Solving a Large Multi-product Production-Routing Problem with Delivery Time Windows

Fábio Neves-Moreira[a,*], Bernardo Almada-Lobo[a], Jean-François Cordeau[b], Luís Guimarães[a], Raf Jans[b]

[a] *INESC TEC and Faculty of Engineering, University of Porto, R. Dr. Roberto Frias, 4200 Porto, Portugal*
[b] *HEC Montréal and CIRRELT, 3000 Chemin de la Côte-Sainte-Catherine, Montréal, Canada*

Abstract

Even though the joint optimization of sequential activities in supply chains is known to yield significant cost savings, the literature concerning optimization approaches that handle the real-life features of industrial problems is scant. The problem addressed in this work is inspired by industrial contexts where vendor-managed inventory policies are applied. In particular, our study is motivated by a meat producer whose supply chain comprises a single meat processing centre with several production lines and a fleet of vehicles that is used to deliver different products to meat stores spread across the country. A considerable set of characteristics, such as product family setups, perishable products, and delivery time windows, needs to be considered in order to obtain feasible integrated plans. However, the dimensions of the problem make it impossible to be solved exactly by current solution methods. We propose a novel three-phase methodology to tackle a large Production-Routing Problem (PRP) combining realistic features for the first time. In the first phase, we attempt to reduce the size of the original problem by simplifying some dimensions such as the number of products, locations and possible routes. In the second phase, an initial PRP solution is constructed through a problem decomposition comprising several inventory-routing problems and one lot-sizing problem. In the third phase, the initial solution is improved by different mixed-integer programming models which focus on small parts of the original problem and search for improvements in the production, inventory management and transportation costs. Our solution approach is tested both on simpler instances available in the literature and on real-world instances containing additional details, specifically developed for a European company's case study. By considering an integrated approach, we achieve global cost savings of 21.73% compared to the company's solution.

Keywords: production-routing, time windows, perishability, fix-and-optimize, matheuristic, case study

1. Introduction

Competitiveness is intimately connected with the ability to be prepared, to predict market changes, and to be ready for adopting fast changes in various operations. Today, competitiveness is being considered at a supply chain level because no single organizational unit is solely responsible for the competitiveness of its products and services [33]. Provided that companies are aware of the latter facts, the receptiveness for systematic and optimized planning processes has been steadily increasing in the last few years. A recent trend in operations research is to integrate and coordinate various planning problems in order to obtain better plans, trading-off pros and cons from a much wider and inclusive perspective [27]. Indeed, in the past, most optimization processes comprised a number of problems that were solved sequentially. In most cases, the effects caused by decisions fixed in previous stages of the planning process were not even measured. Therefore, most entities were not aware that optimizing a certain problem could prevent the achievement of better solutions for subsequent problems and, consequently, a better global solution for the whole planning process.

The literature has been reporting various experiments in which considerable cost savings arise from planning integration [16, 34, 21]. Activities are optimized with a monolithic model, as opposed to sequentially optimizing parts of the global problem (Figure 1). Logically, the impressive results obtained

*Corresponding author
Email address: fnevesmoreira@fe.up.pt (Fábio Neves-Moreira)

by monolithic models contributed to augment the interest in integrated planning processes. However, the enormous set of decisions and factors to be considered often results in intractable problems [31].

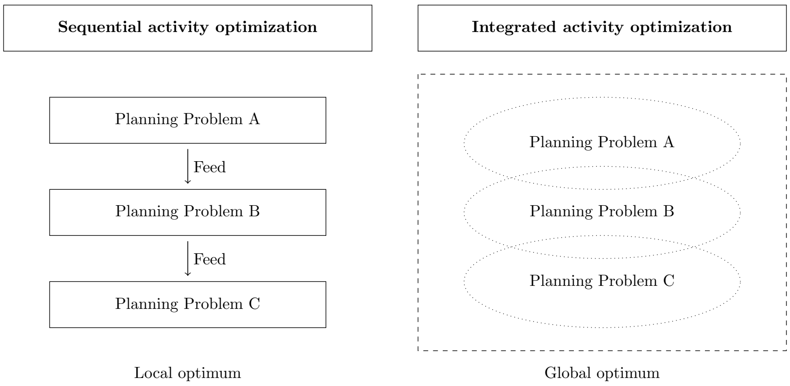


Figure 1: Sequential activity optimization and integrated activity optimization planning processes. Integrated planning allows for lower global cost but poses a more difficult problem to solve.

The Production-Routing Problem (PRP) is an integrated planning problem which jointly optimizes production, inventory management and transportation decisions. The integration of these activities is particularly interesting in a Vendor-Managed Inventory (VMI) context where inventories held at retail sites are managed by a single entity, usually the supplier of the products. The supplier produces a set of products deciding whether it sends them directly to be stocked at retail sites or if it stocks them in its own warehouse in order to distribute them at later periods. Therefore, as it is shown in Figure 2, typical decisions comprised by the PRP include (1) when and how much to produce; (2) when and how much to deliver to each retail site; (3) how to route vehicles such that production, inventory and transportation costs are minimized while meeting retail sites' demand. These decisions are usually to be made during a planning horizon composed of several days.

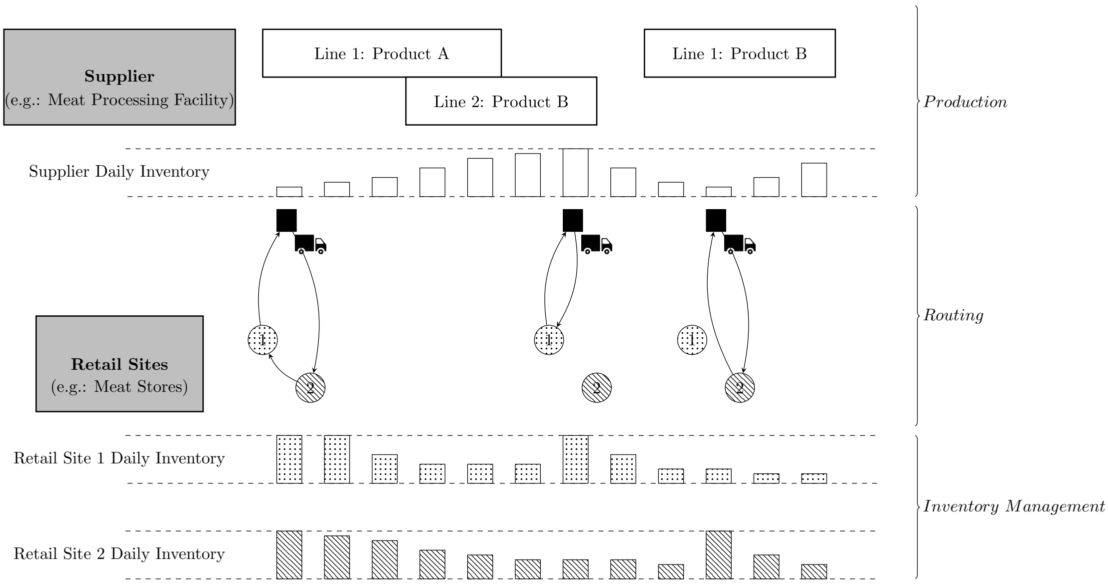


Figure 2: Decisions to be made in a PRP. Production, inventory management, and transportation are coordinated to deliver cost-effective solutions from a monolithic perspective.

This problem is extremely challenging as it integrates two classic optimization problems, the Lot-Sizing Problem (LSP) and the Vehicle Routing Problem (VRP), which were proposed by Wagner and Whitin [35] and by Dantzig and Ramser [19], respectively. Similarly to its particular case where production is not taken into account, the Inventory-Routing Problem (IRP), the PRP is NP-hard, as it demands the solution of several VRPs [4]. However, Chandra and Fisher [16] reported 3 to 20% cost savings coming from integration, which is a strong motivation to study the PRP in the context of realistic and large planning problems.

We aim to propose a novel mathematical formulation considering product and family setups when producing perishable products that are to be delivered to retail sites within certain time windows. The set of features comprised in the model allows great flexibility, which enables the possibility to apply it to several process industries. However, the inherent complexity of such a PRP requires the development of advanced solution methods in order to efficiently explore its large and heterogeneous structure. The proposed solution approach consists of three main phases comprising a size reduction phase, an initial solution construction phase, and an improvement phase based on a fix-and-optimize (F&O) approach.

The developed algorithm is first tested on standard PRP instances available in the literature and then on real-world instances considered in the case study of a European meat store chain that motivated the underlying research. The real-world challenge clearly shows the necessity of considering sets of realistic features in the context of the PRP where any kind of extension is rare to be found in the literature. Additionally, the current practice of the company is based on decoupled plans considering demand for a single period ahead. Therefore, we devise an integrated approach so as to show its advantages with respect to various practical dimensions.

The contributions of this paper are fourfold:

- We propose a novel mathematical formulation for the PRP with realistic features including multiple vehicles performing routes with time windows, multiple perishable products, and multiple production lines with different specifications.
- We present a decomposition approach for large PRPs where available data are used to reduce the problem size and divide the original problem into several tractable subproblems. The decomposition requires the solution of several IRPs and one LSP.
- We provide an improvement matheuristic based on a F&O scheme, capable of exploring the solution space of large PRPs which are intractable for the best general-purpose solvers available.
- The algorithm is tested on multi-product multi-vehicle instances available in the literature and validated with real-world instances belonging to the case study of a European vertical meat store chain.

The remainder of this paper is organized as follows. In Section 2 a literature review is presented, focusing on different extensions and real-world applications of planning problems integrating production, inventory management and distribution. Section 3 gives a description of the general PRP to be tackled in this work. Section 4 presents the proposed solution approach to efficiently solve large PRPs. The computational experiments obtained in the literature instances are discussed in Section 5. Section 6 details the case study of a meat store chain. Finally, Section 7 summarizes the main achievements and conclusions, pointing out future research opportunities.

2. Literature Review

In this research, we are particularly interested in the PRP, which integrates production, inventory management and routing decisions [4]. Chandra and Fisher [16] are amongst the first attempts to integrate production decisions with distribution, reporting considerable gains coming from activity integration. Since then, several authors addressed supply chain coordination [34, 21, 8, 11, 12, 13] and there are some works reporting case studies where large gains were obtained [15, 14]. For a review on the origins of the PRP, the reader is referred to the work of Sarmiento and Nagi [30]. Although there are some PRP variants, we focus on problems where lot sizing, inventory management, and explicit routing decisions are integrated at an operational level.

Lei et al. [26] study a production, inventory, and distribution routing problem and propose a two-phase solution approach. In the first phase, only direct shipments are considered and in the second phase, an associated consolidation problem is solved in order to eliminate direct shipment inefficiencies. Unlike most uncoupled approaches, this one does not completely separate production from distribution. Bard and Nananukul [9] present a comparative analysis of various heuristics for the PRP. Since optimal solutions could not be achieved with exact methods, a two-step procedure first allocates daily delivery quantities and solves a VRP afterwards. They show that the IRP component can be solved efficiently with a branch-and-price framework. Absi et al. [1] propose a heuristic which iteratively focuses on production and distribution decisions. The production part is modelled as a Capacitated Lot-Sizing Problem (CLSP) whereas the daily distribution is modelled as a Travelling Salesman Problem (TSP) or a VRP. A single-item PRP under the Maximum Level (ML) policy is studied. Both the single and multiple vehicle versions are explored. Computational results show that this solution approach outperforms previously proposed methods. Armentano et al. [7] use a combination of Tabu Search (TS) and a Path Relinking (PR) procedure and test it on instances with up to 10 products. They introduce a mathematical model which is able to consider multiple vehicles and products but resort to a phased approach where a production problem is solved before an unlimited fleet distribution problem. Instances with up to 15 retail sites, 14 periods and five products are solved with less than 2% gaps based on CPLEX optimal solutions. While CPLEX needs more than four hours to compute the solutions, the heuristic only needs less than 30 seconds. Adulyasak et al. [3] introduce an Adaptive Large Neighbourhood Search (ALNS) heuristic for the PRP which handles setup variables by an enumeration scheme and routing variables with upper-level search operators. The continuous variables are computed in a network flow subproblem. The proposed algorithm outperforms existing approaches, computing superior quality solutions in short amounts of time. Computational tests were performed on instances with up to 200 retail sites, 13 vehicles and 20 periods. Recently, Qiu et al. [28] address a pollution PRP considering carbon emissions. The authors propose a mathematical formulation and solve it by means of a branch-and-price heuristic. Managerial insights are provided for instances with 14 retail sites and 6 periods, showing that it is possible to reduce carbon emissions and operational costs simultaneously.

The above-mentioned papers consider heuristic methods to tackle the integrated planning problems. Given the complexity of integrated planning, these are still the most suitable methods to address large instances of this problem. Exact approaches are quite rare in the context of the PRP and the size of the problems that are possibly solved is obviously smaller. Their complexity seems to have repelled researchers from studying them. Indeed, few exact methods are found in the literature and the problems they address are usually simplified in the number of entities or in the type of features considered. Solyalı and Süral [32] propose a mathematical formulation and a Lagrangian relaxation based approach to solve a single-vehicle, single-product PRP. Although the supplier is able to decide order quantities, the problem is simplified because the production activity is uncapacitated. The authors tackle instances with up to 50 retail sites and 30 periods with constant demands. Bard and Nananukul [9, 10] propose a branch-and-price algorithm for a PRP. This algorithm is a combination of a Dantzig-Wolfe (D-W) decomposition with traditional Branch-and-Bound (B&B). Different methods to obtain initial solutions, branching rules and rounding schemes are tested. After tuning, their approach is able to solve instances with up to 50 retail sites and 8 periods. Ruokokoski et al. [29] present efficient formulations and a branch-and-cut algorithm for the PRP. They consider a problem with uncapacitated plants and vehicles and test new formulations strengthened by valid inequalities. Randomly generated instances with up to 40 customers and 15 periods or with 80 customers and eight periods are solved to optimality. A heuristic algorithm is also tested, obtaining solutions with an average cost increase of 0.33% within less than 1% of the time required by the exact approach. Archetti et al. [6] develop a hybrid heuristic for the PRP with ML policy which quickly obtains high quality solutions by solving the production and distribution problems sequentially. The authors compare the single-vehicle case with an exact solution method based on a branch-and-cut scheme. Note that the latter two approaches only create delivery routes separately, after deciding the quantities to be delivered to each retail site. Adulyasak et al. [2] tackle the multi-vehicle PRP. The authors present vehicle and non-vehicle index formulations strengthening them using symmetry breaking constraints and several cuts, respectively. An ALNS heuristic is used to build initial solutions which are improved by a branch-and-cut algorithm. Instances with around 30 customers, 3 vehicles, and 3 periods are solved both for the Order-up-to Level (OU) and ML policies. Adulyasak et al. [5] tackle a PRP with uncertain demands. Production setups and customer visit schedules are defined in the first stage while the production and delivery quantities are determined in subsequent stages. A Benders decomposition approach using lower-bound lifting inequalities, scenario cuts and Pareto-optimal cuts is implemented. Besides this paper, few other papers deal with PRPs with uncertainty. Adulyasak et al. [4] provide a review of formulations and suggest future research directions regarding the PRP.

The importance of considering realistic constraints is undeniable in several industries. However, the literature is still scarce when dealing with realistic integrated planning at an operational level. Furthermore, despite their increasing popularity, matheuristics for the PRP are still rare. We provide a summery of the algorithms proposed in the last decade in Table 1.

Table 1: Main recent PRP approaches

Authors	Production			Inventory		Routing		Solution Method	
	#Plants	#Prod	C.	Policy	C.	#Vehicles	C.	Type	Approach
Lei et al. [26]	Multiple	Single	✓	ML		Limited (Het)	✓	H	Decomposition
Boudia et al. [11]	Single	Single	✓	ML	✓	Limited (Hom)	✓	H	GRASP
Boudia et al. [12]	Single	Single	✓	ML	✓	Limited (Hom)	✓	H	Decomposition
Bard and Nananukul [8]	Single	Single	✓	ML	✓	Limited (Hom)	✓	H	Tabu search
Boudia and Prins [13]	Single	Single	✓	ML	✓	Limited (Hom)	✓	H	Memetic algorithm
Solyalı and Süral [32]	Single	Single		OU	✓	Limited (Hom)	✓	H/L	Lagrangean relaxation
Bard and Nananukul [9, 10]	Single	Single	✓	ML	✓	Limited (Hom)	✓	H/L	Branch-and-price
Ruokokoski et al. [29]	Single	Single		ML		Single		E	Branch-and-cut
Armentano et al. [7]	Single	Multiple	✓	ML	✓	Limited (Hom)	✓	H	Tabu search / Path relinking
Archetti et al. [6]	Single	Single		ML/OU	✓	Single	✓	E/H	Branch-and-cut / MIP heuristic
Adulyasak et al. [3]	Single	Single	✓	ML	✓	Multiple (Hom)	✓	H	ALNS
Adulyasak et al. [2]	Single	Single	✓	ML/OU	✓	Multiple (Hom)	✓	E/H	Branch-and-cut / ALNS
Absi et al. [1]	Single	Single	✓	ML/OU	✓	Multiple (Hom)	✓	H	Iterative MIP heuristic
Adulyasak et al. [5]	Single	Single	✓	ML	✓	Multiple (Hom)	✓	E	Benders-based branch-and-cut
Our approach	Single	Multiple	✓	ML	✓	Multiple (Het)	✓	MH	Fix-And-Optimize

Legend:
Prod. - Products | C. - Capacitated | Hom. - Homogeneous | Het. - Heterogeneous | H - Heuristic | E - Exact | MH - Matheuristic | L - Lower bound computation

3. Problem Description and Formulation

In this section, we first define the problem in general terms. We start by describing the relevant entities as well as the parameters that are related to them. Afterwards the constraints to capture production and routing decisions are presented, as well as the linking constraints to connect both dimensions of the problem.

3.1. Notation

The PRP considered in this paper can be defined on a complete directed graph $G = (V, A)$ where V represents a set of locations including a supplier and retail sites indexed by $i \in \{0, .., n\}$, and $A = \{(i, j) : i, j \in V, i \neq j\}$ is the set of arcs. The supplier manages the set of retail sites $\mathcal{V}' = V \setminus \{0\}$ applying a VMI policy and it owns a set of production lines $\mathcal{M} = \{0, ..., |\mathcal{M}|\}$ which are used to produce a set of product families $\mathcal{F} = \{1, ..., |\mathcal{F}|\}$ over a finite horizon $\mathcal{T} = \{1, ..., |\mathcal{T}|\}$. Each family is composed of several products which belong to the entire set of products $\mathcal{P} = \{1, ..., |\mathcal{P}|\}$ to be delivered by the supplier. Each line m can only produce some products belonging to the set $\mathcal{P}_m$. Deliveries are made by a fleet of vehicles $\mathcal{K} = \{1, ..., |\mathcal{K}|\}$ that is owned by the supplier. Each vehicle is able to perform one route in each shipping period $t \in \mathcal{T}$. The quantity of each product available in each retail site is used to satisfy the demand of each consumption period $l \in \mathcal{T}$. The parameters and decision variables included in our formulation are the following:

Parameters

Production

- u_{mg} — production capacity of line m in production period g (time);
- a_{fm} — setup time for family f on line m (time);
- b_{pm} — setup time for product p on line m (time);
- p_{pm} — processing time of product p on line m (time);

Inventory

- $\bar{c}_i$ — capacity of the warehouse at location i;
- h_{ip} — holding cost of product p at location i;
- d_{ipl} — demand for product p at location i in consumption period l;

Routing

- v_k — capacity of vehicle k;
- t_{ij} — travel time between i and j;
- c_{ij} — travel cost between i and j;
- $[a_i, b_i]$ — service time window of location i.

Decision Variables	**Production**
A_{mfg}	equal to 1 if line m is set up to produce family f in production period g, 0 otherwise;
B_{mpg}	equal to 1 if line m is set up to produce product p in production period g, 0 otherwise;
P_{pmgt}	quantity of product p produced on line m in production period g to be delivered in shipping period t;
	Routing
X_{ijkt}	equal to 1 if arc (i,j) is traversed by vehicle k in shipping period t, 0 otherwise;
Z_{ikt}	equal to 1 if location i is visited by vehicle k in shipping period t, 0 otherwise;
W_{ikt}	arrival time of vehicle k at location i in shipping period t;
D_{ipktl}	quantity of product p to be delivered to location i by vehicle k in shipping period t to be consumed in consumption period l.

3.2. Formulation

We assume an ML policy and that quantities produced in period g are ready to be delivered in period $g+1$. Quantities received at each retail site are ready to be consumed in the period they are shipped. M is a big number. Given this, the multi-product PRP with time windows can be formulated as follows:

(MPRPTW):

$$\begin{aligned}\text{minimize} &\sum_{p\in\mathcal{P}}\sum_{m\in\mathcal{M}}\sum_{g\in\mathcal{T}}\sum_{h=0}^{g}\sum_{t=g+1}^{|\mathcal{T}|} h_{0p}\cdot P_{pmht}\\ &+\sum_{i\in\mathcal{V}'}\sum_{p\in\mathcal{P}}\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}}\sum_{h=0}^{t}\sum_{l=t+1}^{|\mathcal{T}|} h_{ip}\cdot D_{ipkhl}\\ &+\sum_{i\in\mathcal{V}}\sum_{j\in\mathcal{V}}\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}} c_{ij}\cdot X_{ijkt}\end{aligned} \tag{1}$$

s.t.

$$B_{mpg} \le A_{mf_pg} \quad \forall p\in\mathcal{P}_m, m\in\mathcal{M}, g\in\mathcal{T} \tag{2}$$

$$\sum_{t=g+1}^{|\mathcal{T}|} P_{pmgt} \le M\cdot B_{mpg} \quad \forall p\in\mathcal{P}_m, m\in\mathcal{M}, g\in\mathcal{T} \tag{3}$$

$$\sum_{f\in\mathcal{F}_m} a_{fm}\cdot A_{mfg} + \sum_{p\in\mathcal{P}_m}(b_{pm}\cdot B_{mpg} + \sum_{t=g+1}^{|\mathcal{T}|} p_{pm}\cdot P_{pmgt}) \le u_{mg} \quad \forall m\in\mathcal{M}, g\in\mathcal{T} \tag{4}$$

$$\sum_{p\in\mathcal{P}}\sum_{m\in\mathcal{M}}\sum_{h=0}^{g}\sum_{t=g+1}^{|\mathcal{T}|} P_{pmht} \le \bar{c}_0 \quad \forall g\in\mathcal{T} \tag{5}$$

$$\sum_{g=0}^{t-1}\sum_{m\in\mathcal{M}} P_{pmgt} = \sum_{i\in\mathcal{V}'}\sum_{k\in\mathcal{K}}\sum_{l=t}^{|\mathcal{T}|} D_{ipktl} \quad \forall p\in\mathcal{P}, t\in\mathcal{T} \tag{6}$$

$$\sum_{t=0}^{l}\sum_{k\in\mathcal{K}} D_{ipktl} = d_{ipl} \quad \forall i\in\mathcal{V}', p\in\mathcal{P}, l\in\mathcal{T} \tag{7}$$

$$\sum_{v \in \mathcal{K}} \sum_{p \in \mathcal{P}} \sum_{h=0}^{t} \sum_{l=t}^{|\mathcal{T}|} D_{ipkhl} \leq \bar{c}_i \quad \forall i \in \mathcal{V}', t \in \mathcal{T} \tag{8}$$

$$\sum_{i \in \mathcal{V}'} \sum_{p \in \mathcal{P}} \sum_{l=t}^{|\mathcal{T}|} D_{ipktl} \leq v_k \quad \forall k \in \mathcal{K}, t \in \mathcal{T} \tag{9}$$

$$D_{ipktl} \leq \min\{d_{ipl}, v_k\} \cdot Z_{ikt} \quad \forall i \in \mathcal{V}', p \in \mathcal{P}, k \in \mathcal{K}, t \in \mathcal{T}, l \in \mathcal{T} \tag{10}$$

$$\begin{aligned} \sum_{j \in \mathcal{V}} X_{ijkt} = Z_{ikt} \quad \forall i \in \mathcal{V}, k \in \mathcal{K}, t \in \mathcal{T} \\ \sum_{j \in \mathcal{V}} X_{jikt} = Z_{ikt} \quad \forall i \in \mathcal{V}, k \in \mathcal{K}, t \in \mathcal{T} \end{aligned} \tag{11}$$

$$\sum_{k \in \mathcal{K}} Z_{ikt} \leq 1 \quad \forall i \in \mathcal{V}, t \in \mathcal{T} \tag{12}$$

$$W_{ikt} + t_{ij} \leq W_{jkt} + M \cdot (1 - X_{ijkt}) \quad \forall i \in \mathcal{V}', j \in \mathcal{V}', k \in \mathcal{K}, t \in \mathcal{T} \tag{13}$$

$$a_i \leq W_{ikt} \leq b_i \quad \forall i \in \mathcal{V}, k \in \mathcal{K}, t \in \mathcal{T} \tag{14}$$

$$X_{ijkt}, Z_{ikt}, A_{mfg}, B_{mpg} \in \{0, 1\}; \quad P_{pmgt}, D_{ipktl}, W_{ikt} \geq 0. \tag{15}$$

The objective function (1) minimizes the holding cost at the supplier and retail sites as well as the routing cost. Constraints (2) - (5) model the lot sizing part of the problem. Constraints (2) and (3) impose that each production line can only produce if the family and product set up operations have been performed. Constraints (4) model the time capacity for each production line. Constraints (5) impose the maximum capacity for the inventory at the supplier. Constraints (6) ensure that the quantities produced of each product satisfy the deliveries made in each shipping period to each retail site, linking the production and routing decisions. Constraints (7) - (14) define the routing part of the problem. Constraints (7) ensure the demand satisfaction of each retail location. Constraints (8) are inventory management constraints to impose the maximum capacity for the inventory at each retail site. Constraints (9) are added to the formulation to impose vehicle capacities. Constraints (10) ensure that a location can only receive products if it is visited. Constraints (11) are the so-called vehicle flow conservation constraints. The degree constraints (12) impose at most one visit per location in each period. In order to define arrival times at each location for each vehicle, constraints (13) are added. Finally, these constraints also serve as subtour elimination constraints. Time windows are imposed by constraints (14). This formulation has been tested for toy instances but, when the number of locations, vehicles, and products grows, the problem becomes intractable. In fact, just enumerating the set of D_{ipktl} variables is already a very time-consuming task. For that reason, and considering the ultimate objective of solving real-world instances, the next section details a new matheuristic approach for large PRPs with realistic constraints.

4. Solution Approach

The proposed solution approach was designed by considering the dimension and complexity of the instances to be solved in a real world context. Indeed, various decomposition mechanisms are added to ensure that the problem can be solved, regardless of its size.

The first phase (***Size Reduction***) attempts to decrease the granularity of the problem to largely reduce computational times in later phases (or make the problem tractable). Reduced subproblems are derived to serve as inputs for the following procedures. Retail sites with similar geographic location and time windows are clustered into hypernodes (a hypernode is a set of retail sites to be visited in a given sequence). Afterwards, the obtained set of hypernodes is partitioned into regions (sets of hypernodes) to obtain tractable subproblems. A reduced set of routes (visiting hypernodes) is created for each region, taking advantage of a smaller number of locations (hypernodes) to reduce the complexity of the routing decisions. Products with similar production specifications and low demand are aggregated in hyperproducts (sets of products), as it is assumed that smaller priority should be given to them.

In the second phase (***Initial Solution***), the algorithm solves a set of subproblems. Each subproblem corresponds to an IRP where delivery quantities and routes are defined for each region. When a solution for the IRP subproblems is obtained, we also obtain a delivery schedule for the original problem. Afterwards, an LSP is solved to define the setup decisions and production quantities at the supplier, taking the delivery schedule as an input. This procedure builds an initial decoupled production-routing plan.

The third phase (***Improvement***) aims at improving the initial solution. Since the initial solution is obtained by partitioning the original problem, we seek to find solution improvements by relating variables describing entities that have been initially aggregated or separated in different subproblems. To do so, Mixed-Integer Program (MIP) formulations are used to improve different parts of the PRP solution.

The three phases are described in detail in the following subsections.

4.1. Size Reduction

4.1.1. Node Clustering And Region Decomposition

We propose a node clustering algorithm (Algorithm 1) to create hypernodes. The algorithm mainly focuses on geographical and time aspects. The order in which nodes are inserted into the hypernode defines the order in which each node should be served when a vehicle visits the hypernode. Hence, each hypernode has an associated path. The procedure needs two parameters: a maximum distance d_{max} and a maximum duration t_{max}. Nodes are first ordered by their earliest service time a_i and then by their latest service time b_i. From this ordered list of candidates, we take the first element to be visited, the hypernode seed, and open a new hypernode. Afterwards, a list of candidates to join the currently opened hypernode is created. A node is a candidate if, after its insertion, the following criteria are met:

(i) the travel distance of the hypernode path (following the insertion order) is less than d_{max};

(ii) the travel time of the hypernode path (following the insertion order) is less than t_{max};

(iii) the time windows of all nodes in the hypernode are respected.

From the list of candidates obtained by applying these criteria, the nearest candidate is selected. Nodes are added to the hypernode until no candidates are available. At that point, a new hypernode is opened with the first element of the remaining ordered nodes.

Algorithm 1 Node Clustering

```
procedure CLUSTERNODES(Nodes, d_max, t_max)
    hn ← Open a new hypernode
    while Nodes ≠ ∅ do
        n ← Get a candidate from the set Nodes, considering the hypernode hn, d_max, and t_max
        if n ≠ ∅ then
            hn ← Add the candidate node n to the currently opened hypernode hn
            Remove the added node n from the set Nodes
        else
            Hypernodes ← Add the hypernode h to the set Hypernodes
            hn ← Open a new hypernode
    return Hypernodes
```

Algorithm 1 defines a set of hypernodes that will be used in the initial solution phase. This set of hypernodes can still be intractable in the following phase, depending on its size. For that reason, the entire set of hypernodes can be decomposed into subsets called regions. To define the set of regions, we use a K-means algorithm [22] based on the distance between the centroids of each hypernode. Since we incorporate some routing aspects in the node clustering and region decomposition steps, the set of routes to be considered, visiting hypernodes, can only be generated after these steps. This is an advantage as the resulting set of routes of each region becomes more compact.

4.1.2. Route Generation

In order to create a new set of routes serving hypernodes, we use a simple procedure (Algorithm 2), as further route improvements can be performed later. The goal of this step is to generate a set of routes that is sufficiently rich to build good feasible solutions yet small enough to be solved using a general-purpose solver to tackle a set-partitioning formulation. The procedure uses four parameters: a set of depots *Depots*, a set of locations to be visited $Vertices$ (i.e., hypernodes), the maximum number of visits per route $maxVisits$, and the maximum duration of each route $maxDuration$. For each hypernode, the $nNearest = maxVisits - 1$ nearest neighbours are selected (line 4, Algorithm 2). Then, we create every possible combination containing the considered hypernodes. This means that we will create sets with 1 to $maxVisits$ hypernodes. Afterwards, for each depot and each set of hypernodes, we create a route using a cheapest insertion algorithm (line 12, Algorithm 2). After randomly choosing a hypernode to be the seed, the algorithm analyses each hypernode in the set and tries to insert it in the cheapest position. If it is not possible to insert the hypernode, the next hypernode is analysed. All the generated routes obey the time windows and maximum route duration constraints.

Algorithm 2 Route Creator

```
 1: procedure CREATEROUTES(Depots, Vertices, maxVisits, maxDuration)
 2:     nNearest ← maxVisits − 1
 3:     for all i ∈ Vertices do
 4:         NeighboursSet ← Create a set with vertex i and its nNearest nearest neighbours
 5:         Subsets ← Create all distinct subsets of NeighboursSet with 1 to maxVisits vertices
 6:         AllSubsets ← Save the subsets of vertices created considering vertex i
 7:     AllSubsets ← Clear repeated subsets that are found in AllSubsets
 8:     for all d ∈ Depots do
 9:         for all S ∈ AllSubsets do
10:             r ← Open a new route starting and finishing at depot d
11:             for all i ∈ S do
12:                 r ← Insert vertex i in the cheapest position considering route r (respecting time windows)
13:             if r.duration ≤ maxDuration then
14:                 RouteSet ← Add route r to the set RouteSet
15:     return RouteSet
```

4.1.3. Product Clustering

We assume that higher priority should be given to products with higher demand, which implies that these should be represented with a higher level of detail when approximations are performed. Products within the same family have similar production constraints (i.e., setup times and production lines are similar) thus aggregating them in a hyperproduct is reasonable. The product clustering procedure uses three parameters: the set of demands *Demands*, a maximum quantity of products allowed inside each hyperproduct *maxQuantity*, and the set of product families $Families$. Within each family, we aggregate low demand products in hyperproducts. The aggregated quantity of each hyperproduct should not exceed *maxQuantity* (line 9, Algorithm 3). Note that if the aggregated quantities are small, their impact in the production and transportation capacities is not considerable. We sort products by total demand and aggregate the products corresponding to the bottom of the list (usually around 80% of the products are aggregated). For each family, this clustering procedure ends up with a reduced set of products (which are not clustered because their demand is large), and a set of hyperproducts (including products with small demand). Algorithm 3 details the aforementioned product clustering procedure.

Algorithm 3 Product Clustering

```
procedure clusterProducts(Demand, maxQuantity, Families)
    for all f ∈ Families do
        D ← From the set of demands Demand get the demands for products belonging to family f
        ClassABC ← Classify each demand in D according to its quantity (A - large, B - medium, or C - small)
        ToCluster ← Get the demands to be clustered (i.e., class B or C, according to ClassABC)
        hp ← Open a new hyperproduct hp
        while ToCluster ≠ ∅ do
            p ← Get the product p with the lowest demand from the set of products ToCluster
            if hp.total_quantity + p.demand ≤ maxQuantity then
                hp ← Add product p to the currently opened hyperproduct hp
                ToCluster ← Remove product p from the set ToCluster
            else
                Hyperproducts ← Add the currently opened hyperproduct hp to the set Hyperproducts
                hp ← Open a new hyperproduct hp
    return Hyperproducts
```

The product clustering is the final step in the **Size Reduction** phase. At this point, a set of hyperproducts, and a set of generated routes visiting hypernodes inside each region are available to serve as input to the next phase.

4.2. Initial Solution

In the beginning of the second phase, the outputs provided by the first phase will serve as inputs to be used in the construction of the initial solution. These inputs comprise one IRP instance for each region (including a set of routes visisting hypernodes and a set of hyperproducts) and the data related to the LSP part of the problem. Figure 3 presents the procedure to obtain the initial solution.

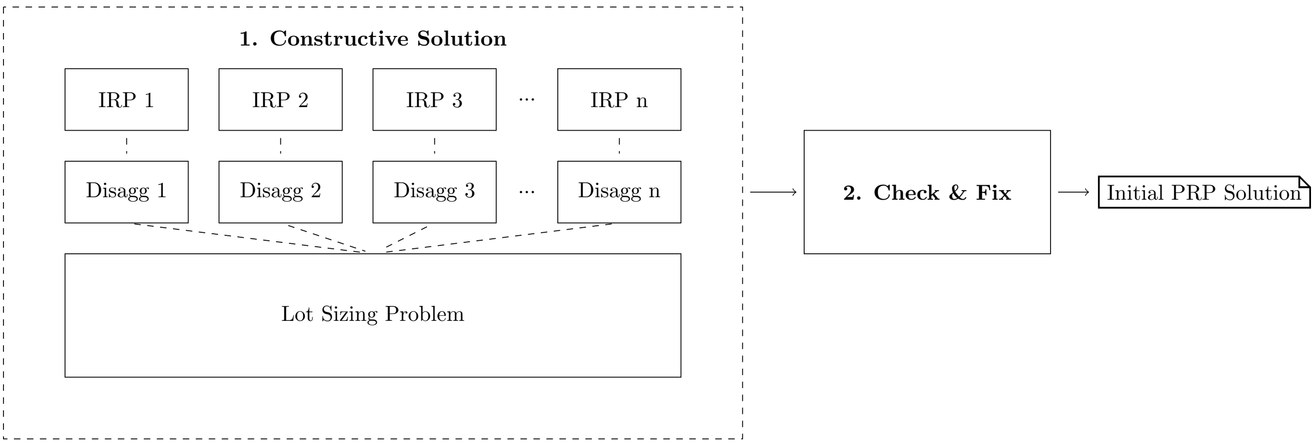


Figure 3: To build the initial solution an IRP is solved for each region. The shipping quantities demanded by each region are then disaggregated and used to solve an LSP to define production quantities. Finally, infeasibilities are fixed in the Check & Fix procedure.

In order to construct the initial solution for the PRP, the algorithm starts by solving the IRP subproblem corresponding to each region defined in the first phase. We assume infinite production capacity. Each of these IRPs is solved, using a set-partitioning formulation based on path variables, corresponding to the routes created in the first phase. The path variables are used to decide which routes of the reduced route set will be selected to make the deliveries in each period. After solving all IRP subproblems, the aggregated quantities of each hyperproduct to be delivered by each vehicle to each hypernode are known and are used to define the demand for the LSP formulation. The LSP formulation is solved by a general-purpose solver, completing the solution to the original problem. We name this step *Constructive Solution* because this initial decoupled PRP solution may be infeasible. Considering this fact, in the second step, some repair procedures are applied in order to obtain a feasible initial PRP solution. In the following subsections, we detail the steps of the second phase.

4.2.1. Constructive Solution

Regional delivery schedule definition

The algorithm starts by solving the IRP subproblem corresponding to each region. To do so, an IRP set-partitioning formulation is used. Consider the decision variables Θ_{rvt} which are equal to 1 if route r is performed by a vehicle of type v in period t, and Q_{ipvtl} which define the quantity of product p delivered to retail site i by a vehicle of type v in period t, to be consumed in period l. We define parameters o_{ir} to indicate if location i is visited by route r, c_r for the cost of performing route r, and cap_v for the capacity of a vehicle of type v. $\mathcal{R}$ is the set including only the previously generated routes and $|\mathcal{K}_v|$ is the available number of vehicles of type v. The remaining parameters are retained from the $MPRPTW$ formulation. The considered set-partitioning formulation can be defined as follows:

(SPIRP):

$$\text{minimize} \sum_{i \in \mathcal{V}'} \sum_{p \in \mathcal{P}} \sum_{v \in \mathcal{K}} \sum_{t \in \mathcal{T}} \sum_{h=0}^{t} \sum_{l=t+1}^{|\mathcal{T}|} h_{ip} \cdot D_{ipvhl} + \sum_{r \in \mathcal{R}} \sum_{v \in \mathcal{K}} \sum_{t \in \mathcal{T}} c_r \cdot \Theta_{rvt} \tag{16}$$

s.t.

$$\sum_{v \in \mathcal{K}} \sum_{t=0}^{l} Q_{ipvtl} = d_{ipl} \quad \forall i \in \mathcal{V}', p \in \mathcal{P}, l \in \mathcal{T} \tag{17}$$

$$\sum_{v \in \mathcal{K}} \sum_{p \in \mathcal{P}} \sum_{h=0}^{t} \sum_{l=t}^{|\mathcal{T}|} D_{ipvhl} \leq \bar{c}_i \quad \forall i \in \mathcal{V}', t \in \mathcal{T} \tag{18}$$

$$\sum_{v \in \mathcal{K}} \sum_{r \in \mathcal{R}} o_{ir} \cdot \Theta_{rvt} \leq 1 \quad \forall i \in \mathcal{V}', t \in \mathcal{T} \tag{19}$$

$$\sum_{r \in \mathcal{R}} \Theta_{rvt} \leq |\mathcal{K}_v| \quad \forall v \in \mathcal{K}, t \in \mathcal{T} \tag{20}$$

$$Q_{ipvtl} \leq d_{ipl} \cdot \sum_{r \in \mathcal{R}} o_{ir} \cdot \Theta_{rvt} \quad \forall i \in \mathcal{V}', p \in \mathcal{P}, v \in \mathcal{K}, t \in \mathcal{T}, l \in \mathcal{T}, t \leq l \tag{21}$$

$$\sum_{i \in \mathcal{V}'} \sum_{p \in \mathcal{P}} \sum_{l=t}^{|\mathcal{T}|} Q_{ipvtl} \leq cap_v + M \cdot (1 - \Theta_{rvt}) \quad \forall r \in \mathcal{R}, v \in \mathcal{K}, t \in \mathcal{T} \tag{22}$$

$$\begin{aligned} \Theta_{rvt} \in \{0,1\} \quad \forall r \in \mathcal{R}, v \in \mathcal{K}, t \in \mathcal{T} \\ Q_{ipvtl} \geq 0 \quad \forall i \in \mathcal{V}', p \in \mathcal{P}, v \in \mathcal{K}, t \in \mathcal{T}, l \in \mathcal{T}, t \leq l. \end{aligned} \tag{23}$$

The objective function comprises two terms. If a location i stocks a unit of product p in period t, it incurs an inventory holding cost of h_{ip}. Additionally, if route r is made by a vehicle of type v in period t, a transportation cost c_r is incurred. Constraints (17) make sure that the delivered quantities satisfy the demand. Constraints (18) ensure that the inventory capacity is respected for each retail site. Constraints (19) impose at most one visit per retail site in each period. Constraints (20) impose a maximum number of vehicles of each type to be used in each period. Constraints (21) ensure that retail sites may only receive deliveries if they are visited. Constraints (22) are vehicle capacity constraints. Note that what is called retail sites and products in this formulation, are actually hypernodes and hyperproducts that have been aggregated in the previous phase. Therefore, the solutions obtained with this formulation include aggregated deliveries (including various products) that are made to aggregated nodes (including various retail sites).

Disaggregation

Since the IRP subproblems are created using hypernodes and hyperproducts provided by the *Size Reduction* phase, it is necessary to disaggregate the delivered quantities to match the original problem. Note that at this point the inventories and delivery quantities of each individual retail site are not known. Further holding cost optimization may be achieved by assigning delivery quantities at a disaggregated level. A simple linear programming problem is solved, minimizing the holding cost of the disaggregated inventories. Here, the indices i are individual retail sites, p are individual products, $hn \in \mathcal{HV}'$ are hypernodes, and $hp \in \mathcal{HP}$ are hyperproducts. The model works with continuous variables Q_{ipvtl}, which define the quantity of product p delivered to retail site i by a vehicle of type v in period t, to be consumed in period l. The quantities $q_{hn,hp,vtl}$ are parameters provided by the solutions obtained by solving the IRPs of each region. The remaining parameters are maintained from the $MPRPTW$ formulation. The disaggregation linear programming program can be defined as follows:

(Disagg):

$$\text{minimize} \sum_{i \in \mathcal{V}'} \sum_{p \in \mathcal{P}} \sum_{v \in \mathcal{K}} \sum_{t \in \mathcal{T}} \sum_{h=0}^{t} \sum_{l=t+1}^{|\mathcal{T}|} h_{ip} \cdot Q_{ipvhl} \tag{24}$$

s.t.

$$\sum_{i \in \mathcal{V}'_{hn}} \sum_{p \in \mathcal{P}_{hp}} Q_{ipvtl} = q_{hn,hp,vtl} \quad \forall hn \in \mathcal{HV}', hp \in \mathcal{HP}, v \in \mathcal{K}, t \in \mathcal{T}, l \in \mathcal{T}, t \leq l \tag{25}$$

$$\sum_{v \in \mathcal{K}} \sum_{t=0}^{t} Q_{ipvtl} = d_{ipl} \quad \forall i \in \mathcal{V}', p \in \mathcal{P}, t \in \mathcal{T} \tag{26}$$

$$\sum_{v \in \mathcal{K}} \sum_{p \in \mathcal{P}} \sum_{h=0}^{t} \sum_{l=t}^{|\mathcal{T}|} Q_{ipvtl} \leq \bar{c}_i \quad \forall i \in \mathcal{V}', t \in \mathcal{T} \tag{27}$$

$$Q_{ipvtl} \geq 0 \quad \forall i \in \mathcal{V}', p \in \mathcal{P}, v \in \mathcal{K}, t \in \mathcal{T}, l \in \mathcal{T}, t \leq l. \tag{28}$$

The objective function minimizes the holding cost incurred by all retail sites. Constraints (25) impose that the sum of the disaggregated quantities must be equal to the aggregated quantities of each hypernode and hyperproduct. Constraints (26) make sure that the disaggregated demand of each retail site is satisfied. Constraints (27) impose the inventory capacities for each retail site. Constraints (28) are the non-negativity constraints.

Lot sizing problem

After disaggregating the quantities of each hyper entity, the demand d_{pt} of each product p, for each shipping period t to be satisfied by the supplier can be defined. In order to define how the supplier should satisfy this demand, a CLSP formulation is used (see [24] for a review). This formulation works with continuous variables P_{pmgt} to decide the quantity of product p that must be produced on production line m in production period g to be shipped in shipping period t. Additionally, the model uses the binary variables A_{mfg} equal to one if production line m is set up to produce products of family f in production period g, and binary variables B_{mpg} equal to 1 if the production line m is set up to produce product p in period g. The formulation is defined as follows:

(CLSP):

$$\text{minimize} \sum_{p \in \mathcal{P}} \sum_{m \in \mathcal{M}} \sum_{g \in \mathcal{T}} \sum_{h=0}^{g} \sum_{t=g+1}^{|\mathcal{T}|} h_{0p} \cdot P_{pmht} \tag{29}$$

s.t. (2), (3), (4), (5),

$$\sum_{m \in \mathcal{M}} \sum_{g=0}^{t-1} P_{pmgt} = d_{pt} \quad \forall p \in \mathcal{P}, t \in \mathcal{T} \tag{30}$$

$$P_{pmgt} \geq 0 \quad \forall p \in \mathcal{P}, m \in \mathcal{M}, g \in \mathcal{T}, t \in \mathcal{T}, g < t. \tag{31}$$

The objective function minimizes the total holding cost incurred by the supplier. Constraints (30) impose that the produced quantities must meet the demand $d_{pt} = \sum_{i,v,l} Q_{ipvtl}$ for each product p in each shipping period t. Note that quantities produced in a production period g only become available to be shipped in the next period. In order to define the necessary conditions to model production line capacities, setups, and the warehouse capacity of the supplier, we add constraints (2), (3), (4), and (5) from the original $MPRPTW$ formulation presented in Section 3.

4.2.2. Check & Fix

As indicated in Figure 3, after solving the LSP, a constructive solution for the PRP becomes available by joining every IRP solution of each region with the LSP solution. However, since an a priori vehicle assignment is made in the *Size Reduction* phase, it is possible that the created solution exceeds the number of available vehicles. In that case, a procedure is used in order to merge routes and lower the number of vehicles in the problematic periods. At the end of the Check & Fix step, an initial PRP solution is found.

4.3. Solution Improvement

The improvement phase of the algorithm relies on the F&O heuristic ([23], [17]). Different decomposition strategies are followed to define different subproblems, which allow for a flexible exploration of the search space. The global solution is iteratively improved by analysing the trade-offs that are typically inherent to a PRP. As opposed to most F&O approaches, the variables and constraints related to the entire formulation of our problem are never loaded. In fact, the complete PRP formulation is not used in our approach, as the large number of products and retail sites requires several minutes just to build the model. Therefore, we decided to avoid the case where a single formulation is used, contrasting with the work of other researchers (e.g., the IRP tackled by Larrain et al. [25]). Here, in each decomposition approach, a subinstance is created and solved using a decomposition-tailored formulation. Afterwards, the values of the variables are updated in the global PRP solution. The size of each subproblem is controlled, ensuring that the best possible local decisions are taken for the considered entities, while solving them to optimality. With such a matheuristic approach, it is also easier to find a compromise between solution quality and running time.

4.3.1. Improvement Matheuristic

In general terms, our matheuristic selects a decomposition strategy $\omega \in \Omega$ and explores it until the criteria to proceed to the next decomposition strategy are met. A decomposition strategy works with an associated mathematical formulation plus a subset of entities $\mathcal{ST}^{opt}$, which are used to create a subinstance, smaller than the original problem. The size of the subset of entities (and subinstance) is controlled by a parameter ω_size. For instance, in a decomposition strategy based on the routing part of the problem, a pair (k, t) may represent a route performed by vehicle k in period t. A subset of pairs $(k, t) \in \mathcal{ST}^{opt}$ can be defined to select a set of routes across several periods to be optimized. The number of routes in the subset $\mathcal{ST}^{opt}$ is defined by the parameter ω_size. The algorithm explores various decomposition strategies in the set Ω until the stopping criteria are met. All subproblems are solved using a general-purpose solver. However, if the formulation comprises subtour elimination constraints, a simple branch-and-cut procedure is called, which dynamically adds the necessary cuts for violated constraints. Figure 4 gives a schematic view of the improvement phase.

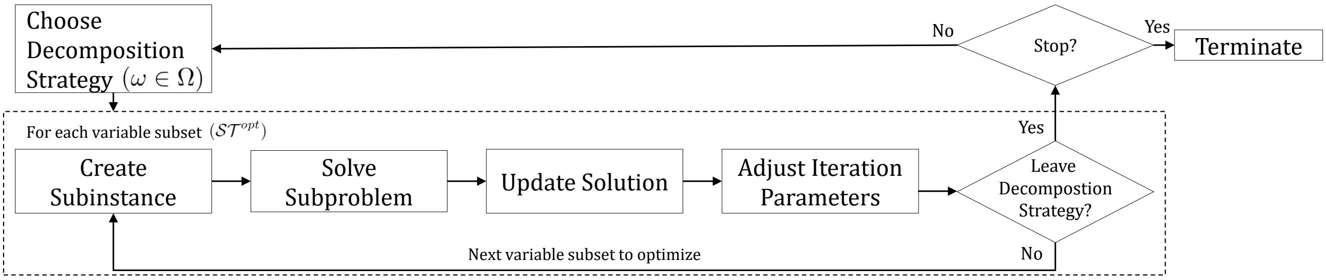


Figure 4: In the improvement phase, decomposition strategies are selected in order to explorer different decisions inherent to PRPs. In each iteration, a subset of entities is selected to create a subproblem to be solved using a decomposition-tailored formulation. The global solution, maintained outside, is updated whenever improvements are found.

After selecting a decomposition strategy ω, the idea is to analyse a small set of related entities $\mathcal{ST}^{opt}$ at a time. In each iteration, we define $\mathcal{ST}^{opt}$ with ω_size entities and create a subinstance. The subinstance is solved using the mathematical formulation associated with the current decomposition strategy. The time it takes to solve each subproblem must be within the interval $[t_{mip}^{lower}, t_{mip}^{upper}]$. If it is below the lower limit, ω_size is increased by one. If it is above the upper limit, ω_size is decreased by one. This allows the algorithm to consider larger sets of entities without maintaining the runtime of each iteration within a defined interval. After solving a subproblem, if the relative gap is excessively large, say larger than a parameter max_gap, the algorithm solves the subinstance with a period-oriented decomposition strategy (i.e., each period is solved separately when the subinstance is multi-periodic). By doing so, we achieve tractable subproblems and ensure that the runtime spent with this iteration is not completely lost. After exploring a decomposition strategy, the percentage of iterations where improvements were found is computed. If the improvement percentage is larger than $ni\%$, the decomposition strategy is explored again. If the improvement percentage is smaller than $ni\%$, the algorithm advances to the next decomposition strategy ($\omega = \omega + 1$). In the last decomposition strategy ($\omega = |\Omega|$) if the improvement percentage is smaller than $ni\%$, the algorithm terminates. Otherwise, it needs to decide if it repeats the same decomposition strategy or if it is worth to go back to the first ($\omega = 1$). If the percentage of improvement is larger than a parameter $reload\%$, it means the solution may have changed considerably and it may be worth to restart from the first decomposition strategy. If the percentage of improvement is not high but still larger than $ni\%$ (note that $ni\% \leq reload\%$), the algorithm re-explores the same decomposition strategy. The pseudo-code of the algorithm is presented in Algorithm 4 below.

Algorithm 4 Improvement Matheuristic

procedure IMPROVE($curr\_sol, max\_gap, t_{max}, t_{mip}^{lower}, t_{mip}^{upper}, ni\%, reload\%$)
  $\omega \leftarrow 1$; $best\_sol \leftarrow curr\_sol$;
  while $curr\_t \leq t_{max}$ and $\omega \leq |\Omega|$ **do**
    for each entities related to the current decomposition strategy ω **do**
      Define the subset $\mathcal{ST}^{opt}$ with ω_size surrounding entities to create a subinstance
      $curr\_sol \leftarrow$ Solve the subinstance with a general-purpose solver
      if $relative\_gap \geq max\_gap$ **then**
        $curr\_sol \leftarrow$ Solve the current subproblem with a periodic F&O procedure
      if $curr\_sol \leq best\_sol$ **then**
        $best\_sol \leftarrow curr\_sol$
      if $t_{last\_it} \notin [t_{mip}^{lower}, t_{mip}^{upper}]$ **then**
        $\omega\_size \leftarrow$ Update the parameter ω_size to adjust the runtime of the next iteration
    Choose decomposition strategy $\omega \in \Omega$ depending on $ni\%$ and $reload\%$
  return $best\_sol$

4.3.2. Decomposition Strategies

We developed three different decomposition strategies in order to achieve a flexible improvement phase, focusing on different solution attributes. We start the search with a decomposition strategy ($\omega = 1$) focused on the routing part of each period. Its base formulation is called *Daily VRP*. The second decomposition strategy ($\omega = 2$) focuses on the lot sizing part of the problem and we call its base formulation *Partial LSP*. The objective of the third decomposition strategy ($\omega = 3$) is to integrate all the decisions inherent to the PRP. Hence, we call its base formulation *Local PRP*. Table 2 summarizes the costs to be improved by each decomposition strategy as well as the entities defining the corresponding sets of entities $\mathcal{ST}^{opt}$ related to the decisions to be optimized.

Table 2: Description of the three developed decomposition strategies. Each strategy focuses on and integrates distinct sets of decisions. Only the *LocalPRP* formulation is able to improve all types of cost (for a subset of retail sites).

				Cost Improvements		
ω	Base Formulation	Entities defining $\mathcal{ST}^{opt}$	Number of subinstances	Supplier Inventory	Retail Sites Inventory	Routing
1	*Daily VRP*	$k = 1, \dots, K, t = 1, \dots, T$	$K \cdot T$			✓
2	*Partial LSP*	$i = 1, \dots, V$	V	✓	✓	
3	*Local PRP*	$i = 1, \dots, V$	V	✓	✓	✓

We detail the base formulation and the procedure to build subinstances of each decomposition strategy below.

Daily VRP ($\omega = 1$) A subinstance is defined for each combination of vehicle and period $k \in \mathcal{K}, t \in \mathcal{T}$. The ω_size routes that are nearest to route k in period t (based on their centroids in the incumbent solution) are inserted into $\mathcal{ST}^{opt}$ to define the *Daily VRP* subproblem.

The objective of the *Daily VRP* formulation (details are given in Appendix A.0.1) is exclusively to improve the routing cost. This can be interesting due to the fact that the solution that is initially constructed by the algorithm may not be optimal regarding the routing cost. There are two main reasons for this fact. First, it is likely that the route generator does not provide a set containing the optimal routes considering the nodes of a certain region. Thus, it is necessary to transform the path variables from the initial IRP set-partitioning formulation, (16) - (23), into the routing variables to be included in the *Daily VRP*, which is an arc-based formulation considering every possible arc between two locations. With these new variables, we seek to find inter-route and intra-route improvements. Second, since the problem may have been decomposed into regions, it is probable that some interesting visiting sequences (containing nodes of different regions) are still available. The *Daily VRP* model allows for an analysis of routes that were part of different regional IRPs, trying to find inter-route improvements, and mixing nodes that could not communicate in the formulation with path variables (see Figure 5).

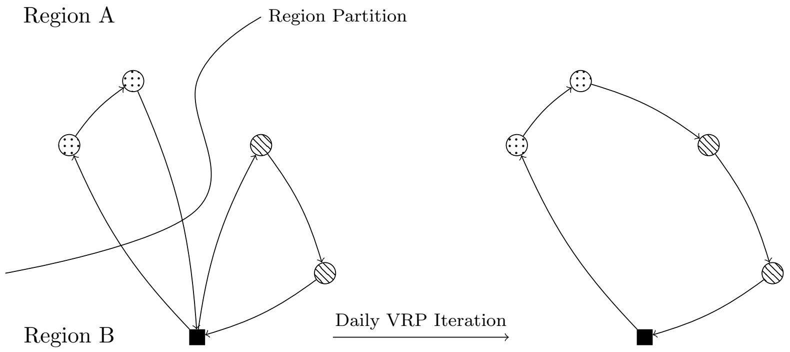


Figure 5: The path variables considered in the initial solution (considering hypernodes) phase are disaggregated into arc variables (between nodes) during the improvement phase. This allows for inter-region, inter-route, and intra-route improvements.

Partial LSP ($\omega = 2$) A subinstance is defined for each retail site $i \in \mathcal{V}'$. The ω_size nearest neighbours of i (surrounding entities) are inserted into the set $\mathcal{ST}^{opt}$. The production and delivery quantities related to the selected retail sites are optimized considering the entire planning horizon.

Considering that the initial solution is composed by decoupled decisions coming from one LSP and several IRP subproblems, further improvements regarding production and inventory management variables may still be found. Furthermore, during the improvement phase, most solution changes are focused on small subproblems (particularly when exploring decomposition strategies with routing decisions, which are more complex) and it is necessary to integrate supplier and retail site decisions at a larger scale. The *Partial LSP* base formulation (details are given in Appendix A.0.2) aims at improving the production setup variables, the production quantities, and the delivery quantities.

Since routing variables are not loaded into this model, larger sets of retail sites can be analysed simultaneously. Nevertheless, the size and complexity of the model can be adjusted, considering the computing time of each iteration. If the model takes too long to solve, each iteration may be simplified either by fixing some production setups or by loading smaller sets of retail sites. Figure 6 gives a schematic overview of the possible decisions to be made with this formulation, showing the entities and variables to be loaded in each time period.

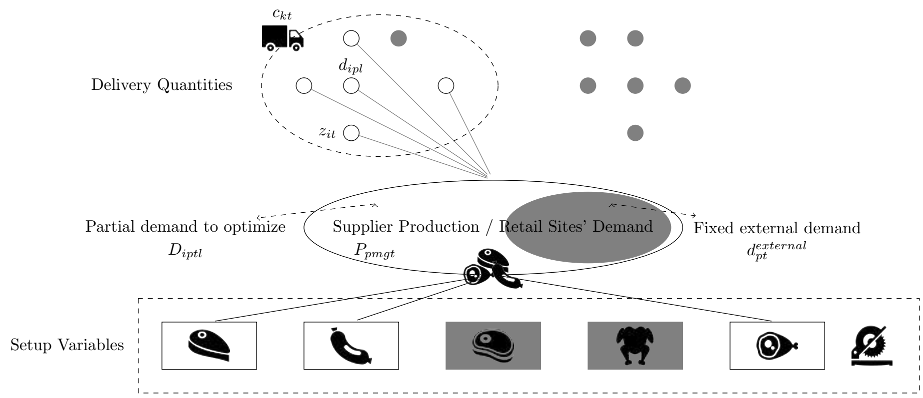


Figure 6: In the Partial LSP, the production and delivery quantities of a subset of retail sites are optimized. All routing decisions are maintained. Only some production flows are allowed (white setup variables, in the bottom) and some shipping flows are allowed (white nodes visited by the vehicle, on the top).

In Figure 6, the entities in white are the ones that can be optimized. This means that only some setup, production, and delivery quantity decisions can be improved. The entities represented in gray are fixed and are not to be loaded into the model. The production line shown at the bottom, can produce (or provide flow for) three products. The retail sites that were loaded into the *Partial LSP* (in white) have some scheduled visits fixed in the current solution. These visits are represented by a parameter z_{it}, which is equal to one if the retail site i is visited in period t in the current solution. If the retail site is visited, the delivered quantities of each product p to satisfy its demand in period l, D_{iptl} can be optimized and the supplier can also adapt the production quantities to better satisfy each retail site included in the model (white part of the ellipse). This means that the supplier can decide not only its own inventories but also the inventories to be held at each retail site in the subinstance. The retail sites that were not loaded into the model (in gray) demand a fixed quantity, $d_{pt}^{external}$, for each product p, in each period t (gray ellipse) and their deliveries must remain as they are in the incumbent solution. Therefore, the supplier can only decide if these quantities are to be made-to-stock or made-to-order, which means that the inventories of the unloaded retail sites are not changed. Note that in the incumbent solution, different vehicles visit different sets of locations (in Figure 6, a set is represented by the dashed ellipse). Therefore, it is necessary to keep track of the deliveries of each day to make sure that the capacity of the vehicles c_{kt} and the maximum inventory levels $\bar{c}_i$ of the retail sites continue to be respected.

Local PRP $(\omega = 3)$ A subinstance is defined for each retail site $i \in \mathcal{V}'$. The ω_size nearest neighbours of i (surrounding entities) are inserted into the set $\mathcal{ST}^{opt}$. These retail sites are considered through the whole planning horizon. For each period t we define the set of vehicles $\mathcal{K}_t$ visiting any of the retail sites belonging to $\mathcal{ST}^{opt}$ (in the incumbent solution). Additionally, all the locations visited by the vehicles in $\mathcal{K}_t$ and not in $\mathcal{ST}^{opt}$ are inserted into $\mathcal{V}_t^{out}$, for each period t. The set of retail sites $\mathcal{V}_t = \mathcal{ST}^{opt} \cup \mathcal{V}_t^{out}$ and vehicles in each subinstance is different in each time period t.

The models presented in the two previous subsections either deal with routing or holding costs separately. Although they can be called iteratively, their solutions will never be able to trade-off holding costs against routing costs. For this reason, a third model is developed. The objective of the *Local PRP* base formulation (details are given in Appendix A.0.3) is to enable the solution approach to perform the aforementioned trade-off analysis, allowing for fully integrated decisions considering production, inventory management and routing aspects (at least locally). Figure 7 provides a schematic representation of the variables to be loaded in each *Local PRP*.

In this model, the set of retail sites $\mathcal{ST}^{opt}$ (white nodes) is considered through the whole planning

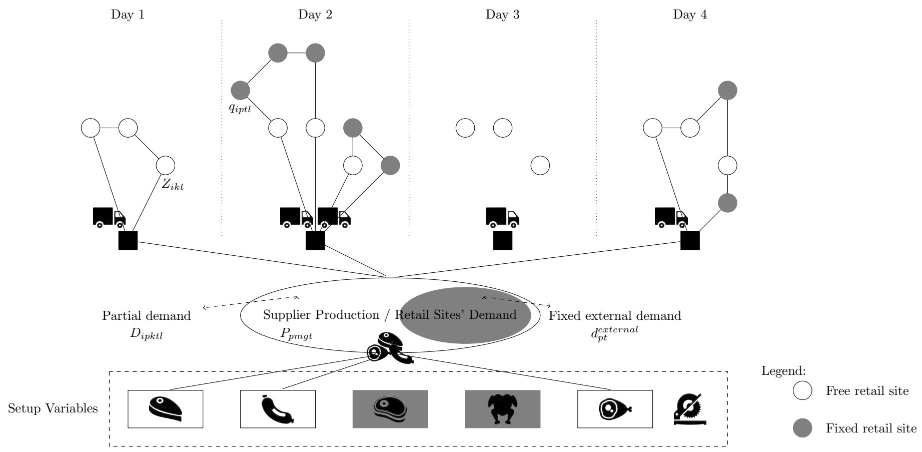


Figure 7: Example showing a Local PRP instance for 3 selected retail sites in the first period

horizon of the global PRP. In the incumbent solution, these retail sites are visited by a set of vehicles which also visit other retail sites which do not belong to $\mathcal{ST}^{opt}$ (gray nodes in periods 2 and 4). These additional retail sites will be different in each period and are represented by the sets $\mathcal{V}_t^{out}$. The deliveries to these external retail sites have to be maintained, therefore they will be fixed in the model. The only decision that can actually change is the vehicle that will perform these visits. In some periods, retail sites of the set $\mathcal{ST}^{opt}$ may not be visited in the incumbent solution (as in period 3). However, they are still loaded into the model in conjunction with vehicles that are not used in that period or with vehicles that are passing nearby. This enables the algorithm to eliminate and start routes. The idea is to create subPRPs allowing for integrated decisions considering only the retail sites of the set $\mathcal{ST}^{opt}$. The LSP part (represented in the lower part of Figure 7) is completely loaded, thus deliveries that are internal and external to the subinstance are both considered in the production plan. Note that it is necessary to ensure that these PRPs are tractable and can be loaded quickly. In order to adjust the running time of these small PRPs, one can fix setup decisions and some routing decisions such as arcs or visiting variables if necessary.

5. Computational Experiments

Our solution approach was developed to solve large multi-product PRPs with time windows including complex production activities. However, the literature does not provide instances considering such complexity. For this reason, we compare our algorithm using instances that are proposed both in the IRP and the PRP literature.

The conditions for all the experiments performed by our matheursitc ($F\&O$) are the following:

- Runs of 3600 seconds;
- CPLEX 12.6.1 is used to solve subproblems;
- Intel Core processors running at @ 2.4 GHz;
- A single thread is used during the improvement phase;
- Regions are created using a k-means algorithm based on centroid distances (regions should have less than 50 nodes to be handled by the general-purpose solver);
- Initial route sets ate created using a nearest neighbour heuristic.

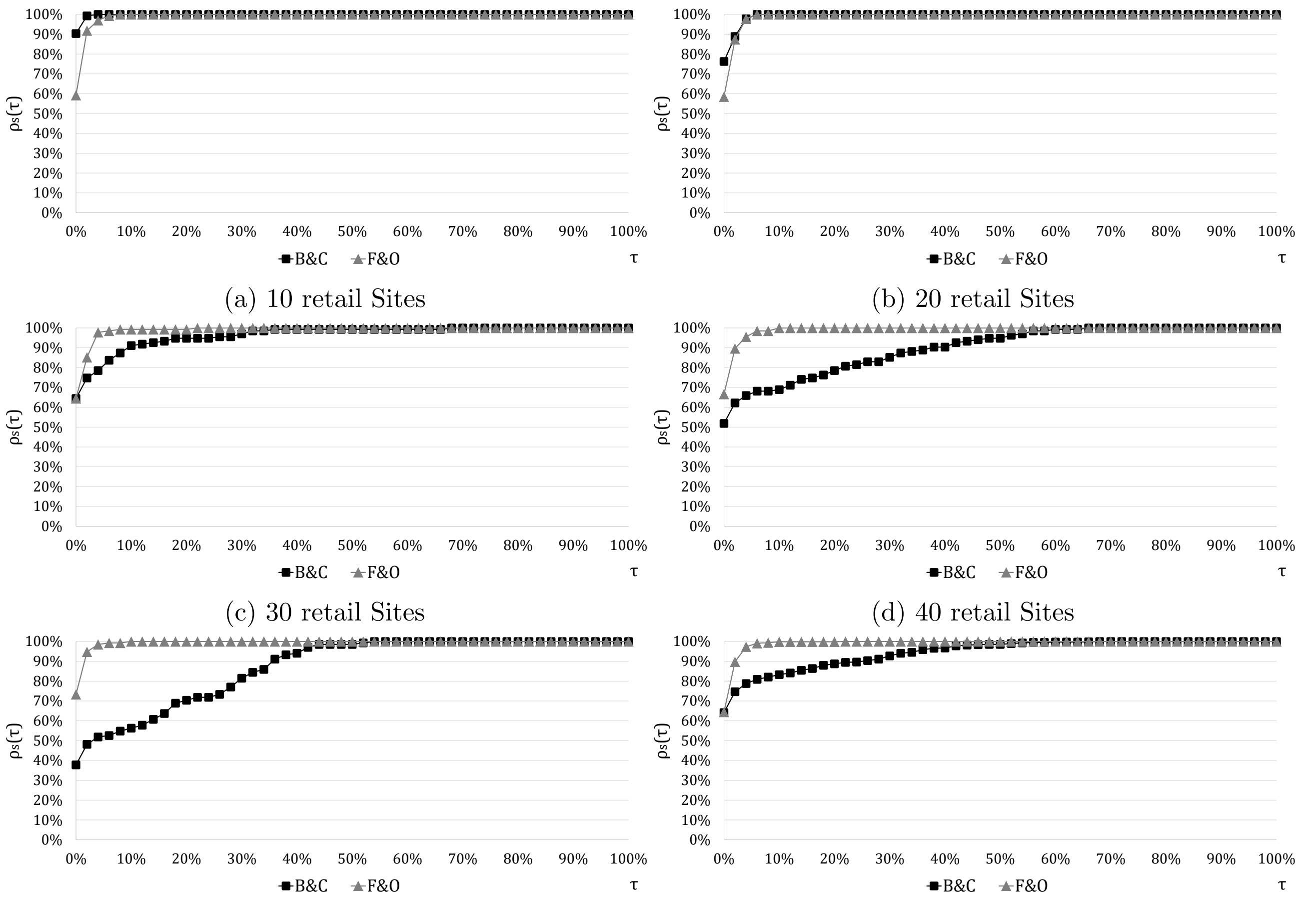
The comparison of our matheuristic with other approaches is based on the ideas presented by Dolan and Moré [20]. The authors present the concept of performance profile, a chart depicting the cumulative probability for each algorithm to obtain a solution with a relative gap smaller than or equal to τ. The relative gap is computed in relation to the best known solution for each instance p. For each approach s the relative gap $r_{p,s}$ is computed. The performance probability $\rho_s(\tau)$ is then defined, for each τ, according to expression

$$\rho_s(\tau) = \frac{1}{n_p}\text{size}\{p \in P : r_{p,s} \leq \tau\},$$

where n_p is the total number of instances and τ is the threshold for the relative gap based on the best solution found among all methods.

Coelho and Laporte, MMIRP instances

In order to compare the performance of our matheuristic we consider the most similar instances available in the literature, which consider a multi-product and multi-vehicle setting. These instances were proposed by Coelho and Laporte [18] in a work related to the IRP and can be accessed in the first author's website (`www.leandro-coelho.com/instances`). Despite disregarding production decisions, we consider that the IRP is closely related to the PRP and this is still an interesting comparison. We compare our *F&O* approach against the *B&C* approach proposed by Coelho and Laporte [18] using the performance profiles presented in Figure 8.



(a) 10 retail Sites
(b) 20 retail Sites
(c) 30 retail Sites
(d) 40 retail Sites
(e) 50 retail Sites
(f) All instances

Figure 8: Performance profiles related to Coelho and Laporte [18] solutions, MMIRP instances

We conclude that for the smaller instances with 10 and 20 retail sites (Figures 8a and 8b), Coelho and Laporte's exact approach presents a larger probability to find best known solutions (for $\tau = 0$). For instances with 30, 40, and 50 (Figures 8c, 8d, and 8e) retail sites our approach outperforms Coelho and Laporte's B&C as it is able to find a larger number of best known solutions (i.e., τ is closer to 0) and presents a larger probability to find solutions with smaller gaps to the best known solution.

In the performance profile including all the instances (Figure 8f) (excluding instances with 100 retail sites that are not solved by Coelho and Laporte's exact approach) we conclude that our F&O never finds solutions with a gap that is larger than 10% whereas Coelho and Laporte's B&C obtains some solutions with a gap larger than 50% for instances with 40 and 50 retail sites. Note that despite being an exact method, Coelho and Laporte's B&C experiments were run for 42 000 seconds. Therefore, we conclude that with a smaller computing time, our matheuristic has a larger probability of finding best known solutions.

Additionally, we present the average improvement in the best known solutions in Table 3.

Table 3: Average upper bound improvement, F&O vs B&C, MMIRP instances (Coelho and Laporte [18])

	#Retail Sites							
Instances	10	20	30	40	50	100	Avg Imp	Avg Time (s)
mmirp-1-1-3	0.00%	0.00%	0.00%	0.00%	0.00%	-	0.00%	122.84
mmirp-3-1-3	0.00%	-0.05%	0.00%	0.00%	-0.11%	-	-0.03%	97.64
mmirp-5-1-3	-0.12%	0.00%	-0.02%	-0.02%	-0.03%	-	-0.04%	104.64
mmirp-1-3-3	0.00%	0.00%	-0.05%	-0.62%	-1.21%	-	-0.38%	1120.92
mmirp-3-3-3	-0.35%	-0.34%	**1.72%**	**8.15%**	**27.78%**	-	**7.39%**	1998.60
mmirp-5-3-3	0.00%	0.00%	-0.11%	-0.78%	-0.84%	-	-0.35%	1840.36
mmirp-1-5-3	-0.34%	-1.35%	**1.65%**	-1.05%	**12.61%**	-	**2.30%**	2202.00
mmirp-3-5-3	0.00%	-2.07%	-2.27%	-1.68%	**2.25%**	-	-0.75%	2703.84
mmirp-5-5-3	0.00%	-0.30%	-0.52%	-1.50%	**6.10%**	-	**0.75%**	2686.56
mmirp-1-1-5	0.00%	-0.13%	-0.38%	-0.41%	**0.13%**	-	-0.16%	371.80
mmirp-3-1-5	-0.24%	-0.43%	-0.45%	-0.71%	-0.65%	-	-0.50%	628.60
mmirp-5-1-5	-0.59%	-0.11%	0.00%	-0.25%	**0.12%**	-	-0.17%	509.16
mmirp-1-3-5	-0.39%	-0.01%	**2.74%**	**12.53%**	**21.26%**	-	**7.23%**	2753.96
mmirp-3-3-5	-0.33%	**0.66%**	-2.49%	-1.01%	**25.45%**	-	**4.46%**	3088.20
mmirp-5-3-5	0.00%	0.00%	0.00%	**0.65%**	**10.35%**	-	**2.20%**	3232.76
mmirp-1-5-5	-1.26%	-0.07%	**4.00%**	**20.99%**	**33.18%**	-	**11.37%**	3165.08
mmirp-3-5-5	-1.94%	-1.64%	**4.98%**	**26.02%**	**14.00%**	-	**8.28%**	3583.56
mmirp-5-5-5	-0.74%	**2.20%**	**10.93%**	**35.82%**	**34.30%**	-	**16.50%**	3600.00
mmirp-1-1-7	-0.36%	-1.90%	-1.06%	-1.07%	-0.64%	-	-1.01%	749.00
mmirp-3-1-7	-0.59%	-1.09%	-0.18%	**0.48%**	**0.71%**	-	-0.14%	1597.40
mmirp-5-1-7	-0.02%	-0.67%	-0.58%	-0.12%	**2.09%**	-	**0.14%**	1746.04
mmirp-1-3-7	-2.52%	-0.76%	**0.34%**	**29.10%**	**26.71%**	-	**10.58%**	3087.48
mmirp-3-3-7	-0.36%	**0.91%**	-1.17%	**6.69%**	**25.40%**	-	**6.29%**	3530.16
mmirp-5-3-7	-0.48%	-0.19%	**0.77%**	**41.73%**	**34.08%**	-	**15.18%**	3600.00
mmirp-1-5-7	-0.23%	-1.22%	**15.60%**	**31.20%**	**29.22%**	-	**14.92%**	3480.72
mmirp-3-5-7	-1.10%	**2.10%**	**30.59%**	**27.65%**	**14.33%**	-	**14.71%**	3600.00
mmirp-5-5-7	**0.30%**	**0.80%**	**4.75%**	**18.84%**	**14.15%**	-	**7.77%**	3600.00
Avg	-0.43%	-0.21%	**2.55%**	**9.28%**	**12.25%**	-	**4.69%**	2177.83
	-0.32%			**8.03%**				

The results suggest that our matheuristic provides better results for instances with larger size. While for smaller instances (#Retail Sites ≤ 20) we provide solutions with an average deviation of -0.32% to the best known solution, for larger instances (#Retail Sites ≥ 30) we find an average improvement of 8.03%. We provide 377 new best solutions, which account for 46% of the entire instance set. This includes 135 instances (the ones with 100 retail sites) which had no solution available in the literature. The average runtime of these tests was 2177.83 seconds.

Region decomposition analysis *F&O*, MMIRP instances with 100 retail sites

Since Coelho and Laporte [18] did not report any result for instances with 100 retail sites, we consider that these instances are more challenging. Therefore we tested our matheuristic for the cases where the initial solution is created without regional decomposition against the cases where the problem is decomposed into 2 to 4 regions. The results are presented in Table 4.

Regional decomposition is particularly useful when the set-partitioning formulation is not able to provide an initial solution, preventing the approach from solving an instance. When the initial solution has a large relative gap, it is also a sign that regional decomposition may be beneficial. Furthermore, cases where the initial solution takes too much time to be found may also be addressed with regional decomposition, as the time available for the improvement phase becomes short. These are the main reasons for the results presented in the *No Decomposition* columns of Table 4. When no regional decomposition is applied, 11 of the 135 instances with 100 retail sites are not solved. When regional decomposition is applied (from 2 to 4 regions) all the instances are solved, though we only compare the 124 instances that are solved in all cases. The quality of the initial solution deteriorates as the number of regions increases. However, the time to obtain an initial solution with more regions is usually shorter, providing more time for the improvement phase. The additional time for improving the solution proves beneficial as the average objective value of the final solution is smaller. Figure 9 shows that, on average, the algorithm spends 22.36% (size of the bubble) of the time to compute the initial solution when no decomposition is performed, whereas the decomposed instances only need around 7% of the total time to obtain the initial solution. The average runtime to achieve the best solution is smaller with 2 and 3 regions.

Table 4: Average objective value comparison of regional decompositions, MMIRP instances with 100 retail sites

			Initial solution objective (Instances solved)				Final solution objective			
Periods	Vehicles	Products	No Decomposition	2 Regions	3 Regions	4 Regions	No Decomposition	2 Regions	3 Regions	4 Regions
3	1	1	12970.84 (5)	10972.42 (5)	10815.78 (5)	10738.82 (5)	**10563.04**	**10563.04**	**10563.04**	**10563.04**
3	1	3	20683.38 (5)	18370.70 (5)	18344.86 (5)	18312.24 (5)	**18188.98**	**18188.98**	**18188.98**	**18188.98**
3	1	5	27919.62 (5)	25811.02 (5)	25697.92 (5)	25659.78 (5)	25522.64	25522.10	25522.88	**25521.78**
3	3	1	13768.00 (5)	14414.96 (5)	15527.16 (5)	15344.48 (5)	**10866.48**	10885.70	10904.86	11126.60
3	3	3	21694.44 (5)	21891.26 (5)	22773.44 (5)	24055.70 (5)	**19132.68**	19253.38	19784.70	19564.20
3	3	5	29223.30 (5)	28966.26 (5)	30041.06 (5)	30641.42 (5)	26609.82	**26168.16**	26484.06	26505.40
3	5	1	15608.24 (5)	14540.94 (5)	15028.74 (5)	16235.54 (5)	11690.06	**11388.92**	11701.62	11701.70
3	5	3	23216.08 (5)	22007.88 (5)	22742.08 (5)	23638.68 (5)	19658.44	**19576.88**	19797.52	19795.58
3	5	5	31274.80 (5)	29756.44 (5)	29909.64 (5)	30417.04 (5)	27823.64	**26531.16**	26646.34	26813.86
5	1	1	19032.84 (5)	17102.76 (5)	16271.80 (5)	17061.68 (5)	15605.06	15742.84	**15589.42**	15665.28
5	1	3	30639.06 (5)	30931.04 (5)	30313.82 (5)	32474.90 (5)	**27972.18**	28402.06	29097.38	28702.44
5	1	5	43436.58 (5)	47693.66 (5)	47069.48 (5)	56131.92 (5)	38935.94	**38893.60**	39035.06	39009.36
5	3	1	26565.96 (5)	23281.58 (5)	25061.92 (5)	23241.24 (5)	19381.26	18982.96	**18338.22**	18739.22
5	3	3	38675.46 (5)	46211.84 (5)	39113.20 (5)	41295.10 (5)	33138.68	38224.50	**32116.44**	32396.98
5	3	5	50038.72 (5)	64979.92 (5)	54824.44 (5)	51498.48 (5)	45407.74	47205.90	**42845.76**	43540.44
5	5	1	32585.06 (5)	27607.10 (5)	27603.22 (5)	28324.92 (5)	22923.06	**20921.00**	20941.68	20945.12
5	5	3	48592.72 (5)	40485.50 (5)	39867.44 (5)	42587.80 (5)	38294.22	34089.00	**33725.78**	34644.96
5	5	5	62724.86 (5)	49403.10 (5)	49219.82 (5)	52328.88 (5)	51161.62	44429.22	**43874.78**	45457.62
7	1	1	26812.10 (5)	38001.98 (5)	53033.32 (5)	43965.76 (5)	22176.50	**21780.94**	22053.54	22367.70
7	1	3	45248.04 (5)	82047.56 (5)	69747.78 (5)	67159.28 (5)	**40630.30**	41712.42	41442.56	40822.90
7	1	5	60317.36 (5)	114063.60 (5)	112091.00 (5)	106659.10 (5)	**55080.28**	56279.94	55608.98	55482.66
7	3	1	37075.93 (4)	44608.50 (4)	41849.78 (4)	40040.78 (4)	26896.30	**25041.78**	26196.00	26291.50
7	3	3	68574.28 (4)	79927.10 (4)	86745.18 (4)	73519.25 (4)	56826.73	**52481.08**	61993.08	52529.73
7	3	5	73996.94 (5)	80079.32 (5)	97818.48 (5)	96876.50 (5)	64271.64	**58845.36**	66494.46	67406.94
7	5	1	45519.00 (3)	41347.47 (3)	40239.80 (3)	45359.13 (3)	39297.50	37337.17	**30288.57**	33779.03
7	5	3	77951.60 (1)	62796.30 (1)	58904.20 (1)	58669.60 (1)	60807.30	52508.80	**51595.90**	52215.50
7	5	5	105180.40 (2)	71640.10 (2)	67168.50 (2)	91626.25 (2)	79515.55	63972.80	**62483.30**	66300.00
Average (Count)			37278.97 (124)	40898.11 (124)	41071.95 (124)	41173.27 (124)	31433.83	**30407.24**	30435.28	30453.64
						Average Time To Best (s)	2363.09	2115.98	**2102.51**	2275.78

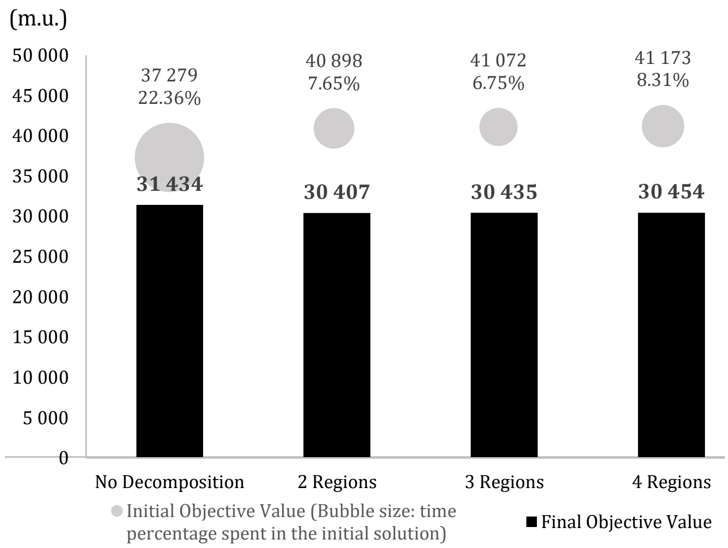


Figure 9: The initial objective value is better if fewer regions are created. However, larger computational times are required to obtain an initial solution. When the original problem is decomposed into more regions, the initial solution is obtained faster. The additional time to use in the improvement phase allows for better final objective values.

One important aspect that needs to be addressed when solving these instances is the feasibility of the fleet. The *Check & Fix* procedure is usually called when the deliveries scheduled for each region cannot be performed by the available fleet. Therefore, this procedure is always called when the number of regions is larger than the number of vehicles (since some vehicles will be repeated in some regions). In larger instances the *Check & Fix* procedure may take more time to repair these infeasibilities as seen in the initial solutions of the instances with 7 periods and 1 vehicle.

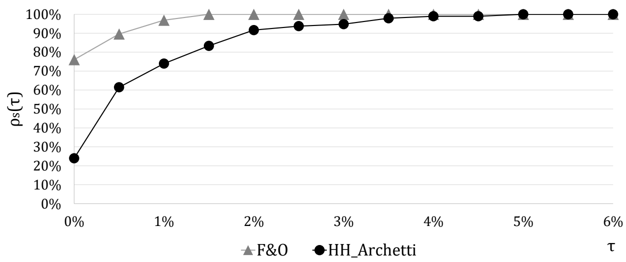
Archetti et al., MPRP instances

Finally, we compare the performance of our algorithm with PRP instances considering a single product and multiple vehicles, presented by Archetti et al. [6] and available on the authors' website (http://or-brescia.unibs.it/instances). This algorithm is tailored for a particular PRP with constant demands and an uncapacitated production facility. Some changes were made in the models used in our matheuristic in order to match the same assumptions (e.g.: the authors consider that production becomes available to be shipped as soon as it is produced). Archetti et al. [6] consider four classes of instances with 19, 50 and 100 retail sites. Since the authors allow for an unlimited fleet, we adapted the instances and considered the maximum number of vehicles that is used in their tests for each number of retail sites. Table 5 shows the number of vehicles that are used in our tests, the average deviation between our solutions and those of Archetti et al. [6] for the four instance classes, and average runtimes.

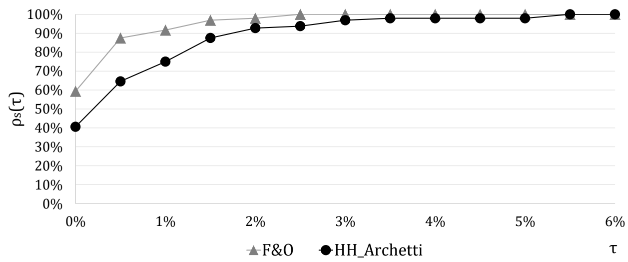
Table 5: Average deviation from Archetti et al. [6] solutions and runtime

Retail Sites (Vehicles)	I	II	III	IV	Dev	Time (s)
19 (1)	-0.83%	-0.12%	-1.15%	-0.24%	**-0.58%**	**241.22**
50 (7)	-0.38%	-0.10%	-0.65%	-0.53%	**-0.41%**	**1733.89**
100 (14)	0.25%	0.05%	1.35%	1.07%	**0.68%**	**3599.20**
Dev	**-0.32%**	**-0.05%**	**-0.15%**	**0.10%**	**-0.11%**	**1858.10**

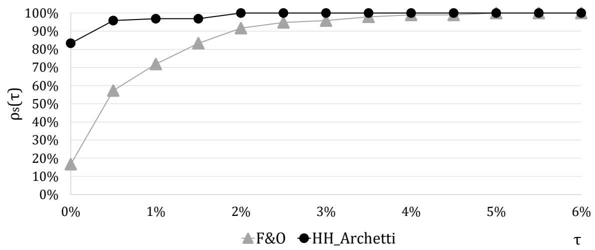
The results suggest an improvement in the average objective function of the instances with 19 and 50 retail sites. For instances with 100 retail sites, the objective value is 0.68% worse on average. Considering the total set of instances, we improve the average difference over the best solution by 0.11%. Note that the problem considered in [6] is slightly different from our context. However, the results show that our matheuristic is also competitive on this instance set. Figure 10 shows the performance profiles for each set of instances. For instances with 19 and 50 retail sites, our matheuristic finds a larger number of best solutions. For instances with 100 retail sites, the approach of Archetti et al. [6] shows a larger probability to find solutions with a deviation smaller than or equal to 5% from the best known solution. The average runtime of these tests was 1861.30 seconds.



(a) 19 retail sites



(b) 50 retail sites



(c) 100 retail sites

Figure 10: Performance profiles related to Archetti et al. [6] solutions, MPRP instances

Considering the afore presented perspectives, our matheuristic is considered to be competitive in terms of solution quality. Indeed, the flexibility provided by our approach is one of its greatest strengths as matheuristics considering problem extensions are still scarce in the IRP and PRP literature.

6. Case Study

6.1. Analysis of the Current Situation

This section details the case study of a European meat store chain which owns a Meat Processing Center (MPC) where several meat products are processed and delivered to several meat stores. The characteristics of the problem include 13 productions lines, 175 perishable products, 185 meat stores with delivery time windows and 35 heterogeneous vehicles. With the objective of detailing the challenge faced by the company, a simplified overview of the problem is presented in Figure 11 and explained below (the numbers in the text correspond to the numbers in the figure).

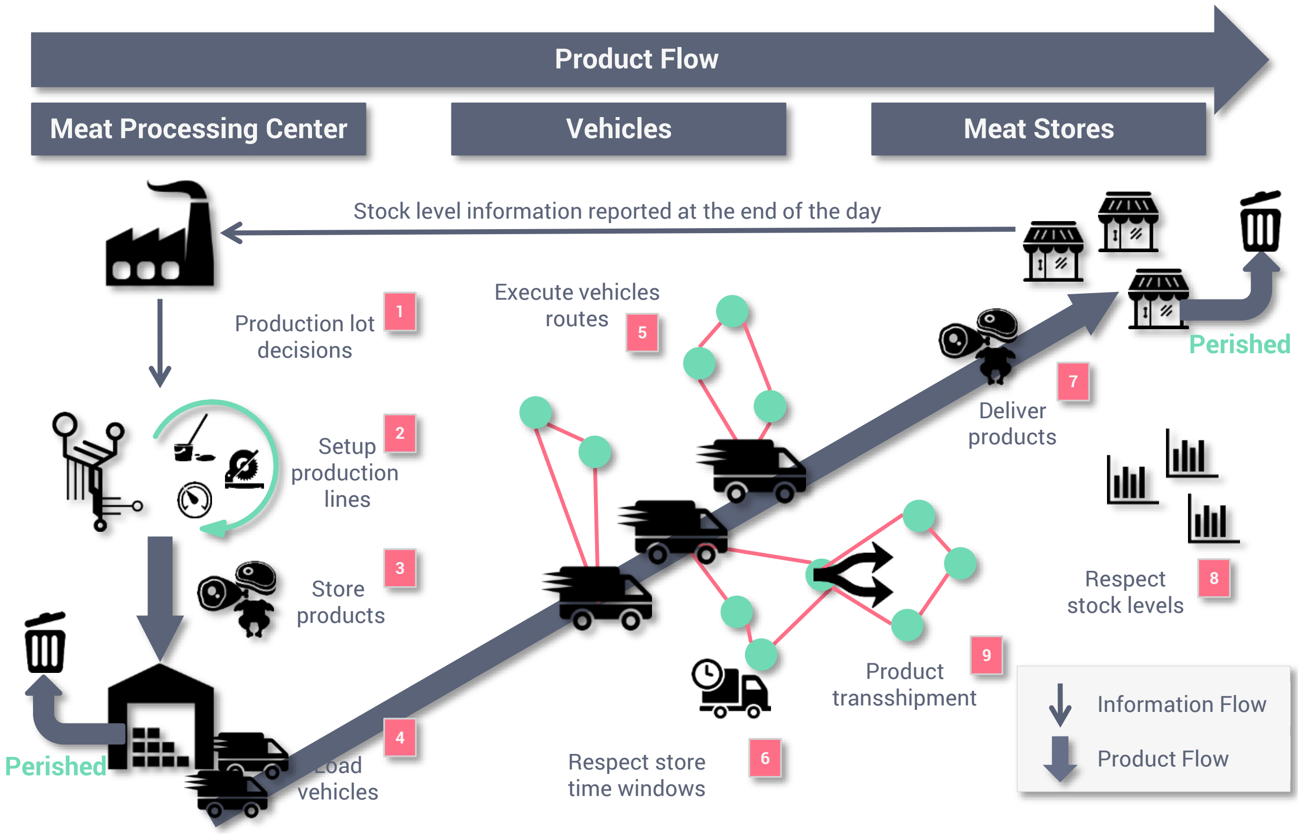


Figure 11: Representation of the realistic multi-product PRP with time windows

The company manages a single MPC where several cutting lines, with particular specifications, are able to make specific cuts to produce different sets of product families. Each family comprises a set of products which are packaged in different sizes. The MPC is the entity which decides the production schedule (1) by applying a VMI policy. In order to produce a certain family, a major setup operation (usually around 1 hour) must be performed on the cutting line. Additionally, in order to set up a cutting line to produce a certain meat product within a product family, a minor setup (usually around 15 minutes) must also be performed. These setup times are usually necessary to clean the machines, to set cutting speeds and thickness, as well as to set the labelling and packaging processes (2). Each time a family is set up on a cutting line, a minimum lot needs to be produced. The MPC works during two work shifts of 8 hours, with a break of 1 hour between shifts. Processed meat products may be directly loaded into vehicles to be delivered or they can be stored at the MPC's refrigerated warehouse to be delivered in a later period. Most of the meat products have a shelf life of around 7 days. Different holding costs per product are incurred when the refrigerated warehouse of the MPC is used (3).

When the products are to be delivered to the meat stores, the fleet is loaded (4), respecting the capacity of each vehicle. Each driver performs a different vehicle route in order to deliver the necessary products and quantities to satisfy the demands of each meat store (5). Note that in this realistic context, each meat store may only receive deliveries during a certain time window (6). The delivery sizes (7) are defined by the MPC, and have to respect the capacity of the warehouse of each meat store (8). Each store can also stock products incurring a holding cost that is different per store and product. The holding costs are incurred on a daily basis.

Since all the vehicle routes need to start from and finish at the MPC (supplier and depot), it is impossible to perform all the deliveries in the north region, given that the legislation prohibits drivers to work more than 9 hours in a day. Therefore, the MPC also needs to make deliveries to a Transshipment Facility (TF) located in the north of the country. A larger vehicle needs to deliver the products to be distributed by the fleet that is based at the TF (9). This transshipment operation also needs to be performed during a certain time window.

Finally, in the meat stores, the customer demands need to be satisfied on a daily basis. Since we are dealing with food, meat products may perish either at the warehouse of the MPC or at the warehouses of the meat stores.

The current planning process of the company considers all these decisions in a decoupled approach. This means that the events that are triggered by changes both in the information and product flows are not received and interpreted in the same manner by each of the comprised entities (MPC and meat stores). This leads us to four main reasons to investigate an integrated approach:

Capacity issues Currently, each store makes orders without having the information about the MPC capacity. Usually, meat stores order more than they need as they are aware that sometimes the MPC is not able to fulfil all the demand (a clear contributor to the bullwhip effect).

Lack of visibility The MPC only has visibility of the demand of the stores for 1 or 2 periods ahead. This fact is clearly hampering the planning activity of the MPC, as it ends up producing almost every product every day. Therefore, a significant part of the available time capacity is spent setting up the cutting lines to produce small lots of each product.

Improve cost One of the objectives of this research is to quantify the savings obtained by an integrated approach, compared to the decoupled approach where each store is making orders individually. As proven by numerous papers in the past (considering simpler versions of the PRP), we expect to improve the global activity cost.

Deal with complexity Since the planning process currently is built manually by experienced planners, it is important to devise a systematic approach. Although the plans obtained by the planners may already present good quality, the process is still very time consuming and non-scientific. Additionally, when unexpected events occur, it may be quite difficult for the planners to react in a timely manner, thus having an algorithm to aid the planning process is valuable.

6.2. Methodology Application

In order to apply the developed solution approach to the case study, some additional constraints need to be taken into account. In fact, these constraints are not common in the context of the PRP literature and need to be adapted to be included in our approach. Table 6 presents the additional constraints as well as the specific parts of our approach that have to be adapted.

Table 6: Case study additional constraints

Label	Act.	Constraint / Description	Mathematical form	Adaptation
(a)	Pro	Minimum Family Lot Size	$\sum_{p\in P,t\in T} P_{pmgt} \geq minLot_{fm} \quad \forall f \in F, m \in M, g \in T$	IS, PLSP, LPRP
(b)	Pro	Retail Site Perishability	$X_{ipktl} = 0 \quad \forall i \in V, p \in P, k \in K, t \in T, l \in T, t + maxConsumptionDays < l$	IS, PLSP, LPRP
(c)	Pro	Supplier Perishability	$P_{pmgt} = 0 \quad \forall p \in P, m \in M, g \in T, t \in T, g + maxShippingDays < t$	IS, PLSP, LPRP
(d)	Pro	Compatible Product/Line	$B_{mpt} \leq comp_{mp} \quad \forall m \in M, p \in P, t \in T$	IS, PLSP, LPRP
(e)	Pro	Time Capacitated Lines	$\sum_f fst_{fm} \cdot A_{mft} + \sum_p pst_{pm} \cdot B_{mpt} + \sum_p pt_{pm} \cdot P_{pmgt} \leq cap_{mt} \quad \forall m \in M, t \in T$	IS, PLSP, LPRP
(f)	Inv	Safety Stocks	$I_{ipt} \geq ss_{ip} \quad \forall i \in V, p \in P, t \in T$	IS, PLSP, LPRP
(g)	Rou	Delivery Time Windows	$a_i \leq W_{it} \leq b_i \quad \forall i \in V, t \in T$	DVRP, LPRP
(h)	Rou	Maximum Route Visits	$\sum_{i\in V} Z_{ikt} \leq maxVis \quad \forall k \in K, t \in T$	DVRP, LPRP
(i)	Rou	Maximum Route Duration	$\sum_{i,j\in V} tt_{ij} \cdot X_{ijkt} \leq maxDur \quad \forall k \in K, t \in T$	DVRP, LPRP
(j)	Rou	Transshipment Facility	$Z_{ikt} \leq T_t \quad \forall i \in V, k \in K_{TF}, t \in T$	DVRP, LPRP

Legend:
Act. - Activity | Pro - Production | Inv - Inventory | Rou - Routing | IS - Initial Solution | PLSP - Partial LSP | DVRP - Daily VRP | LPRP - Local PRP

To impose a minimum lot size $minLot_{fm}$ per family f in each cutting line m, constraints (a) are added. Constraints (b) and (c) impose a maximum time both on the time to ship products after production and on the time to consume products after reception at the meat stores. Constraints (d) ensure that the cutting lines are adequate to cut certain types of meat. For instance, to process minced meat, a different type of cutting line is needed. The parameter $comp_{mp}$ is equal to one if cutting line m is able to process meat product p. Constraints (e) ensure that the time spent on family setups, product setups and processing does not exceed the time capacity cap_{mt} of each line m in each period t. Constraints (f) impose the safety stocks ss_{ip} agreed with each meat store i and product p. This is one method to avoid inadequate stock levels at the end of each planning iteration. Additionally it allows a better absorption of forecast errors when implementing the solutions in real world. Constraints (g) impose the delivery time window at each store. Constraints (h) and (i) model a maximum number of visits $maxVis$ and a maximum duration $maxDur$ for each route. Finally, Constraints (j) model the transshipment facility. We consider that the set of vehicles based at the transshipment facility K_{TF} can only be used if a transshipment operation is performed.

6.3. Results Analysis

To provide managerial insights regarding the integration of the planning process of the considered meat store chain, we performed a set of experiments. We recreated the planning process of the company by devising a rolling horizon approach with only two periods of demand visibility. Note that currently the company does not integrate production decisions with the decisions made by each store (following a Retailer-Managed Inventory (RMI) policy). Therefore, we assume that the company solution can be obtained by using the initial solution of our approach in a rolling horizon approach considering two periods.

We tested the case where all the decisions are integrated while solving the problem, which mimics a VMI policy, and the case where a larger demand visibility is assumed, which would require stores to forecast demands further in time. Table 7 shows the results obtained in the experiments comparing them to the company solution. These results are obtained for instances created using the company data corresponding to the month of June 2015, performing the rolling horizon approach for the entire month.

Table 7: Case study results

	Company Solution	Integrated Production	Increased Visibility	Integrated Production Increased Visibility
Cost	**2RMI**	**2VMI**	**7RMI**	**7VMI**
Routing Cost	170633.00 €	149435.00 €	125793.00 €	118802.00 €
Total Transshipment Cost	7182.00 €	7182.00 €	8208.00 €	8208.00 €
Retailers Holding Cost	29445.18 €	29164.15 €	30124.64 €	30160.04 €
Suppliers Holding Cost	959.56 €	869.60 €	13265.29 €	5799.17 €
Cost Per Delivery	69.69 €	63.47 €	84.35 €	78.58 €
Total Cost	**208219.80 €**	**186650.70 €**	**177390.93 €**	**162969.21 €**
Cost Reduction		21 569.10 €	30 828.87 €	45 250.59 €
Percentage Savings		10.36%	14.81%	21.73%
Other KPIs				
Number of Routes	395	328	320	291
Number of Visits	2988	2941	2103	2074
Total Driving Time	3345.73 h	2993.40 h	2388.51 h	2277.71 h
Setup Time	292.92 h	306.30 h	286.58 h	257.08 h

We present cost related indicators and other operational Key Performance Indicators (KPIs) as well. In Table 7, the company solution corresponds to the case where each planning iteration considers two periods and a RMI policy. We achieve a cost reduction of 10.36% by integrating the production decisions (VMI policy) maintaining the demand visibility of two periods. When we increase demand visibility for seven periods and apply a RMI policy, the cost is reduced by 14.81%. Finally, increasing demand visibility and applying a VMI policy results in a cost reduction of 21.73%

An interesting behaviour is shown by other indicators. Part of the total cost is transferred to the supplier. However, this transfer allows for a large reduction in the routing cost. Furthermore, the cost per delivery raises when demand visibility is increased. However, inventory and routing integrated decisions allowed for a large reduction in the number of visits to the stores.

7. Conclusion

In this paper, a large multi-product PRP with time windows is addressed by means of a F&O based matheuristic. A novel mathematical formulation was proposed in order to provide integrated production, inventory, and routing plans for a vertical meat store chain. The large-size instances result in intractable problems which have to be tackled by efficient solution methods. This fact motivated us to propose a novel size reduction and decomposition technique to the PRP allowing for the construction of good quality initial solutions regardless of the size of the problem. Since these solutions are built by several IRP solutions plus a CLSP they do not integrate all decisions (only inventory management and routing decisions are jointly optimized). Therefore, these initial solutions are very good for problems where the link between the production and routing activities is not strong.

We devised a F&O based matheuristic for the PRP so as to integrate all decisions. Our approach iteratively integrates production, inventory management and routing decisions by solving different MIPs with variable size and scope. Matheuristics are still quite rare both in the IRP and PRP literature. Furthermore, to best of our knowledge, adjusting the size of the subproblems based on the runtime of previous iterations is a fresh contribution to the literature related to the F&O heuristic.

It was shown that the algorithm is efficient for solving large-sized instances both for the IRP and PRP. New best solutions are provided for several instances in shorter runtime, compared to state-of-the-art branch-and-cut implementations. Furthermore, we test our region decomposition approach with large IRP instance and show that the algorithm benefits from it.

Additionally, we presented a case study considering a European meat store chain. A set of more complex instances was solved in order to validate the ideas proposed in this paper within a real-world context. The challenge considers additional constraints which include multiple production lines with different specifications and one transshipment facility. Although it is necessary to introduce new constraints, the extensions are trivial and enforce the value of matheuristic approaches that are strongly based on mathematical formulations. After increasing demand visibility and considering a VMI policy, our solution approach achieves savings of 21.73% compared to the company's solution.

Acknowledgements

This research was partly supported by the PhD grant SFRH/BD/108251/2015, awarded by the Portuguese Foundation for Science and Technology (FCT), and by the Fonds de recherche du Québec - Nature et technologies. This support is gratefully acknowledged.

References

[1] Absi, N., Archetti, C., Dauzère-Pérès, S., and Feillet, D. (2015). A two-phase iterative heuristic approach for the production routing problem. *Transportation Science*, 49(4):784–795.

[2] Adulyasak, Y., Cordeau, J.-F., and Jans, R. (2014a). Formulations and branch-and-cut algorithms for multivehicle production and inventory routing problems. *INFORMS Journal on Computing*, 26(1):103–120.

[3] Adulyasak, Y., Cordeau, J.-F., and Jans, R. (2014b). Optimization-based adaptive large neighborhood search for the production routing problem. *Transportation Science*, 48(1):20–45.

[4] Adulyasak, Y., Cordeau, J. F., and Jans, R. (2014c). The production routing problem: A review of formulations and solution algorithms. *Computers and Operations Research*, 55:141–152.

[5] Adulyasak, Y., Cordeau, J.-F., and Jans, R. (2015). Benders decomposition for production routing under demand uncertainty. *Operations Research*, 63(4):851–867.

[6] Archetti, C., Bertazzi, L., Paletta, G., and Speranza, M. G. (2011). Analysis of the maximum level policy in a production-distribution system. *Computers & Operations Research*, 38(12):1731–1746.

[7] Armentano, V., a.L. Shiguemoto, and Løkketangen, a. (2011). Tabu search with path relinking for an integrated production–distribution problem. *Computers & Operations Research*, 38(8):1199–1209.

[8] Bard, J. F. and Nananukul, N. (2008). The integrated production–inventory–distribution–routing problem. *Journal of Scheduling*, 12(3):257–280.

[9] Bard, J. F. and Nananukul, N. (2009). Heuristics for a multiperiod inventory routing problem with production decisions. *Computers & Industrial Engineering*, 57(3):713–723.

[10] Bard, J. F. and Nananukul, N. (2010). A branch-and-price algorithm for an integrated production and inventory routing problem. *Computers & Operations Research*, 37(12):2202–2217.

[11] Boudia, M., Louly, M. A. O., and Prins, C. (2007). A reactive grasp and path relinking for a combined production-distribution problem. *Computers & Operational Research*, 34(11):3402–3419.

[12] Boudia, M., Louly, M. A. O., and Prins, C. (2008). Fast heuristics for a combined production planning and vehicle routing problem. *Production Planning & Control*, 19(2):85–96.

[13] Boudia, M. and Prins, C. (2009). A memetic algorithm with dynamic population management for an integrated production–distribution problem. *European Journal of Operational Research*, 195(3):703–715.

[14] Brown, G., Keegan, J., Vigus, B., and Wood, K. (2001). The kellogg company optimizes production, inventory, and distribution. *Interfaces*, 31(6):1–15.

[15] Çetinkaya, S., Üster, H., Easwaran, G., and Keskin, B. B. (2009). An integrated outbound logistics model for frito-lay: Coordinating aggregate-level production and distribution decisions. *Interfaces*, 39(5):460–475.

[16] Chandra, P. and Fisher, M. L. (1994). Coordination of production and distribution planning. *European Journal of Operational Research*, 72(3):503 – 517.

[17] Chen, H. (2015). Fix-and-optimize and variable neighborhood search approaches for multi-level capacitated lot sizing problems. *Omega*, 56:25 – 36.

[18] Coelho, L. C. and Laporte, G. (2013). A branch-and-cut algorithm for the multi-product multi-vehicle inventory-routing problem. *International Journal of Production Research*, 51(23-24):7156–7169.

[19] Dantzig, G. B. and Ramser, J. H. (1959). The Truck Dispatching Problem. *Management Science*, 6(1):80–91.

[20] Dolan, E. and Moré, J. J. (2002). Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91(2):201–213.

[21] Fumero, F. and Vercellis, C. (1999). Synchronized development of production, inventory, and distribution schedules. *Transportation Science*, 33(3):330–340.

[22] Hartigan, J. A. and Wong, M. A. (1979). Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108.

[23] Helber, S. and Sahling, F. (2010). A fix-and-optimize approach for the multi-level capacitated lot sizing problem. *International Journal of Production Economics*, 123(2):247 – 256.

[24] Karimi, B., Fatemi Ghomi, S. M. T., and Wilson, J. M. (2003). The capacitated lot sizing problem: A review of models and algorithms. *Omega*, 31(5):365–378.

[25] Larrain, H., Coelho, L. C., and Cataldo, A. (2017). A variable mip neighborhood descent algorithm for managing inventory and distribution of cash in automated teller machines. *Computers & Operations Research*, 85:22–31.

[26] Lei, L., Liu, S., Ruszczynski, A., and Park, S. (2006). On the integrated production, inventory, and distribution routing problem. *IIE Transactions*, 38(11):955–970.

[27] Pochet, Y. and Wolsey, L. (2006). *Production planning by mixed integer programming (Springer Series in Operations Research and Financial Engineering).* Springer-Verlag, New York, Inc., Secaucus, NJ, USA.

[28] Qiu, Y., Qiao, J., and Pardalos, P. M. (2017). A branch-and-price algorithm for production routing problems with carbon cap-and-trade. *Omega*, 68:49–61.

[29] Ruokokoski, M., Solyali, O., Cordeau, J.-F., Jans, R., and Sural, H. (2010). Efficient Formulations and a Branch-and-Cut Algorithm for a Production-Routing Problem. *GERAD Technical Report G-20 10-66, HEC Montreal Canada*, pages 1–43.

[30] Sarmiento, A. M. and Nagi, R. (1999). A review of integrated analysis of production–distribution systems. *IIE Transactions*, 31(11):1061–1074.

[31] Schmid, V., Doerner, K. F., and Laporte, G. (2013). Rich routing problems arising in supply chain management. *European Journal of Operational Research*, 224(3):435–448.

[32] Solyalı, Q. and Süral, H. (2009). A Relaxation Based Solution Approach for the Inventory Control and Vehicle Routing Problem in Vendor Managed Systems. *Modeling, Computation and Optimization*, (2000):171–189.

[33] Stadtler, H. and Kilger, C. (2010). *Supply Chain Management and Advanced Planning: Concepts, Models, Software, and Case Studies.* Springer Publishing Company, Incorporated, 4th edition.

[34] Thomas, D. J. and Griffin, P. M. (1996). Coordinated supply chain management. *European Journal of Operational Research*, 94(1):1 – 15.

[35] Wagner, H. M. and Whitin, T. M. (1958). Dynamic version of the economic lot size model. *Management science*, 5(1):89–96.

Appendix A. Decomposition strategy base formulations

Before presenting the base formulations of each decomposition strategy we comment on the order in which they are called. Since the largest expected size reduction (obtained in the first phase) usually comes from the routing part, and as it is commonly the largest portion of the total cost, the first decomposition strategy starts by exploring the *Daily VRP* formulation which focuses on routing cost. The second decomposition strategy, which uses the *Partial LSP* formulation, is not so promising at this point because all the inventories have already been optimized, taking into account the delivery schedule of the initial solution. Though large gains are not expected, improvements can still be achieved by jointly optimizing inventories of the supplier and of the retail sites, particularly in the cases where the problem is divided into regions. Furthermore, the production line setups may also be changed to better redistribute the production quantities needed to satisfy the shipping quantities in each period. In case the problem considers setup costs, these can also be improved. In fact, the order in which the first two decomposition strategies are explored is irrelevant as the decisions to be analysed by each of them are completely independent. However, in the third decomposition strategy all the decisions inherent to the PRP are taken into account locally. Integrating most of the decisions taken at the supplier with the routing decisions of a set of retail sites results in difficult subproblems. For this reason, we decided to follow this decomposition strategy in the last place (after reaching a local optimum regarding the routing part of the solution). The following subsections describe the three base formulations of the decomposition strategies comprised in our F&O approach.

Appendix A.0.1. Daily VRP base formulation

The inputs to this model comprise a single period t, a set of vehicles $\mathcal{K}$, a set $\mathcal{V}$ including the retail sites visited by these vehicles in the incumbent solution, and the delivery quantities made to each retail site, denoted by d_i. The deliveries are needed because the fleet is not homogeneous (different capacities vc_k), thus their quantities have to be taken into account in the vehicle capacity constraints. Additionally, vehicles must respect the time windows $[a_i, b_i]$ of each location i. The time needed for a vehicle to traverse an arc (i, j) is given by tt_{ij}. Note that it is not necessary to load more than one period, as there are no dependencies between periods in this formulation, given that the delivery quantities are fixed. In fact, this formulation is similar to the routing part of the PRP formulation presented in Section 3 but the index t is dropped. The model works with the binary variables X_{ijk}, which are equal to one if vehicle k traverses arc (i, j). Continuous variables W_{ik} define the time at which vehicle k arrives at location i.

(DailyVRP):

$$\text{minimize} \sum_{i\in\mathcal{V}}\sum_{j\in\mathcal{V}}\sum_{k\in\mathcal{K}} tc_{ij}\cdot X_{ijk} \tag{A.1}$$

s.t.

$$\sum_{j\in\mathcal{V}}\sum_{k\in\mathcal{K}} X_{ijk} = 1 \quad \forall i \in \mathcal{V}' \tag{A.2}$$

$$\sum_{j\in\mathcal{V}} X_{ijk} - \sum_{j\in\mathcal{V}} X_{jik} = 0 \quad i \in \mathcal{V}, k \in \mathcal{K} \tag{A.3}$$

$$\sum_{i\in\mathcal{V}'}\sum_{j\in\mathcal{V}} X_{ijk}\cdot d_i \le vc_k \quad \forall k \in \mathcal{K} \tag{A.4}$$

$$W_{ik} + tt_{ij} \le W_{jk} + M\cdot(1 - X_{ijk}) \quad \forall i \in \mathcal{V}, j \in \mathcal{V}, k \in \mathcal{K} \tag{A.5}$$

$$a_i \le W_{ik} \le b_i \quad \forall i \in \mathcal{V}, k \in \mathcal{K} \tag{A.6}$$

$$\begin{aligned} X_{ijk} \in \{0,1\} \quad \forall i, j \in \mathcal{V}, k \in \mathcal{K} \\ W_{ik} \ge 0 \quad \forall i \in \mathcal{V}, k \in \mathcal{K}. \end{aligned} \tag{A.7}$$

The objective function (A.1) of the *Daily VRP* formulation simply accounts for the routing cost to perform the deliveries that are currently scheduled in the solution and were loaded to be optimized.

Constraints (A.2) ensure that the scheduled deliveries in the incumbent solution must continue to be performed after solving this subproblem. Constraints (A.3) are the so called flow conservation constraints to ensure that if a vehicle visits a node, it has to leave that node. Constraints (A.4) are to ensure that vehicle capacities are respected. Constraints (A.5) define the time at which each retail site is visited by a vehicle. This time is used to define constraints (A.6), where the time windows $[a_i, b_i]$ of each retail site must be respected. M is a big number which is at least the duration of a day.

Appendix A.0.2. Partial LSP base formulation

The $PartialLSP$ formulation uses the same variables presented before but in this case, index k is dropped from delivery quantities D_{iptl} as the visits are assumed to be fixed:

(PartialLSP):

$$\begin{aligned} \text{minimize} \quad & \sum_{p\in\mathcal{P}}\sum_{m\in\mathcal{M}}\sum_{g\in\mathcal{T}}\sum_{h=0}^{g}\sum_{t=g+1}^{|\mathcal{T}|} h_{0p}\cdot P_{pmht} \\ & + \sum_{i\in\mathcal{V}'_t}\sum_{p\in\mathcal{P}}\sum_{t\in\mathcal{T}}\sum_{h=0}^{t}\sum_{l=t+1}^{|\mathcal{T}|} h_p\cdot D_{iphl} \end{aligned} \tag{A.8}$$

s.t. (2), (3), (4), (5), (8),

$$\sum_{i\in\mathcal{V}'_{kt}}\sum_{p\in\mathcal{P}}\sum_{l\in\mathcal{T}} D_{iptl} \le c_{kt} \quad \forall k\in\mathcal{K}, t\in\mathcal{T} \tag{A.9}$$

$$\sum_{g=0}^{t}\sum_{m\in\mathcal{M}} P_{pmgt} = d_{pt}^{external} + \sum_{i\in\mathcal{V}'}\sum_{l=t}^{|\mathcal{T}|} D_{iptl} \quad \forall p\in\mathcal{P}, t\in\mathcal{T} \tag{A.10}$$

$$\sum_{t=0}^{l} z_{it}\cdot D_{iptl} = d_{ipl} \quad \forall i\in\mathcal{V}', p\in\mathcal{P}, l\in\mathcal{T} \tag{A.11}$$

$$P_{pmtl}, D_{iptl} \ge 0. \tag{A.12}$$

The objective function (A.8) is similar to the one presented in the LSP formulation used in the initial solution. However, the inventory holding costs are now accounted for each location and not only for the supplier. In order to define the necessary conditions to model production line capacities, setups, and the warehouse capacity of the supplier, constraints (2), (3), (4), and (5) are added from the $MPRPTW$ formulation presented in Section 3. Constraints (8), from the $MPRPTW$ formulation, are also added to ensure that warehouse capacities are respected for each retail site. However, the index k is now dropped. The vehicles visiting the selected retail sites in a certain period may also visit other retail sites that were not loaded into the model. Accordingly, we define a partial capacity c_{kt} for each vehicle k in each shipping period t. Constraints (A.9) ensure that the deliveries performed to the set of visited retail sites $\mathcal{V}'_{kt}$ (visited by vehicle k in shipping period t) do not exceed the capacity of the vehicle. Constraints (A.9) ensure that the deliveries performed to the set of visited retail sites $\mathcal{V}'_{kt}$ (visited by vehicle k in shipping period t) do not exceed the capacity c_{kt} of the vehicle in each shipping period. Constraints (A.10) force the production quantities to satisfy both the demands of loaded and unloaded retail sites. Constraints (A.11) ensure the demand satisfaction of each loaded retail site. Finaly, the bounds of all variables are defined by constraints (A.12).

Appendix A.0.3. Local PRP base formulation

The $LocalPRP$ formulation includes most of the constraints presented in the $MPRPTW$ formulation and it is defined as follows:

(LocalPRP):

$$\begin{aligned}
\text{minimize} \quad & \sum_{p\in\mathcal{P}}\sum_{m\in\mathcal{M}}\sum_{g\in\mathcal{T}}\sum_{h=0}^{g}\sum_{t=g+1}^{|\mathcal{T}|} h_{0p}\cdot P_{pmht} \\
& + \sum_{i\in\mathcal{V}'_t}\sum_{p\in\mathcal{P}}\sum_{k\in\mathcal{K}_t}\sum_{t\in\mathcal{T}}\sum_{h=0}^{t}\sum_{l=t+1}^{|\mathcal{T}|} h_p\cdot D_{ipkhl} \\
& + \sum_{i\in\mathcal{V}_t}\sum_{j\in\mathcal{V}_t}\sum_{k\in\mathcal{K}_t}\sum_{t\in\mathcal{T}} tc_{ij}\cdot X_{ijkt}
\end{aligned} \tag{A.13}$$

s.t. (2), (3), (4), (5),

$$\sum_{g=0}^{t}\sum_{m\in\mathcal{M}} P_{pmgt} = d_{pt}^{ext} + \sum_{i\in\mathcal{V}_t}\sum_{k\in\mathcal{K}_t}\sum_{l=t}^{|\mathcal{T}|} D_{ipktl} \quad \forall p\in\mathcal{P}, t\in\mathcal{T} \tag{A.14}$$

$$\sum_{k\in\mathcal{K}_t} D_{ipktl} = q_{iptl} \quad \forall i\in\mathcal{V}_t^{out}, p\in\mathcal{P}, t\in\mathcal{T}, l\in\mathcal{T}, t\le l \tag{A.15}$$

$$\sum_{k\in\mathcal{K}_t} Z_{ikt} = 1 \quad \forall i\in\mathcal{V}_t^{out}, t\in\mathcal{T}. \tag{A.16}$$

The objective function minimizes both the supplier and retail sites' holding costs, and the routing cost. From the $MPRPTW$ formulation, constraints (2), (3), (4), and (5) are added to model the family setups, the product setups, the time capacity of each production line, and the warehouse capacity constraints of the supplier, respectively. Constraints (A.14) impose that the produced quantities must satisfy both the internal and external demand considered in the model. Note that each period considers a different set of vehicles $\mathcal{K}_t$, which includes the vehicles performing the routes in the incumbent solution visiting some retail site belonging to $\mathcal{V}'$. The sets of nodes $\mathcal{V}_t$ includes all the retail sites that can be visited in a period (white and grey nodes in Figure 7). Constraints (A.15) force the delivery quantities to the fixed retail sites (gray nodes), belonging to the set $\mathcal{V}_t^{out}$, to be equal to the delivery quantities q_{iptl} defined by the incumbent solution. For each retail site in these sets, the delivery quantities will be fixed, thus we also now know that their warehouse capacity is not violated. However, the vehicle that performs the delivery can still change. Constraints (A.16) ensure that a visit is still performed to the fixed retail sites of each period. The remaining constraints belong to the $MPRPTW$ formulation and are added with proper changes in the sets to be considered. For the selected retail sites, constraints (7) ensure demand satisfaction of the free retail sites. In constraints (9) and (10), which model vehicle capacity and delivered quantities (respectively), the set $\mathcal{V}'$ is replaced by the sets $\mathcal{V}_t\backslash\{0\}$ and the set $\mathcal{K}$ is substituted by the sets $\mathcal{K}_t$, in each period t. For the remaining constraints regarding the routing part of the problem, namely (11) - (15), the set $\mathcal{V}'$ is substituted by the sets $\mathcal{V}_t$ and the set $\mathcal{K}$ by the sets $\mathcal{K}_t$, in each period t.