than $10 \times 3 \times 3$. The basic OA, on the other hand, can solve all problem instances, and it provides the optimal solution for problems up to size $10 \times 9 \times 2$ and high-quality solutions for larger problems. The reduction of computational time using basic OA compared to DICOPT solver can

Table 7
Investigating the value of pricing flexibility.

Size $ N * T * K $	p_{it}			P_i			P_t			p		
	time	obj	Gap	time	obj	Gap	time	obj	Gap	time	obj	Gap
1	30.1	2,021,180	45%	11.6	1,105,743	45%	11.7	955,847	53%	11.9	585,231	71%
2	345.5	3,872,148	56%	11.8	1,688,987	56%	13.3	1,880,453	51%	13.1	649,915	83%
3	244.1	2,200,014	34%	11.9	1,441,167	34%	12.2	355,973	84%	12.2	(86,501)	104%
4	606.2	5,041,031	75%	14.1	1,268,087	75%	13.5	1,527,129	70%	13.2	(89,210)	102%
5	7200	3,251,921	60%	29.5	1,305,710	60%	20.1	733,039	77%	20.6	(264,654)	108%
6	7200	7,651,224	69%	32.0	2,371,122	69%	26.8	1,618,783	79%	27.8	16,548	100%
7	7200	4,337,939	51%	187.0	2,108,812	51%	71.6	375,421	91%	68.5	(697,828)	116%
8	7200	9,124,126	71%	247.6	2,674,011	71%	245.2	1,286,822	86%	223.6	(1,305,370)	114%

exceed 90% for medium size problems. The results of the two-phase OA algorithm are even more encouraging: by setting $\varepsilon = 5\%$, the computational time can be reduced by more than 95% for medium and large problem instances compared to the basic OA algorithm, while the deviation from the obtained solution by basic OA is less than 0.5% for these instances.

Tables 2 and 3 show that “Gap-up” (i.e., the gap between the objective value and the upper bound obtained in the last iteration of the two-phase OA algorithm) is greater for instances with nonlinear demand functions (compared to that for instances with linear demand functions). On the other hand, “Gap-opt” (i.e., the gap between the objective value provided by the two-phase OA algorithm and the objective value provided by the basic OA algorithm) is smaller for instances with nonlinear demand functions (compared to that for instances with linear demand functions). The reasoning for this observation is as follows. The upper bounds provided during the two-phase OA algorithm for the linear demand case are tighter than those provided for the nonlinear demand case. Hence, more effort is required to reach the stopping criteria in the nonlinear demand functions. Therefore, more solutions are tried in primal problem for nonlinear case and the reported solutions are closer to the optimal solution in this case.

It should be mentioned that, the computational time is higher in this case in comparison with instances with a linear demand function. This can be attributed to two major factors: 1) larger capacities, which enlarges the feasible region, in instances with a nonlinear demand function, and 2) additional outer approximations that are needed to be performed in order to linearize the feasible space in instances with a nonlinear demand function. As expected, in Table 2 the objective value for the problem instances with 3 vehicles is higher than the same instances with 2 vehicles. But, in Table 3 the objective value for the instances with 2 and 3 vehicles is almost the same. In fact, since the considered capacities are greater for nonlinear demand function, in both 2 and 3 vehicle cases, only 2 vehicles are used.

5.3. Sensitivity analysis

We consider a linear demand function in the form of $d_{it}(p_{it}) = a_{it} - b_{it}p_{it}$ for our sensitivity analysis and use the basic OA algorithm to solve the instances in this section. To investigate the effect of different characteristics of retailers including the distance and price sensitivity on the pricing and delivery decisions under different production capacities, we consider a problem with six time periods, two vehicles and two different retailers referred to as retailer I and retailer II. In the first analysis, we assume that the retailer I is more sensitive to price, but the other parameters are the same for both retailers. For this analysis, the demand parameters are assumed constant over the planning horizon. Also, for ease of exposition, we assume that a single price is implemented across the time periods. Given the set of parameters in Table 4, the optimal price and the resulting demand for each retailer as well as the optimal profit for the manufacturer are demonstrated in Fig. 1.

According to Fig. 1(a) and (b), the producer is more eager to serve retailer II whose demand is less sensitive, since it can charge retailer II higher prices. Therefore, for the limited capacity, the manufacturer just considers the market related to retailer II. When the capacity increases, the market of retailer I is also included. The fluctuations of demand which are mostly seen in the demand of retailer I are due to the setup cost. With higher capacity, it may be more profitable not to produce in some future periods, but to increase the production amount in the earlier periods. In other words, when the capacity is higher, it is possible to produce more in the earlier periods and eliminate some setups in the future peri-

ods. The objective function in our problem maximizes the profit, i.e., revenue less the cost. Hence, it may be the case that lower demand (i.e., revenue) leads to higher profits if the costs (e.g., set-up costs) are also reduced, which is the case in this setting (i.e., the drops in demand). That is, higher capacity enables the retailer to produce more in a single set-up. Then, the manufacturer chooses to reduce set-ups to save costs. This leads to lower production in total, and lower demand. However, this is still beneficial for the manufacturer as it leads to a higher profit. As shown in Fig. 1, the demand of retailers is capped at 60, due to the retailer's inventory capacity. According to profit plot (Fig. 1(c)), by increasing the capacity the profit will rise. But after a while, the profit tends to be constant and increasing the capacity is a waste of effort.

Another important retailer characteristic pertains to the distance to the production plant. Retailer I is assumed to be farther away from the production plant, but the other characteristics of the retailers including the demand function are assumed the same according to Table 5. The results are depicted in Fig. 2.

According to Fig. 2(a) and (b), the manufacturer sets higher prices for the retailer I whose travel cost is greater and thus demand of this retailer is lower. In other words, retailer II has higher priority to be served by manufacturer because of lower travel costs. Therefore, as illustrated in Fig. 2(b), in the case of limited capacity the manufacturer only considers the market of retailer II. As the production capacity increases, the market of retailer I is also covered, although with higher prices. Similar to the first analysis, as depicted in Fig. 2(c), the profit increases by providing more production capacity. However, increasing the production capacity more than a specific amount leads to no more profit because retailers' inventory capacity may limit the demand that can be supported (Retailer II in Fig. 2). Moreover, because of the high travel cost in some cases it may not be beneficial to produce and deliver more than a specific amount of demand to a certain retailer (Retailer I in Fig. 2).

As mentioned before, a linear demand model, in the form of $d_{it}(p_{it}) = a_{it} - b_{it}p_{it}$, is considered for this experimental analysis. In order to investigate the impact of the market size (a_{it}) and the price sensitivity (b_{it}) on the profit, we performed an additional study (see the setting of other parameters in Table 6). To investigate the impact of the market size, we assume $b_{it} = 6$, and to study the impact of the price sensitivity we assume $a_{it} = 1000$. The results are illustrated in the Fig. 3.

According to Fig. 3, larger market size or lower price sensitivity leads to more profit as expected. However, the profit is increasing almost linearly as the market size becomes larger, but the profit grows exponentially by decreasing the price sensitivity. Since, the price sensitivity has stronger impact on the profit, it is beneficial for the manufacturer to put more efforts on a customer segment, which is less price-sensitive.

In order to obtain additional insights on the value of pricing flexibility (both across retailers and across time) and its impact on computational time, we performed additional experiments by considering three restricted versions of the general problem: 1) prices vary across retailers, but are constant over the time horizon (p_i); 2) prices vary across time, but are constant across retailers (p_t); and finally 3) prices are constant across all retailers and all time periods (p). We conduct this analysis with a linear demand function, and let $\bar{d}_i = \max_t \{a_{it} - b_{it} * p'\}$, where $p' = \min_{i,t} \{a_{it}/b_{it}\}$, in our experiments to ensure feasibility (due to different price sensitivities, it may be the case that limited and varying capacities across retailers may make a uniform pricing scheme across retailers infeasible). The other parameters are generated according to Table 1. To compare the different settings, the objective value, computational time and the gap between the objective value in the restricted version and the general version ("Gap" column) are

stated in Table 7. To obtain reliable results, five different instances are generated for each problem size and the average results are reported. The basic OA algorithm is used to solve the instances.

According to Table 7, although the computational time is much less in the restricted versions, the profit is considerably reduced in these settings. For the generated instances, the profit losses are generally higher when ignoring variations across retailers (p_i) compared to the case when the time dimension is ignored (p_t). We also observe that the profit reduction percentage generally increases as the problem size gets bigger. Moreover, for the most restricted version, i.e., the same price for all the retailers and all the time periods (p), the profit can be negative. Note that in this setting, problem parameters across retailers and time vary, but the stringent pricing policy does not allow the manufacturer to adjust the prices in line with these changes in the environment. Due to this inflexibility, profits can be significantly reduced.

6. Conclusions

A production-routing problem with price-dependent demand (PRP-PD) is studied in this paper. It is assumed that the demand follows a general convex, differentiable, continuous and strictly decreasing function in price. The problem is modeled as a MINLP, and two algorithms, named as basic OA algorithm and two-phase OA algorithm, are developed to solve the PRP-PD using an OA approach. The efficiency of proposed algorithms in comparison with the standard MINLP solvers implemented in GAMS is demonstrated using a wide range of problem instances, which are generated based on the PRP literature. The obtained results show that the basic OA algorithm outperforms the commercial MINLP solvers in both solution quality and computational time aspects in terms of providing an exact solution. On the other hand, the two-phase OA algorithm provides good quality near-optimal solutions very efficiently, especially for large problem instances that none of the GAMS solvers can provide solutions to in a reasonable time. Such findings prevail both for linear and for nonlinear demand functions. In addition to the analysis of the proposed algorithms' computational performance, another experimental analysis is conducted to investigate the impact of different problem parameters on the optimal solution. According to these experiments, the manufacturer should give higher priority to the retailer who is with less price-sensitive demand and who is closer to the manufacturer. The other observation is that the larger market size and the lower price sensitivity leads to more profit.

Improving the proposed solution algorithm by accelerating the solution of the master problem using techniques such as branch and cut is an opportunity for future research. Considering a stochastic demand function and modifying the outer approximation method to solve it is the other step of extending this paper. Although we have price as a decision variable, our proposed framework can be easily extended to other marketing related decision variables such as advertising. However, dependencies between periods may be more complicated in such settings. For example, it would be an interesting avenue to look at how marketing decisions made in earlier periods would affect demand in later periods. Although our model can handle dynamic pricing, the pricing decisions in period t only affect the demand in period t . It would be interesting to study a problem where marketing decisions would affect demand, hence, setup, production, and delivery problems in the future periods.

CRedit authorship contribution statement

Somayeh Torkaman: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Writing - original

draft, Writing - review & editing. **Mohammad Reza Akbari Jokar:** Conceptualization, Methodology, Validation, Writing - review & editing, Supervision. **Nevin Mutlu:** Methodology, Validation, Formal analysis, Investigation, Writing - original draft, Writing - review & editing, Supervision. **Tom Van Woensel:** Methodology, Validation, Writing - review & editing, Supervision.

Appendix A. Proofs

A.1. Quasiconvexity of equality constraints

In order to show that $\tau_{it}^l H(I_{i,t-1}, \bar{q}_{it}, p_{it}, I_{i,t})$ is jointly quasiconvex, we need to consider three cases. Firstly, if $\tau_{it}^l = 0$, then $\tau_{it}^l H(I_{i,t-1}, \bar{q}_{it}, p_{it}, I_{i,t}) = 0$, which is a constant, hence, trivially quasiconvex. Secondly, if $\tau_{it}^l = -1$, then $\tau_{it}^l H(I_{i,t-1}, \bar{q}_{it}, p_{it}, I_{i,t}) = -H(I_{i,t-1}, \bar{q}_{it}, p_{it}, I_{i,t}) = -I_{i,t-1} - \sum_{k \in K} q_{ikt} + I_{i,t} + d_{it}(p_{it})$. The function is additively separable in different variables; hence, in order to show joint convexity, which is a stronger condition than quasiconvexity, it is sufficient to show that the function is convex in all variables separately. The first three terms in the function are linear in different variables, implying that the function is convex in those variables. The last term is also convex in p_{it} by assumption. Hence, the entire function is jointly convex in the variables, which implies that the function is also jointly quasiconvex in the variables. Finally, if $\tau_{it}^l = 1$, then $\tau_{it}^l H(I_{i,t-1}, \bar{q}_{it}, p_{it}, I_{i,t}) = H(I_{i,t-1}, \bar{q}_{it}, p_{it}, I_{i,t}) = I_{i,t-1} + \sum_{k \in K} q_{ikt} - I_{i,t} - d_{it}(p_{it})$. To show that function $H(I_{i,t-1}, \bar{q}_{it}, p_{it}, I_{i,t})$ is jointly quasi-convex in all variables, it is sufficient to show that all principal minors of the bordered Hessian matrix for $H(I_{i,t-1}, \bar{q}_{it}, p_{it}, I_{i,t})$ are less than equal to zero. Below is the bordered Hessian matrix for $H(I_{i,t-1}, \bar{q}_{it}, p_{it}, I_{i,t})$:

$$G = \begin{bmatrix} 0 & \frac{\partial H}{\partial I_{i,t-1}} & \frac{\partial H}{\partial I_{i,t}} & \frac{\partial H}{\partial p_{it}} & \frac{\partial H}{\partial q_{i1t}} & \cdots & \frac{\partial H}{\partial q_{i|K|t}} \\ \frac{\partial H}{\partial I_{i,t-1}} & \frac{\partial^2 H}{\partial I_{i,t-1}^2} & \frac{\partial^2 H}{\partial I_{i,t-1} \partial I_{i,t}} & \frac{\partial^2 H}{\partial I_{i,t-1} \partial p_{it}} & \frac{\partial^2 H}{\partial I_{i,t-1} \partial q_{i1t}} & \cdots & \frac{\partial^2 H}{\partial I_{i,t-1} \partial q_{i|K|t}} \\ \frac{\partial H}{\partial I_{i,t}} & \frac{\partial^2 H}{\partial I_{i,t} \partial I_{i,t-1}} & \frac{\partial^2 H}{\partial I_{i,t}^2} & \frac{\partial^2 H}{\partial I_{i,t} \partial p_{it}} & \frac{\partial^2 H}{\partial I_{i,t} \partial q_{i1t}} & \cdots & \frac{\partial^2 H}{\partial I_{i,t} \partial q_{i|K|t}} \\ \frac{\partial H}{\partial p_{it}} & \frac{\partial^2 H}{\partial p_{it} \partial I_{i,t-1}} & \frac{\partial^2 H}{\partial p_{it} \partial I_{i,t}} & \frac{\partial^2 H}{\partial p_{it}^2} & \frac{\partial^2 H}{\partial p_{it} \partial q_{i1t}} & \cdots & \frac{\partial^2 H}{\partial p_{it} \partial q_{i|K|t}} \\ \frac{\partial H}{\partial q_{i1t}} & \frac{\partial^2 H}{\partial q_{i1t} \partial I_{i,t-1}} & \frac{\partial^2 H}{\partial q_{i1t} \partial I_{i,t}} & \frac{\partial^2 H}{\partial q_{i1t} \partial p_{it}} & \frac{\partial^2 H}{\partial q_{i1t}^2} & \cdots & \frac{\partial^2 H}{\partial q_{i1t} \partial q_{i|K|t}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial H}{\partial q_{i|K|t}} & \frac{\partial^2 H}{\partial q_{i|K|t} \partial I_{i,t-1}} & \frac{\partial^2 H}{\partial q_{i|K|t} \partial I_{i,t}} & \frac{\partial^2 H}{\partial q_{i|K|t} \partial p_{it}} & \frac{\partial^2 H}{\partial q_{i|K|t} \partial q_{i1t}} & \cdots & \frac{\partial^2 H}{\partial q_{i|K|t}^2} \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 1 & -1 & -d' & 1 & \cdots & 1 \\ 1 & 0 & 0 & 0 & 0 & \cdots & 0 \\ -1 & 0 & 0 & 0 & 0 & \cdots & 0 \\ -d' & 0 & 0 & -d'' & 0 & \cdots & 0 \\ 1 & 0 & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & 0 & 0 & \cdots & 0 \end{bmatrix}$$

The r^{th} principle minor of a bordered matrix is the determinant of the top left-hand corner $(r+1) \times (r+1)$ submatrix. Function H is quasi-convex if all principal minors of the bordered Hessian matrix of H are non-positive in sign.

Let D_r for $r = 1, \dots, |K| + 3$, be the top left-hand corner $(r+1) \times (r+1)$ submatrix of the bordered Hessian matrix of function H . Therefore, $|D_r|$ for $r = 1, \dots, |K| + 3$ is the r^{th} principle minor of this bordered Hessian matrix.

Following the procedure of calculating the determinants of a square matrix, $|D_r|$ for $r = 1, \dots, |K| + 3$ is formulated as below:

$$|D_r| = \sum_{i=1}^{r+1} (-1)^{i+j} d_{ij}^r M_{ij}^r \quad (\text{for a fixed } j)$$

$$= \sum_{j=1}^{r+1} (-1)^{i+j} d_{ij}^r M_{ij}^r \quad (\text{for a fixed } i)$$

Where d_{ij}^r referred to the entry in the i^{th} row and j^{th} column of the matrix D_r , and M_{ij}^r is obtained by taking the determinant of D_r with row i and column j crossed out. According to this formulation, $|D_r|$ for $r = 1, \dots, |K| + 3$ considering the first row of the D_r as the fixed row, is computed as the following:

$$|D_1| = \begin{vmatrix} 0 & 1 \\ 1 & 0 \end{vmatrix} = (-1)^{1+1}(0)(0) + (-1)^{1+2}(1)(1) = 0 - 1 = -1$$

$$|D_2| = \begin{vmatrix} 0 & 1 & -1 \\ 1 & 0 & 0 \\ -1 & 0 & 0 \end{vmatrix} = (-1)^{1+1}(0) \begin{vmatrix} 0 & 0 \\ 0 & 0 \end{vmatrix} + (-1)^{1+2}(1) \begin{vmatrix} 1 & 0 \\ -1 & 0 \end{vmatrix} + (-1)^{1+3}(-1) \begin{vmatrix} 1 & 0 \\ -1 & 0 \end{vmatrix} = 0$$

$$|D_3| = \begin{vmatrix} 0 & 1 & -1 & -d' \\ 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ -d' & 0 & 0 & -d'' \end{vmatrix} = (-1)^{1+1}(0) \begin{vmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -d' \end{vmatrix} + (-1)^{1+2}(1) \begin{vmatrix} 1 & 0 & 0 \\ -1 & 0 & 0 \\ -d' & 0 & -d'' \end{vmatrix} + (-1)^{1+3}(-1) \begin{vmatrix} 1 & 0 & 0 \\ -1 & 0 & 0 \\ -d' & 0 & -d'' \end{vmatrix} + (-1)^{1+4}(-d') \begin{vmatrix} 1 & 0 & 0 \\ -1 & 0 & 0 \\ -d' & 0 & 0 \end{vmatrix} = 0$$

$$|D_4| = \begin{vmatrix} 0 & 1 & -1 & -d' & 1 \\ 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 \\ -d' & 0 & 0 & -d'' & 0 \\ 1 & 0 & 0 & 0 & 0 \end{vmatrix} = (-1)^{1+1}(0) \begin{vmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -d'' & 0 \\ 0 & 0 & 0 & 0 \end{vmatrix} + (-1)^{1+2}(1) \begin{vmatrix} 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ -d' & 0 & -d'' & 0 \\ 1 & 0 & 0 & 0 \end{vmatrix} + (-1)^{1+3}(-1) \begin{vmatrix} 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ -d' & 0 & -d'' & 0 \\ 1 & 0 & 0 & 0 \end{vmatrix} + (-1)^{1+4}(-d') \begin{vmatrix} 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ -d' & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{vmatrix} + (-1)^{1+5}(1) \begin{vmatrix} 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ -d' & 0 & 0 & -d'' \\ 1 & 0 & 0 & 0 \end{vmatrix} = 0$$

For $r > 4$, M_{ij}^r for $j = 1, \dots, r+1$ is calculated as below:

$$M_{11}^r = \begin{vmatrix} 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & -d'' & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 0 \end{vmatrix}$$

$$M_{12}^r = M_{13}^r = \begin{vmatrix} 1 & 0 & 0 & 0 & \dots & 0 \\ -1 & 0 & 0 & 0 & \dots & 0 \\ -d' & 0 & -d'' & 0 & \dots & 0 \\ 1 & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & 0 & \dots & 0 \end{vmatrix}$$

$$M_{14}^r = \begin{vmatrix} 1 & 0 & 0 & 0 & \dots & 0 \\ -1 & 0 & 0 & 0 & \dots & 0 \\ -d' & 0 & 0 & 0 & \dots & 0 \\ 1 & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & 0 & \dots & 0 \end{vmatrix}$$

$$M_{15}^r = \dots = M_{1,(r+1)}^r = \begin{vmatrix} 1 & 0 & 0 & 0 & \dots & 0 \\ -1 & 0 & 0 & 0 & \dots & 0 \\ -d' & 0 & 0 & -d'' & \dots & 0 \\ 1 & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & 0 & \dots & 0 \end{vmatrix}$$

Therefore, $|D_r|$ is calculated as below:

$$\begin{aligned} |D_r| = & (-1)^{1+1}(0) \begin{vmatrix} 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & -d'' & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 0 \end{vmatrix} + (-1)^{1+2}(1) \begin{vmatrix} 1 & 0 & 0 & 0 & \dots & 0 \\ -1 & 0 & 0 & 0 & \dots & 0 \\ -d' & 0 & -d'' & 0 & \dots & 0 \\ 1 & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & 0 & \dots & 0 \end{vmatrix} \\ & + (-1)^{1+3}(-1) \begin{vmatrix} 1 & 0 & 0 & 0 & \dots & 0 \\ -1 & 0 & 0 & 0 & \dots & 0 \\ -d' & 0 & -d'' & 0 & \dots & 0 \\ 1 & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & 0 & \dots & 0 \end{vmatrix} + (-1)^{1+4}(-d') \begin{vmatrix} 1 & 0 & 0 & 0 & \dots & 0 \\ -1 & 0 & 0 & 0 & \dots & 0 \\ -d' & 0 & 0 & 0 & \dots & 0 \\ 1 & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & 0 & \dots & 0 \end{vmatrix} \\ & + (-1)^{1+5}(1) \begin{vmatrix} 1 & 0 & 0 & 0 & \dots & 0 \\ -1 & 0 & 0 & 0 & \dots & 0 \\ -d' & 0 & 0 & -d'' & \dots & 0 \\ 1 & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & 0 & \dots & 0 \end{vmatrix} + \dots + (-1)^{1+(r+1)}(1) \begin{vmatrix} 1 & 0 & 0 & 0 & \dots & 0 \\ -1 & 0 & 0 & 0 & \dots & 0 \\ -d' & 0 & 0 & -d'' & \dots & 0 \\ 1 & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & 0 & \dots & 0 \end{vmatrix} \end{aligned}$$

If we consider the second column in M_{1j}^r for $j = 1, \dots, r+1$ as the fixed column to calculate M_{1j}^r , it is obvious that $M_{1j}^r = 0$, since all entries in the second column of M_{1j}^r are equal to zero. Consequently, $|D_r| = 0$ for $r = 2, \dots, |K| + 3$. As shown $D_1 < 0$, and therefore all principal minors of the bordered Hessian matrix of H are non-positive in sign and function H is quasi-convex.

A.2. Concavity of the revenue functions

To prove the concavity of the revenue functions, we show that the second derivative of the revenue functions are negative. For the linear demand function the revenue function is defined as the following: $r_{it}(p_{it}) = p_{it}(a_{it} - b_{it}p_{it})$, where $a_{it}, b_{it} > 0$. The second derivative of this quadratic function is simply calculated as the following which is obviously negative: $r_{it}^{(Prime)(p_{it})} = -2b_{it} \leq 0$.

As mentioned in Section 5.2, a power function is selected as the nonlinear demand function. Subsequently, the revenue function is defined as follows: $r_{it}(p_{it}) = p_{it}(\delta_{it}p_{it} + \beta_{it})^{-\xi}$, where $\delta_{it}, \beta_{it} > 0$ and $\xi > 1$. The second derivative of revenue function is calculated as the following:

$$\begin{aligned} r_{it}'(p_{it}) &= -\xi \delta_{it} p_{it} (\delta_{it} p_{it} + \beta_{it})^{-\xi-1} + (\delta_{it} p_{it} + \beta_{it})^{-\xi} \\ &= (\delta_{it} p_{it} + \beta_{it})^{-\xi-1} [\delta_{it} p_{it} (1 - \xi) + \beta_{it}] \\ \Rightarrow r_{it}''(p_{it}) &= (-\xi - 1) \delta_{it} (\delta_{it} p_{it} + \beta_{it})^{-\xi-2} [\delta_{it} p_{it} (1 - \xi) + \beta_{it}] \\ &\quad + \delta_{it} (1 - \xi) (\delta_{it} p_{it} + \beta_{it})^{-\xi-1} \\ &= (\delta_{it} p_{it} + \beta_{it})^{-\xi-2} \delta_{it} \xi [\delta_{it} p_{it} (\xi - 1) - 2\beta_{it}] \end{aligned}$$

To prove the concavity, it must be shown that the second derivative is negative, i.e., $r_{it}''(p_{it}) \leq 0$. Since $(\delta_{it} p_{it} + \beta_{it})^{-\xi-2} \delta_{it} \xi \geq 0$, it must hold that $[\delta_{it} p_{it} (\xi - 1) - 2\beta_{it}] \leq 0$, which happens when $p_{it} \leq \frac{2\beta_{it}}{\delta_{it}(\xi-1)}$. Therefore, the revenue function is concave when the price is bounded from above by $2\beta_{it}/(\delta_{it}(\xi-1))$.

Appendix B. Validation of integer cuts

In order to show that Constraint (54) holds, we define two following sets:

$$\begin{aligned} B^v &= \{x_{ijkt}, z_{ikt}, y_t : x_{ijkt}^v, z_{ikt}^v, y_t^v = 1\} \\ NB^v &= \{x_{ijkt}, z_{ikt}, y_t : x_{ijkt}^v, z_{ikt}^v, y_t^v = 0\} \end{aligned}$$

Using the above mentioned sets, we can rewrite Constraint (54) similar to Duran and Grossmann (1986) as the following:

Table 8
Investigating the impact of Demand parameter.

Demand parameter	5 * 3 * 2		7 * 3 * 2		10 * 3 * 2	
	time	profit	time	profit	time	profit
0.0001	31.5	2,154,013	187.2	3,258,076	6211.8	4,544,795
1	29.6	2,154,013	174.8	3,258,076	5766.3	4,544,795
20	13.1	988,865	17.6	1,271,256	285.8	1,866,005
40	12.7	551,668	15.5	697,729	209.9	1,027,945
60	12.1	376,339	15.6	475,581	169.4	702,297
80	11.9	282,511	14.8	358,244	138.1	529,964
100	11.9	224,188	13.1	285,840	128.2	423,445
120	11.8	184,511	12.4	236,740	130	351,157
140	11.9	155,904	12	201,251	131.6	298,884
160	11.7	134,277	11.9	174,406	131	259,377
180	12	117,457	12.2	153,391	130.5	228,630
200	11.9	103,962	12	136,493	128.8	203,904
10000	9.4	0	10.6	0	12.5	0

Table 9
Comparison between basic OA and two-phase OA algorithms for specific gaps ($\epsilon > 0\%$) with a linear demand function.

Size $ N * T * K $	Basic OA			Two-phase OA		
	$\epsilon = 0\%$			$\epsilon = 10\%$		
	time	obj	Gap-opt	time	obj	Gap-opt
1	5'3"2	224,188	0.00%	11.5	217,859	2.82%
2	5'3"3	224,928	0.00%	12	217,650	3.24%
3	5'6"2	280,476	0.00%	12.5	272,494	2.85%
4	5'6"3	324,459	0.00%	12.6	314,306	3.13%
5	5'9"2	602,878	0.04%	14.5	586,827	2.70%
6	5'9"3	604,427	0.03%	14.8	589,303	2.50%
7	7'3"2	285,840	0.00%	11.7	278,193	2.68%
8	7'3"3	285,897	0.00%	13.1	281,569	1.51%
9	7'6"2	420,167	0.03%	13	399,735	4.86%
10	7'6"3	420,715	0.03%	13.3	394,477	6.24%
11	7'9"2	655,256	0.09%	14.7	632,741	3.44%
12	7'9"3	655,911	0.08%	15.3	626,178	4.53%
13	10'3"2	423,445	0.00%	19.9	409,296	3.34%
14	10'3"3	411,146	0.00%	21.9	399,432	2.85%
15	10'6"2	755,528	0.02%	31.2	729,975	3.40%
16	10'6"3	692,591	0.04%	41	684,219	1.21%
17	10'9"2	723,539	0.00%	51.7	706,756	2.32%
18	10'9"3	485,416	0.01%	70	704,668	1.36%
19	12'3"2	2015.6	0.05%	112.6	429,799	1.30%
20	12'3"3	3064.7	0.08%	162.8	418,377	3.96%
21	12'6"2	964,010	0.15%	328.1	946,564	1.81%
22	12'6"3	966,760	0.13%	559.5	961,424	0.55%
23	12'9"2	985,790	0.10%	957.4	974,257	1.17%
24	12'9"3	986,738	0.24%	1627	973,616	1.33%

Basic OA			Two-phase OA		
$\epsilon = 10\%, 5\% \text{ and } 3\%$			$\epsilon = 5\%$		
time	obj	Gap-opt	time	obj	Gap-opt
1	224,188	0.18%	12	221,636	1.14%
2	224,928	0.07%	12.9	221,945	1.92%
3	280,476	0.01%	13.9	274,641	1.33%
4	324,459	0.00%	14	317,168	2.08%
5	602,878	0.00%	14.6	586,827	2.25%
6	604,427	0.06%	14.7	589,303	3.08%
7	285,840	0.00%	12	279,781	3.98%
8	285,897	0.00%	13.2	281,569	2.70%
9	420,167	0.08%	13.2	412,801	2.50%
10	420,715	0.09%	17.5	409,735	2.12%
11	655,256	0.00%	19	638,194	1.84%
12	655,911	0.16%	18.6	642,864	1.51%
13	423,445	0.04%	20.5	642,864	3.35%
14	411,146	0.00%	22	413,495	3.98%
15	755,528	0.00%	25	406,138	2.61%
16	692,591	0.05%	36.7	743,175	1.22%
17	723,539	0.02%	42.2	684,219	2.46%
18	714,413	0.01%	53.9	706,756	1.65%
19	435,444	0.11%	72.4	704,668	1.21%
20	435,369	0.02%	114	429,799	1.30%
21	964,010	0.22%	182.9	429,820	2.05%
22	966,760	0.57%	327.5	946,564	1.34%
23	985,790	0.46%	560.4	961,424	2.47%
24	986,738	0.75%	961.8	974,257	0.55%

Basic OA			Two-phase OA		
$\epsilon = 3\%$			$\epsilon = 3\%$		
time	obj	Gap-opt	time	obj	Gap-opt
1	224,188	0.00%	12.1	221,636	1.14%
2	224,928	0.00%	13.5	222,38	1.92%
3	280,476	0.00%	18.1	276,396	1.33%
4	324,459	0.00%	18.4	319,491	2.08%
5	602,878	0.04%	22.7	591,369	2.25%
6	604,427	0.03%	26.8	592,132	3.08%
7	285,840	0.00%	13.3	280,268	3.98%
8	285,897	0.00%	15.3	283,379	2.70%
9	420,167	0.08%	20	415,974	2.50%
10	420,715	0.09%	20.8	413,107	2.12%
11	655,256	0.00%	62.6	644,988	1.84%
12	655,911	0.16%	111.1	646,988	1.51%
13	423,445	0.04%	29.6	416,339	3.35%
14	411,146	0.00%	32.3	406,443	3.98%
15	755,528	0.00%	42.9	748,188	2.61%
16	692,591	0.05%	52	685,157	1.22%
17	723,539	0.02%	62.8	713,322	2.46%
18	714,413	0.01%	98.1	704,668	1.21%
19	435,444	0.11%	126	430,209	1.30%
20	435,369	0.02%	183.6	429,820	2.05%
21	964,010	0.22%	350.8	955,847	1.34%
22	966,760	0.57%	559.5	961,424	2.47%
23	985,790	0.46%	1178.5	981,254	0.55%
24	986,738	0.75%	1833.6	979,570	0.99%

$$\sum_{B^v} \left(\sum_{i \in N} \sum_{j \in N} \sum_{k \in K} \sum_{t \in T} x_{ijkt} + \sum_{i \in N} \sum_{k \in K} \sum_{t \in T} z_{ikt} + \sum_{t \in T} y_t \right) - \sum_{NB^v} \left(\sum_{i \in N} \sum_{j \in N} \sum_{k \in K} \sum_{t \in T} x'_{ijkt} + \sum_{i \in N} \sum_{k \in K} \sum_{t \in T} z'_{ikt} + \sum_{t \in T} y'_t \right) \leq |B^v| - 1$$

$$\forall i, j \in N, \forall k \in K, \forall t \in T, \forall v \in \{0, \dots, l-1\}$$

The above mentioned constraint is obviously violated for the combination of the binary variables in the iteration v , since for this combination the left-hand side of the equation is equal to $|B^v|$ which is greater than $|B^v| - 1$. Moreover, we should show that this constraint is not violated by other combinations rather than the combination of the binary variables in the iteration v . Suppose that there is another combination like $(x'_{ijkt}, z'_{ikt}, y'_t)$ such that $(x'_{ijkt}, z'_{ikt}, y'_t) \neq (x_{ijkt}^v, z_{ikt}^v, y_t^v)$ and the above constraint is violated for this new combination. Since we assume that these two combinations are different, some of the variables with 1 value in previous combination are zero (n variables) in new combination and vice versa (m variables with 0 value are 1 in new combination). Therefore, the violated constraint for the new combination will be written as the following:

$$\sum_{B^v} \left(\sum_{i \in N} \sum_{j \in N} \sum_{k \in K} \sum_{t \in T} x'_{ijkt} + \sum_{i \in N} \sum_{k \in K} \sum_{t \in T} z'_{ikt} + \sum_{t \in T} y'_t \right) - \sum_{NB^v} \left(\sum_{i \in N} \sum_{j \in N} \sum_{k \in K} \sum_{t \in T} x'_{ijkt} + \sum_{i \in N} \sum_{k \in K} \sum_{t \in T} z'_{ikt} + \sum_{t \in T} y'_t \right) > |B^v| - 1$$

$$\forall i, j \in N, \forall k \in K, \forall t \in T, \forall v \in \{0, \dots, l-1\}$$

Therefore,

$$|B^v| - n - m > |B^v| - 1; n, m > 0$$

$$n + m < 1 \Rightarrow n = m = 0$$

It means that the new combination cannot be different from the combination of the binary variables in the iteration v and the constraint is only violated for this combination.

Appendix C. Additional computational results

C.1. Sensitivity analysis on demand parameter

In Table 1, we let $\bar{d}_i = \sum_t A_{it} / (100|T|)$. In this section, we provide a sensitivity analysis on the parameter that was initially assumed as 100. Problem instances with the sizes $5 \times 3 \times 2$, $7 \times 3 \times 2$ and $10 \times 3 \times 2$ are used to conduct this analysis using the basic OA algorithm with a linear demand function. For each problem size, 5 instances are considered and the average results are reported in Table 8 (all other parameters are generated based on Table 1). The "Demand Parameter" column reports the different values for this parameter.

According to Table 8, by increasing the demand parameter and subsequently decreasing capacities, the profit decreases. To investigate the extreme values for this parameters, extremely small and large values for this parameter are also considered, such as 0.0001 and 10,000. As illustrated in Table 8, for extremely large value of demand parameter (highly limited capacities), the profit is zero since the generated demand would be zero in these cases. On the other hand, decreasing the demand parameter (increasing the capacities) more than a specific amount leads to no more profit because of the market saturation. Note that the computational time is greater for the less limited capacities (smaller demand parameters) since considering more capacities enlarges the feasible region which leads to more computational effort.

Table 10
Comparison between basic OA and two-phase OA algorithms for specific gaps ($\varepsilon > 0\%$) with a nonlinear demand function.

Size $ N * T * K $	Basic OA			Basic OA			Two-phase OA												
	$\epsilon = 0\%$			$\epsilon = 10\%$			$\epsilon = 5\%$			$\epsilon = 10\%$			$\epsilon = 5\%$						
	time	obj	Cap-up	time	obj	Cap-up	time	obj	Cap-up	time	obj	Cap-up	time	obj	Cap-up				
1	5'3"2	17.0	284578	12.6	284446	0.05%	4.18%	13.3	284546	0.01%	4.03%	12.8	283912	0.23%	4.82%	13.6	284481	0.03%	4.16%
2	5'6"2	18.4	582042	13.4	580894	0.20%	5.26%	14.4	581768	0.05%	4.87%	13.3	577001	0.87%	6.62%	16.4	582034	0.00%	4.86%
3	7'3"2	20.1	417694	13.2	417374	0.08%	4.87%	15.4	417669	0.01%	4.60%	13.1	414218	0.83%	6.14%	15.4	417367	0.08%	4.76%
4	7'6"2	84.5	872881	35.2	871501	0.16%	4.97%	36.3	872062	0.09%	4.78%	14.6	864299	0.98%	6.44%	28.2	871918	0.11%	4.66%
5	10'3"2	486.9	563939	70.4	561334	0.46%	5.72%	84.5	563686	0.04%	4.69%	20.8	556232	1.37%	6.76%	63.4	563234	0.13%	4.92%
6	10'6"2	5448.3	1,130,022	128.5	1,127,586	0.22%	3.46%	131.16	1,128,253	0.16%	3.32%	129.4	1,118,152	1.05%	4.39%	82.8	1,125,623	0.39%	3.63%
7	12'3"2	7200	697,194	172.22	696,694	0.07%	2.10%	172.22	696,694	0.07%	2.10%	129.4	693,049	0.59%	3.25%	143.7	696,215	0.14%	2.34%
8	12'6"2	7200	1,452,402	5643.8	1,451,710	0.05%	3.74%	5643.8	1,451,710	0.05%	3.74%	366.5	1,441,736	0.73%	4.65%	439.6	1,448,399	0.28%	4.12%

Fig. 4. Power demand function.

C.2. Comparison of the basic OA and two-phase OA algorithms for specific gaps

In [Tables 2 and 3](#), the results of the basic OA algorithm for the optimal solution ($\varepsilon = 0$) are compared with the results of the two-phase OA algorithm for specific optimality gaps ($\varepsilon > 0$). In order to provide insight into the performance of the basic OA algorithm with specific gaps, we ran additional experiments by setting optimality gaps as $\varepsilon = 10\%, 5\%, 3\%$. In [Table 2](#), we compare our results to the performance of the two-phase OA algorithm for the same optimality gaps, considering a linear demand function as in [Table 2](#) (note that the results regarding the two-phase OA algorithm in [Table 2](#) are presented again in [Table 9](#) to facilitate a comparison). The same analysis is conducted for the nonlinear demand function presented in [Table 3](#).

In Table 9, there is only 1 column for the basic OA with $\varepsilon > 0$, since the results are the same for all three gaps (Gap-up is at most 1% for this algorithm). According to Table 9, even by setting a high amount such as 10% for the ε in basic OA algorithm, the obtained solutions are very close to the optimal solutions. Although the solutions obtained by the basic OA are very close to the optimal solutions and reported gaps are less than those of two-phase OA, the computational time is considerably higher for the basic OA.

According to Table 10, although the results of basic OA for $\varepsilon > 0$ are more similar to those of two-phase OA (compared to linear demand function), the solutions provided by basic OA are closer to the optimal solutions and providing them is more time consuming. Therefore, as mentioned in Section 5.1, it is suggested to use the basic OA for solving the problem optimally for small problem instances, but to apply the two-phase OA to provide good solutions fast for medium and large problem instances.

C.3. Nonlinear demand curve

The power demand function for $\xi = 2$, $\delta = 0.000348$, and $\beta = 0.034816$ is depicted in Fig. 4. For this demand function, the market size is $a = 825$.

References

- Abbi, N., Archetti, C., Dauzère-Pérès, S., Feillet, D., 2014. A two-phase iterative heuristic approach for the production routing problem. *Transportation Science* 49 (4), 784–795.
- Adulyasak, Y., Cordeau, J.-F., Jans, R., 2012. Optimization-based adaptive large neighborhood search for the production routing problem. *Transportation Science* 48 (1), 20–45.
- Adulyasak, Y., Cordeau, J.-F., Jans, R., 2014. Formulations and branch-and-cut algorithms for multivehicle production and inventory routing problems. *INFORMS Journal on Computing* 26 (1), 103–120.
- Adulyasak, Y., Cordeau, J.-F., Jans, R., 2015a. Benders decomposition for production routing under demand uncertainty. *Operations Research* 63 (4), 851–867.

- Adulyasak, Y., Cordeau, J.-F., Jans, R., 2015b. The production routing problem: a review of formulations and solution algorithms. *Computers & Operations Research* 55, 141–152.
- Ahn, H.-S., Gümüş, M., Kaminsky, P., 2007. Pricing and manufacturing decisions when demand is a function of prices in multiple periods. *Operations Research* 55 (6), 1039–1057.
- Archetti, C., Bertazzi, L., Laporte, G., Speranza, M.G., 2007. A branch-and-cut algorithm for a vendor-managed inventory-routing problem. *Transportation Science* 41 (3), 382–391.
- Archetti, C., Bertazzi, L., Paletta, G., Speranza, M.G., 2011. Analysis of the maximum level policy in a production-distribution system. *Computers & Operations Research* 38 (12), 1731–1746.
- Avci, M., Yildiz, S.T., 2019. A matheuristic solution approach for the production routing problem with visit spacing policy. *European Journal of Operational Research*.
- Bajwa, N., Sox, C.R., Ishfaq, R., 2016. Coordinating pricing and production decisions for multiple products. *Omega* 64, 86–101.
- Bard, J.F., Nananukul, N., 2009. Heuristics for a multiperiod inventory routing problem with production decisions. *Computers & Industrial Engineering* 57 (3), 713–723.
- Bard, J.F., Nananukul, N., 2010. A branch-and-price algorithm for an integrated production and inventory routing problem. *Computers & Operations Research* 37 (12), 2202–2217.
- Biller, S., Chan, L.M.A., Simchi-Levi, D., Swann, J., 2005. Dynamic pricing and the direct-to-customer model in the automotive industry. *Electronic Commerce Research* 5 (2), 309–334.
- Bizfluent, 2017. What causes fluctuating unit cost? Date accessed: September 19, 2019. URL: <https://bizfluent.com/13309664/what-causes-fluctuating-unit-cost..>
- Bonami, P., Biegler, L.T., Conn, A.R., Cornuéjols, G., Grossmann, I.E., Laird, C.D., Lee, J., Lodi, A., Margot, F., Sawaya, N., Wächter, A., 2008. An algorithmic framework for convex mixed integer nonlinear programs. *Discrete Optimization* 5 (2), 186–204.
- Boudia, M., Louly, M.A.O., Prins, C., 2007. A reactive grasp and path relinking for a combined production–distribution problem. *Computers & Operations Research* 34 (11), 3402–3419.
- Boudia, M., Louly, M.A.O., Prins, C., 2008. Fast heuristics for a combined production planning and vehicle routing problem. *Production Planning and Control* 19 (2), 85–96.
- Brahimi, N., Aouam, T., 2016. Multi-item production routing problem with backordering: a milp approach. *International Journal of Production Research* 54 (4), 1076–1093.
- Çetinkaya, S., Üster, H., Easwaran, G., Keskin, B.B., 2009. An integrated outbound logistics model for frito-lay: coordinating aggregate-level production and distribution decisions. *Interfaces* 39 (5), 460–475.
- Chandra, P., 1993. A dynamic distribution model with warehouse and customer replenishment requirements. *Journal of the Operational Research Society* 44 (7), 681–692.
- Chandra, P., Fisher, M.L., 1994. Coordination of production and distribution planning. *European Journal of Operational Research* 72 (3), 503–517.
- Chitsaz, M., Cordeau, J.-F., Jans, R., 2019. A unified decomposition matheuristic for assembly, production, and inventory routing. *INFORMS Journal on Computing* 31 (1), 134–152.
- Dayarian, I., Desaulniers, G., 2019. A branch-price-and-cut algorithm for a production-routing problem with short-life-span products. *Transportation Science* 53 (3), 829–849.
- Deng, S., Yano, C.A., 2006. Joint production and pricing decisions with setup costs and capacity constraints. *Management Science* 52 (5), 741–756.
- Díaz-Madroño, M., Peidro, D., Mula, J., 2015. A review of tactical optimization models for integrated production and transport routing planning decisions. *Computers & Industrial Engineering* 88, 518–535.
- Duran, M.A., Grossmann, I.E., 1986. An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Mathematical Programming* 36 (3), 307–339.
- Eliashberg, J., Steinberg, R., 1993. Marketing-production joint decision-making. *Handbooks in Operations Research and Management Science* 5, 827–880.
- Floudas, C.A., 1995. *Nonlinear and Mixed-Integer Optimization: Fundamentals and Applications*. Oxford University Press.
- Forbes, 2015. Sales and operations planning continues to evolve. Date accessed: September 19, 2019. URL: <https://www.forbes.com/sites/stevebanker/2015/07/16/sales-operations-planning-continues-to-evolve/#75e576316e27..>
- Fumero, F., Vercellis, C., 1999. Synchronized development of production, inventory, and distribution schedules. *Transportation Science* 33 (3), 330–340.
- Geunes, J., Merzifonluoğlu, Y., Romeijn, H.E., 2009. Capacitated procurement planning with price-sensitive demand and general concave-revenue functions. *European Journal of Operational Research* 194 (2), 390–405.
- Geunes, J., Romeijn, H.E., Taaffe, K., 2006. Requirements planning with pricing and order selection flexibility. *Operations Research* 54 (2), 394–401.
- Gilbert, S.M., 1999. Coordination of pricing and multi-period production for constant priced goods. *European Journal of Operational Research* 114 (2), 330–337.
- Grossmann, I.E., 2002. Review of nonlinear mixed-integer and disjunctive programming techniques. *Optimization and Engineering* 3 (3), 227–252.
- Insight2profit, 2019. How capacity utilization can and should drive pricing decisions. Date accessed: September 19, 2019. URL: <https://www.insight2profit.com/blog/how-capacity-utilization-can-and-should-drive-pricing-decisions..>
- Kocis, G.R., Grossmann, I.E., 1987. Relaxation strategy for the structural optimization of process flow sheets. *Industrial & Engineering Chemistry Research* 26 (9), 1869–1880.
- Kunreuther, H., Richard, J.F., 1971. Optimal pricing and inventory decisions for non-seasonal items. *Econometrica* (pre-1986), 39(1):173..
- Lapide, L., 2006. Demand management: Matching supply and demand over time. Date accessed: September 19, 2019. URL: <https://ctl.mit.edu/pub/paper/demand-management-matching-supply-and-demand-over-time..>
- Lei, L., Liu, S., Ruszczyński, A., Park, S., 2006. On the integrated production, inventory, and distribution routing problem. *IIE Transactions* 38 (11), 955–970.
- Li, Y., Chu, F., Chu, C., Zhu, Z., 2019. An efficient three-level heuristic for the large-scaled multi-product production routing problem with outsourcing. *European Journal of Operational Research* 272 (3), 914–927.
- Miranda, P.L., Morabito, R., Ferreira, D., 2018. Optimization model for a production, inventory, distribution and routing problem in small furniture companies. *TOP* 26 (1), 30–67.
- Qiu, Y., Qiao, J., Pardalos, P.M., 2017. A branch-and-price algorithm for production routing problems with carbon cap-and-trade. *Omega* 68, 49–61.
- Ruokokoski, M., Solyali, O., Cordeau, J.-F., Jans, R., Süral, H., 2010. Efficient formulations and a branch-and-cut algorithm for a production-routing problem. *GERAD Technical Report G-2010-66*.
- Russell, R.A., 2017. Mathematical programming heuristics for the production routing problem. *International Journal of Production Economics* 193, 40–49.
- Solyali, O., Süral, H., 2009. A relaxation based solution approach for the inventory control and vehicle routing problem in vendor managed systems. In: *Modeling, Computation and Optimization*. World Scientific, pp. 171–189.
- Solyali, O., Süral, H., 2017. A multi-phase heuristic for the production routing problem. *Computers & Operations Research* 87, 114–124.
- Thomas, J., 1970. Price-production decisions with deterministic demand. *Management Science* 16 (11), 747–750.
- Vahdani, B., Niaki, S., Aslzanade, S., 2017. Production-inventory-routing coordination with capacity and time window constraints for perishable products: Heuristic and meta-heuristic algorithms. *Journal of Cleaner Production* 161, 598–618.
- Van den Heuvel, W., Wagelmans, A.P., 2006. A polynomial time algorithm for a deterministic joint pricing and inventory model. *European Journal of Operational Research* 170 (2), 463–480.
- Whitin, T.M., 1955. Inventory control and price theory. *Management Science* 2 (1), 61–68.
- Wu, X., Xu, H., Chu, C., Zhang, J., 2017. Dynamic lot-sizing models with pricing for new products. *European Journal of Operational Research* 260 (1), 81–92.
- Yano, C.A., Gilbert, S.M., 2005. Coordinated pricing and production/prerequisite decisions: a review. In: *Managing Business Interfaces*. Springer, pp. 65–103.
- Zhang, Q., Sundaramoorthy, A., Grossmann, I.E., Pinto, J.M., 2017. Multiscale production routing in multicommodity supply chains with complex production facilities. *Computers & Operations Research* 79, 207–222.