TOP (2018) 26:30–67
https://doi.org/10.1007/s11750-017-0448-1

ORIGINAL PAPER

Optimization model for a production, inventory, distribution and routing problem in small furniture companies

Pedro L. Miranda[1] · **Reinaldo Morabito**[1] · **Deisemara Ferreira**[2]

Received: 12 December 2016 / Accepted: 2 May 2017 / Published online: 20 May 2017

Abstract Production and distribution are two key decisions in supply chain planning. In order to achieve an effective operational performance, it is important for these two decisions to be integrated, especially in supply chains with low inventory levels. In this paper, we propose a mixed integer programming model to integrate production, inventory, distribution and routing decisions in a single framework. The model was inspired by small Brazilian furniture companies and focuses on production and distribution decisions at an operational level. In particular, we consider a scenario in which only one production line and one vehicle, which makes multiple trips over the planning horizon, are available to produce items and deliver final products, respectively. We also take into account some features rarely considered in the literature, but commonly found in real-world applications, such as producing and stocking multiple items, distribution routes extending over one or more periods, multiple time windows and customers' due dates. Computational tests on a set of randomly generated instances were carried out using a well-known optimization software and six relax-and-fix heuristics, which explore different criteria for partitioning and fixing variables. We also implemented

The authors would like to thank FAPESP (Grants 2010/10133-0 and 2014/10565-8) and CNPq (Grants PQ 140179/2014-3 and 312569/2013-0) for their financial support.

✉ Reinaldo Morabito
morabito@ufscar.br

Pedro L. Miranda
pmiranda@dep.ufscar.br

Deisemara Ferreira
deise@ufscar.br

1 Department of Production Engineering, Federal University of São Carlos, São Carlos, SP, Brazil

2 Department of Physics, Chemistry and Mathematics, Federal University of São Carlos, Sorocaba, SP, Brazil

two hybrid heuristics in which an initial solution is first constructed and then fed into the optimization software to improve it. The results showed that one relax-and-fix and the two hybrid heuristics performed better than the solver on the largest instances.

Keywords Production, inventory, distribution routing problem · Production routing problem · Integrated production–distribution planning · Furniture industry

Mathematics Subject Classification 90B06 · 90C11 · 90C27

1 Introduction

A supply chain is an integrated process whereby various entities, such as suppliers, manufacturers, distributors and retailers work together in order to acquire raw materials, transform these raw materials into different final products and deliver these final products to final customers (Beamon 1998). A typical supply chain consists of sequential activities of production, storage and distribution. Usually, each individual activity is planned and optimized using decisions made in preceding activities (Adulyasak et al. 2015b). For example, in order to reduce production, inventory and setup costs, production plans are defined considering capacity limitations, raw material availability, machine setup times, storage limitations, etc. Afterwards, production planning decisions are used as input to make distribution decisions in order to determine how products should be delivered to customers. However, in today's globalized supply chains and high competitive markets, firms should use their resources efficiently to strengthen customer service levels and reduce lead times and total costs. Taking this into account, considering production and transportation planning activities simultaneously may lead to a boost in efficiency and saving costs (Chen 2004; Díaz-Madroñero et al. 2015).

In this paper, we studied a coupled lot-sizing and routing problem considering common features in some real-world applications. Typically, the lot-sizing problem (LSP) consists of deciding the production quantities and inventory levels over a given planning horizon in order to minimize production, setup and inventory costs. On the other hand, the vehicle routing problem (VRP) aims to design vehicle routes to make deliveries to customers at a minimum logistics cost. As shown above, in practice it is common for lot-sizing and vehicle routing decisions to be made separately. Integrating these decisions may lead to: (i) reducing the operational cost of the production–distribution system, and (ii) meeting customers' demands more effectively. This coupled problem, obtained by integrating the LSP and VRP decisions, optimizes production, inventory and routing costs by jointly determining the production quantities and inventory levels in each period, the periods when each customer should be visited, the delivery quantities of each product in each visit and the corresponding vehicle routes.

Here, we propose a mixed integer program (MIP) to represent and solve an operational problem in which the manufacturer has only one production line with low inventory levels and only one delivery vehicle that may perform multiple trips over a multi-period short-term planning horizon (e.g., the days of a week). The model includes various characteristics and complicating constraints commonly found in practice, such as producing and stocking parts (instead of final products) in each period, limited production and distribution capacities at operational levels, routes extending over one or more production periods of the planning horizon, multiple daily time windows and customers' due dates. These features are not commonly found in the literature concerning integrated lot-sizing and routing problems or integrated scheduling and routing problems (see Sect. 2) and lead to a more difficult problem from the modeling and computational points of view.

Routes spanning several periods might be found, for example, in long-haul carrier services (see Goel 2012; Rancourt et al. 2013) and in liquid gas industries (Savelsbergh and Song 2008). Studies carried out by Goel (2012) and Rancourt et al. (2013) are also good examples of vehicle routing problems with multiple time windows. Other cases where multiple time windows can be found are, for example, distribution food companies (see Amorim et al. 2014) and furniture and electronic device delivery services (see Belhaiza et al. 2014). Our study is motivated by the practice of small Brazilian furniture companies, in which the manufacturers are responsible for both producing and delivering the final products. In this scenario, the last production stage, namely painting, and the distribution stage should be planned at an operational level in a coordinated way in order to obtain detailed production (lot-sizing and scheduling) and delivery (vehicle routing) plans that strike a balance between inventory, setup and routing costs.

This problem can be seen as a combined lot-sizing, production scheduling and vehicle routing problem over a short-term planning horizon, in which we need to define which products to produce as well as the lot sizes of these products for each day. Therefore, we define the production sequence for these lots of products over the days, without being concerned about the specific sequence within each day. Defining a sequence for each day is not necessary as setups are not sequence-dependent and because routes departing from the depot in a given day do not carry production from that day, but from the inventory available at the end of the previous day.

The contribution of this paper is twofold. First, we modeled and studied a number of practical considerations commonly found in real-world applications. To the best of our knowledge, this is the first attempt in which all of these features in the context of integrated lot-sizing/scheduling and vehicle routing problems have been modeled. Second, as the resulting MIP model is hard to solve, we developed several relax-and-fix (RF) heuristics, each one considering different criteria for partitioning and fixing variables. These heuristics enabled us to find good feasible solutions for the largest instances in competitive computational times.

The remainder of the paper is organized as follows: Sect. 2 briefly presents related work on the integrated production and distribution planning problem. In Sect. 3, we describe the production and distribution processes in furniture companies based on the current practice of a representative Brazilian furniture company. The MIP formulation integrating lot-sizing/scheduling and vehicle routing decisions is presented in Sect. 4. In Sect. 5, we describe different RF strategies to solve the model. Computational tests with a state-of-the-art commercial solver and the proposed heuristics are outlined in Sect. 6. Finally, Sect. 7 provides some concluding remarks and possible research perspectives.

2 Literature review

Given the increasing demand for an integrated production and distribution planning, various optimization models and solution methods have been proposed to help the decision-making process in different hierarchical levels (i.e., strategic, tactical and operational). Surveys of models and solution methods are available in the literature, for example, studies carried out by Mula et al. (2010) at strategic and tactical levels, Chen (2010) and Moons et al. (2017) at an operational level, and Adulyasak et al. (2015b) at a tactical-operational level.

At the strategic level, the main concern is the design of the production–distribution system, which entails deciding the number and location of production facilities and warehouses, selecting transportation modes, capacity planning, among others (Sarmiento and Nagi 1999; Vidal and Goetschalckx 1997). At the tactical level, production and distribution planning uses aggregated information to determine production lot-sizes, inventory levels, delivery quantities and vehicle routes over a given planning horizon, considering production and distribution capacity availability (Adulyasak et al. 2015b; Díaz-Madroñero et al. 2015). At this point, it is not necessary to consider detailed production and delivery operations, which are determined later on a daily basis.

At the operational level, production and distribution planning focuses mainly on machine and vehicle scheduling issues, and attempts to optimize detailed production and delivery operations, considering each customer order individually. The main decisions are to assign customer orders to resources, determining the start and completion times for each customer order, assigning customer orders to delivery vehicles and defining delivery routes and delivery times for each customer order. Therefore, the outcome is a detailed production and distribution schedule with the exact timing at which each individual customer order is executed to satisfy customer demands on time (Moons et al. 2017).

Detailed scheduling of production and distribution operations is important in many different practical situations. For example, in industries with low inventory levels, these operations need to be scheduled together to achieve a proper trade-off between transportation costs and customer service levels (Chen and Pundoor 2006; Chen and Vairaktarakis 2005). Furthermore, in perishable or time-sensitive product industries, such as food (Chen et al. 2009; Farahani et al. 2012), industrial adhesive materials (Armstrong et al. 2007; Geismar et al. 2008), newspapers (Chiang et al. 2009; Russell et al. 2008), etc., there is no inventory between production and distribution and products must be delivered within very short lead times.

This paper focuses on detailed production and distribution decisions over multi-period short-term planning horizons, motivated by small furniture companies with low inventory levels. It considers relevant lot-sizing/scheduling and inventory decisions in each period (e.g., a day) and vehicle routing and scheduling operations along the horizon (e.g., a week). In this section, we focus on studies explicitly dealing with integrated production–distribution problems where the manufacturer must decide the lot-sizes, inventory levels, delivery quantities and delivery routes over a finite and multi-period planning horizon in order to minimize production, inventory and routing costs. In the literature, this problem is known by different names, such as the production routing problem (PRP) (Adulyasak et al. 2015b), the integrated production, inventory, distribution and routing problem (PIDRP) (Lei et al. 2006) and the integrated production and distribution problem (IPDP) (Armentano et al. 2011). We prefer to use the name "production, inventory, distribution and routing problem (PIDRP)", as it clearly highlights the main decisions involved in the small furniture industry problem.

We note that the PIDRP is a generalization of the lot-sizing problem with direct shipments (Molina et al. 2016), in which routing decisions are disregarded, as well as the inventory routing problem (IRP), where production decisions are dropped (Adulyasak et al. 2014b). The literature related to the PIDRP is relatively new and scarce when compared with the LSP and VRP literature. Indeed, the seminal paper concerning this problem was written by Chandra and Fisher (1994), who studied the economic value of integrating and coordinating production and routing decisions.

Table 1 shows the differences between the problem we studied and other research in the literature. Column "References" shows the papers in chronological order. Column "#Product" indicates if a given study considered one ("Single") or more ("Multiple") final products. Column "Time windows (TW)" shows if the problem includes no time window ("No TW"), one time window ("One TW") or multiple time windows ("Multiple TW") over the planning horizon. Column "Route duration" states whether routes are restricted to fit into one period ("<1") or they may span over more than one period (">1"). Column "#Routes per vehicle" indicates if a vehicle performs at most one route in each period ("1 per period") or if multiple trips are allowed (">1 per period"). Finally, the column "Solution method" indicates the approach used to solve the problem.

Most of the papers in Table 1 focused on problems with a single product and homogeneous vehicles, which perform at most one route per period. In order to solve this problem, researchers proposed different kind of methods, such as branch-and-cut (Adulyasak et al. 2014a), Benders-based branch-and-cut (Adulyasak et al. 2015a), branch-and-price-based (Bard and Nananukul 2009a, 2010), mathematical programming-based heuristics (Absi et al. 2014; Archetti et al. 2011; Bertazzi et al. 2005), greedy randomized adaptive search procedure (GRASP) (Boudia et al. 2007), genetic algorithm with population management (GAPM) (Boudia and Prins 2009), reactive tabu search (RTS) (Bard and Nananukul 2009b) and adaptive large neighborhood search (ALNS) (Adulyasak et al. 2014b).

Papers considering multiple products are even scarcer and, similar to the previous studies, research focused on developing heuristics and metaheuristics solution methods. Chandra and Fisher (1994) proposed a decomposition approach and evaluated the value of integrating production and routing decisions. The results showed that this integration provides saving costs between 3–20% compared with the traditional decoupled planning process. Fumero and Vercellis (1999) proposed a Lagrangian heuristic that decomposes the integrated problem into four subproblems which are easier to solve, namely production, inventory, distribution and routing subproblems. Each subproblem was solved and a feasible solution for the integrated problem was constructed by combining the subproblem solutions. Computational tests reported similar conclusions to those proposed by Chandra and Fisher (1994).

Shiguemoto and Armentano (2010) studied a PIDRP considering the uncapacitated dynamic lot-sizing problem in the production part. The authors proposed a tabu

Table 1 Summary of PIDRP related studies

References	#Products		Time windows (TW)			Route duration		#Routes per vehicle		Solution method
	Single	Multiple	No TW	One TW	Multiple TW	<1 period	>1 period	1 per period	>1 per period	
Chandra and Fisher (1994)		✓	✓			✓		✓		Decomposition-based heuristic
Fumero and Vercellis (1999)		✓	✓			✓		✓		Lagrangian relaxation-based heuristic
Bertazzi et al. (2005)	✓		✓			✓		✓		Decomposition-based heuristic
Lei et al. (2006)	✓		✓			✓			✓	Decomposition-based heuristic
Boudia et al. (2007)	✓		✓			✓		✓		GRASP[a]
Boudia and Prins (2009)	✓		✓			✓		✓		Genetic algorithm
Bard and Nananukul (2009b)	✓		✓			✓		✓		Tabu search
Bard and Nananukul (2009a)	✓		✓			✓		✓		Branch-and-price-based heuristic
Bard and Nananukul (2010)	✓		✓			✓		✓		Branch-and-price-based heuristic
Shiguemoto and Armentano (2010)		✓	✓			✓		✓		Tabu search
Archetti et al. (2011)	✓		✓			✓		✓		MIP-based heuristic
Armentano et al. (2011)		✓	✓			✓		✓		Tabu search
Amorim et al. (2013)		✓		✓		✓		✓		CPLEX solver
Adulyasak et al. (2014a)	✓		✓			✓		✓		Branch-and-cut
Adulyasak et al. (2014b)	✓		✓			✓		✓		ALNS[b]
Absi et al. (2014)	✓		✓			✓		✓		MIP-based heuristic
Adulyasak et al. (2015a)	✓		✓			✓		✓		Benders-based branch-and-cut
Brahimi and Aouam (2016)		✓	✓			✓		✓		MIP-based heuristic
This paper		✓			✓		✓		✓	MIP-based heuristic

[a] Greedy randomized adaptive search procedure
[b] Adaptive large neighborhood search

search (TS) algorithm and solved a set of single product instances proposed by Bertazzi et al. (2005). Randomly generated instances with multiple products were used to compare TS with the classical decoupled approach, in which production and distribution planning are solved sequentially. For both sets of instances, the authors showed the advantages of TS in terms of solution quality. Later, Armentano et al. (2011) considered a limited production capacity and proposed two variants of a TS algorithm, with and without a path-relinking procedure. The authors compared their results with those of Bard and Nananukul (2009b) and Boudia and Prins (2009) for the single product case, and showed that both TS algorithms provided better results for all the instances. In addition, tests on randomly generated instances with multiple products showed that the TS with path-relinking achieved the best solutions. These solutions came at a higher computational cost, though.

Amorim et al. (2013) used the commercial solver CPLEX to solve a PIDRP with perishability considerations. To the best of our knowledge, this is the only study including sequencing decisions in the production part. The authors proposed two MIP models to evaluate the impact of batching versus lot-sizing decisions in production and distribution planning. They found that the best results are obtained when production decisions are made on the basis of the lot-sizing model.

Brahimi and Aouam (2016) presented two MIP formulations and proposed a hybrid solution procedure that combined an RF heuristic and a local search algorithm. This procedure decomposed the problem into two subproblems, the production–distribution subproblem and the routing subproblem, without removing the routing variables from the model, but only relaxing their integrality. An RF heuristic was used to solve the production–distribution subproblem, which determines lot-sizes, inventory levels and customers to be visited in each period. Then, a VRP local search algorithm was used in order to determine the delivery routes. Computational results showed that this hybrid procedure outperformed the XPRESS solver in terms of solution quality and CPU time.

All the papers considering multiple products assumed that the manufacturer owns a fleet of homogeneous vehicles, each vehicle performs at most one route per period and routes do not span over several periods (i.e., each vehicle must depart from and return to the depot within the same period). In the context of chemical industries, Lei et al. (2006) considered a problem in which the manufacturer had multiple production facilities, where each one produces the same product. Deliveries were made by a heterogeneous fleet of vehicles and each vehicle could perform multiple trips per period. As usual, routes could not span over more than one period. The authors formulated a MIP model and proposed a two-phase decomposition heuristic to solve it. In the first phase, a lot-sizing model with direct shipments was solved in order to decide production quantities, delivery quantities, inventory levels and the number of trips made by each vehicle. In the second phase, for each plant and time period, a heuristic procedure was applied to consolidate less than truckload (LTL) shipments and determine vehicle routes. Computational tests showed that this approach provided better solutions than those obtained by solving the full integrated model using the CPLEX solver. Results on a real dataset of a chemical company were also reported.

In summary, the PIDRP literature is rather scarce and most of the research done so far has focused on the problem with a single product and homogeneous fleet of vehicles. Solution methods are mainly decomposition heuristics, MIP-based heuristics and metaheuristics, which reflect the complexity of solving PIDRP problems using general-purpose optimization solvers or specific exact methods. In this context, this paper contributes to the literature by: (i) studying a PIDRP problem with characteristics rarely considered in the literature, but commonly found in some industries (e.g., furniture, carrier services, food, electronic devices, etc.), such as multiple products, multiple long duration trips, multiple time windows and customers' due dates; and (ii) proposing heuristics based on mathematical programming, such as relax-and-fix in order to solve the problem.

3 Problem description

The PIDRP considered in this paper consists of one production line that produces a set $\mathcal{C}$ of items, $c \in \mathcal{C}$, required to meet the demand of a set $\mathcal{P}$ of final products, $p \in \mathcal{P}$. To assemble one unit of product p, η_{cp} units of item c need to be produced. We assume that $\eta_{cp} = 0$ if $c \notin \mathcal{F}_p$, where $\mathcal{F}_p \subseteq \mathcal{C}$ represents the subset of items required to assemble product p. If production of item c takes place in period t, $t \in \mathcal{T}$, a fixed setup cost f_c is incurred. The production capacity in period t, measured in time units, is K_t and the time required to produce one unit of item c is ρ_c. Items produced in period t are only available for shipping in period $t+1$ and, therefore, shipments in period t must come from the inventory available at the end of period $t-1$. Production is make-to-order and back-orders are not allowed. The unit inventory holding cost of item c is h_c, and the initial and minimum inventory levels of item c are denoted by I_{c0} and $I_c^{\min}$, respectively.

Let $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ define a complete directed graph, where $\mathcal{N} = \{0, 1, \ldots, n+1\}$ is the set of nodes and $\mathcal{A} = \{(i, j) : i, j \in \mathcal{N}, i \neq j\}$ is the set of arcs. The depot is denoted by both 0 and $n+1$ and $\bar{\mathcal{C}} = \mathcal{N} \backslash \{0, n+1\}$ denotes the set of customers. The travel time (cost) of going from node i to node j is denoted by τ_{ij} (c_{ij}), $i, j \in \mathcal{N}$. For each customer i, $i \in \bar{\mathcal{C}}$, let d_{pi} be the demand of product p at customer i and $[\delta_{it}, \bar{\delta}_{it}]$ denote the time window of customer i in period t. These time windows represent the time intervals within which a delivery is allowed. In furniture companies, it is common for these time windows to be the regular business hours, therefore δ_{it} and $\bar{\delta}_{it}$ represent the beginning and the end of a regular customer working day. Each customer i must be visited once during s_i time units, without service interruption and before a preset deadline Δ_i. Consequently, a single time window has to be chosen for each customer. If customer i is visited in period t, then its service must start anytime between the initial time instant δ_{it} and the final time instant $\bar{\delta}_{it}$. If the vehicle arrives at the node before δ_{it}, then it has to wait until the time window opens to start the service. Furthermore, if the vehicle arrives after $\bar{\delta}_{it}$, then it has to wait until the next available time window opens to start the service.

Deliveries are made by a single vehicle that might perform several routes $r = 1, \ldots, R$ over the planning horizon. Each route must depart from node 0 and arrive at node $n+1$. Before starting a new route, the vehicle must be reloaded at the depot. We assume that the loading time is proportional to the service time of the customers served by the route, and therefore it is variable. The vehicle capacity is θ and each unit of product p on board of the vehicle consumes φ_p units of capacity. We do not impose a limit on the duration of the routes, so that they can depart from node 0 in period $t \in \mathcal{T}$ and arrive at node $n+1$ in period $t' \in \mathcal{T} : t' \geq t$. This means that routes can extend over several periods over the planning horizon.

The problem is to determine how many items to produce and stock in each period, the vehicle routes and the time each customer should be served over a finite multi-period planning horizon in order to minimize setup, inventory and routing costs. Note that we do not consider inventory holding costs at customers' sites as we do not assume that the manufacturer manages the customers' inventory. This scenario is commonly found, for example, in furniture industries. A typical furniture company produces a variety of final products, which are manufactured using hardwood lumber, hardwood plywood, medium density fiberboard (MDF), particleboard, steel, etc.

The production process starts at the cutting stage, whereby steel plates and coils are cut into smaller parts depending on the size of the final products. At the machining stage, specific operations, such as folding, drilling and welding give the final shape to parts and join some of them into sub-assemblies required for final products. Next, at the painting stage, parts and sub-assemblies are coated using liquid or powder paint. Many furniture companies use liquid paint since it requires less specialized equipment and does not need significant changeover times (setup times are considered insignificant). After the painting stage, parts and sub-assemblies are packaged and sent to the stock. Note that only parts and sub-assemblies, and not final products, are stocked. This practice is common in the furniture industry as the same part or sub-assembly can be used to assemble different final products.

Although the production process comprises multiple stages, in practice it suffices to determine a production plan only for the painting line, which is the production bottleneck. Therefore, plans for the remaining stages are obtained based on the requirements of the painting line. When planning the painting line, the most important decisions are the quantity to be produced (i.e., to be painted) and the inventory level of each item in each period (e.g., a day) of the planning horizon (e.g., a week). Therefore, production planners focus on lot-sizing and scheduling decisions at operational levels.

The distribution process starts by picking packages that contain the parts and sub-assemblies needed to assemble the final products. Afterward, packages are loaded into the vehicle and delivered to customers, which are usually retail stores. The products are assembled in the retail store or by the final customer (a kind of postponement strategy). Routes may extend over several days, as a result of the vast geographical dispersion of manufacturers and customers. Consequently, the vehicle may be unavailable for long periods of time (as it is on route). Furthermore, since there is only one vehicle, it might have to go on more than one route in order to meet all the customers' demands. Finally, as routes may extend over several days, each customer has multiple time windows, but it must be visited only once (see Rancourt et al. 2013). Similarly to the production process, distribution planning comprises several stages. In practice, vehicle routing represents the main decisions and picking and loading plans are obtained based on the delivery plan.

As described above, in the context of furniture companies, painting and delivery are considered the most relevant operations to meet the customers' demands. Therefore, we can see the problem studied here as an instance of a small furniture company

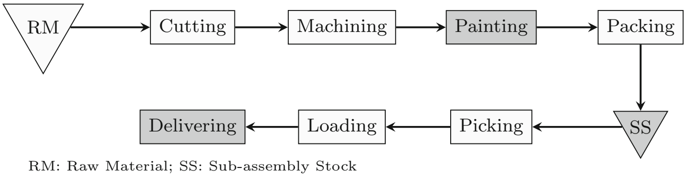

Fig. 1 Illustration of the integrated furniture production and distribution process

with one production line and only one vehicle. The production line refers only to the painting line and, therefore we need to define production lot-sizes and inventories for each period, that is, the quantities of parts and sub-assemblies to be painted and kept in stock in order to satisfy the demand of final products. Moreover, we also need to define the different routes to deliver the final products to a set of retail stores, taking into account time windows and due dates on customer deliveries.

Figure 1 shows the different stages of the furniture production and distribution process. Boxes in a darker color highlight the operations we integrated into a single framework as they represent the most important decisions to be made.

4 Mixed integer programming model

In this section, we present a MIP model to represent the scenario described above (Sect. 3). For the lot-sizing decisions, we used a discrete time framework that divides the planning horizon into 1-day periods. We relied on the multi-item capacitated lot-sizing problem (Jans and Degraeve 2008; Quadt and Kuhn 2007) to model these decisions.

For the vehicle routing decisions, we used a continuous time framework that enabled us to determine the timing of each route (i.e., the time each route began and ended, as well as the time customers were served). We modeled the routing part of our problem based on the work of Azi et al. (2007). We introduced a new set of binary variables to save the periods in which each route began and ended, respectively. These variables were useful to determine the shipment quantities of each item in each period, which were then used to couple the two sub-models through inventory flow constraints.

Note that the structure of our model is different from the one usually found in the literature. Most models, such as those proposed by Fumero and Vercellis (1999), Bard and Nananukul (2009b), Armentano et al. (2011), Adulyasak et al. (2014a), etc., assume that each vehicle performs at most one route per period, usually a day, and each route fits within a period. Each vehicle may be used more than once over the planning horizon, but it is limited to be used at most once per day.

In our model, however, routes may extend over one or more days, and therefore the vehicle may not return to the depot in the same period it departs from. Therefore, a given route does not necessarily fit within a single period. Besides, the vehicle might be used more than once over the planning horizon, but it is not limited to be only once per day. Some routes may be short enough to fit into a period, and therefore the vehicle is allowed to start a new route within the same period.

Before describing the MIP model, let us introduce the following additional notation:

Parameters

λ : Time to load/unload per unit of weight;
s_i : Service time of customer i. $s_i = \lambda \sum_{p\in\mathcal{P}} \varphi_p d_{pi}$;
M_{0j} : A sufficiently large number. $M_{0j} = \bar{\delta}_{0|\mathcal{T}|} + \min\{\lambda\theta, \sum_{i\in\bar{\mathcal{C}}} s_i\} + \tau_{0j}, \quad j \in \bar{\mathcal{C}}$;
M_{ij} : A sufficiently large number. $M_{ij} = \Delta_i + s_i + \tau_{ij}, \quad i \in \bar{\mathcal{C}}, j \in \{\bar{\mathcal{C}}, n+1\}$.

Lot-sizing variables

x_{ct} : Production quantity of item c in period t;
I_{ct} : Inventory of item c at the end of period t;
y_{ct} : Equal to 1 if there is production of item c in period t, 0 otherwise;

Routing variables

w_{ijr} : Equal to 1 if route r travels directly from node i to node j, 0 otherwise;
Q_{prt} : Quantity of product p shipped on route r in period t;
ϕ_{irt} : Equal to 1 if node i is visited by route r in period t, 0 otherwise;
μ_{ir} : Starting time at which node i is served by route r.

The objective function (1) minimizes the manufacturer's setups, inventory and routing costs:

$$\min \sum_{c\in\mathcal{C}} \sum_{t\in\mathcal{T}} f_c y_{ct} + \sum_{c\in\mathcal{C}} \sum_{t\in\mathcal{T}} h_c I_{ct} + \sum_{i\in\{0,\bar{\mathcal{C}}\}} \sum_{j\in\{\bar{\mathcal{C}},n+1\}} \sum_{r=1}^{R} c_{ij} w_{ijr}. \tag{1}$$

Constraints (2) establish the inventory balance of each item c in period t. Note that these constraints take into account the quantity of product p shipped on routes starting in period t to calculate the total quantity of item c shipped in period t:

$$I_{c,t-1} + x_{ct} = \sum_{r=1}^{R} \sum_{p\in\mathcal{P}} \eta_{cp} Q_{prt} + I_{ct}, \quad c \in \mathcal{C}, t \in \mathcal{T}. \tag{2}$$

The set of constraints (3) ensures that the quantity of item c shipped in period t is not greater than the inventory at the end of period $t-1$. It means that the production in period t cannot be used immediately, but from the next period $t+1$:

$$I_{c,t-1} \geq \sum_{r=1}^{R} \sum_{p\in\mathcal{P}} \eta_{cp} Q_{prt}, \quad c \in \mathcal{C}, t \in \mathcal{T}. \tag{3}$$

Inequalities (4) impose a minimum inventory level at the end of each period and (5) represents the production capacity constraints:

$$I_{ct} \geq I_c^{\min}, \quad c \in \mathcal{C}, t \in \mathcal{T}, \tag{4}$$

$$\sum_{c\in\mathcal{C}} \rho_c x_{ct} \leq K_t, \quad t \in \mathcal{T}. \tag{5}$$

The fixed setup cost of item c is considered in the objective function only if there is production of this item, constraints (6). The upper bound on the production quantity may be calculated as $M_{ct} = \min\left(\lfloor \frac{K_t}{\rho_c} \rfloor, \sum_{i\in\bar{\mathcal{C}}} \sum_{p\in\mathcal{P}} \eta_{cp} d_{pi}\right)$:

$$x_{ct} \le M_{ct} y_{ct}, \quad c \in \mathcal{C}, t \in \mathcal{T}. \tag{6}$$

Equalities (7) and (8) ensure that each route begins and ends at the depot, respectively. Note that the depot is represented by nodes 0 and $n+1$, so that a route starts departing from node 0 and finishes arriving at node $n+1$. We assume that a route traveling directly from node 0 to node $n+1$ is an empty route:

$$\sum_{j\in\{\bar{\mathcal{C}},n+1\}} w_{0jr} = 1, \quad r = 1,\ldots,R, \tag{7}$$

$$\sum_{i\in\{0,\bar{\mathcal{C}}\}} w_{i(n+1)r} = 1, \quad r = 1,\ldots,R. \tag{8}$$

Flow conservation is ensured by constraints (9). Note that it is not possible to arrive at node 0 and depart from node $n+1$, even though both of them represent the same location (i.e., the depot):

$$\sum_{\substack{j\in\{\bar{\mathcal{C}},n+1\}\\ j\neq i}} w_{ijr} = \sum_{\substack{j\in\{0,\bar{\mathcal{C}}\}\\ j\neq i}} w_{jir}, \quad i \in \bar{\mathcal{C}}, r = 1,\ldots,R. \tag{9}$$

Equation (10) establishes that each customer must be visited once over the planning horizon, i.e., split deliveries are not allowed:

$$\sum_{\substack{i\in\{0,\bar{\mathcal{C}}\}\\ i\neq j}} \sum_{r=1}^{R} w_{ijr} = 1, \quad j \in \bar{\mathcal{C}}. \tag{10}$$

Precedences between routes to reduce symmetry in the solution space are imposed by constraints (11). These inequalities indicate that route $r+1$ can be used only if route r is used. Therefore, these constraints force empty routes to be the last ones:

$$\sum_{i\in\bar{\mathcal{C}}} w_{i(n+1)r} \ge \sum_{i\in\bar{\mathcal{C}}} w_{0i(r+1)}, \quad r = 1,\ldots,R-1. \tag{11}$$

Constraints (12) set the quantity of product p shipped on route r in period t to zero if route r does not start in period t:

$$Q_{prt} \le \min\left\{\left\lfloor \frac{\theta}{\varphi_p} \right\rfloor, \sum_{i\in\bar{\mathcal{C}}} d_{pi}\right\} \phi_{0rt}, \quad p \in \mathcal{P},\ r = 1,\ldots,R,\ t \in \mathcal{T}. \tag{12}$$

Equalities (13) determine the total quantity of product p shipped on route r. This value corresponds to the sum of the demands of the customers served by route r. Since any non-empty route satisfies $\sum_{t\in\mathcal{T}} \phi_{0rt} = 1$, and taking into account (12), it follows that at most one variable Q_{prt} on the left-hand side summation of (13) can take a positive value:

$$\sum_{t\in\mathcal{T}} Q_{prt} = \sum_{i\in\bar{\mathcal{C}}} d_{pi} \sum_{\substack{j\in\{\bar{\mathcal{C}},n+1\}\\ j\neq i}} w_{ijr}, \quad p \in \mathcal{P},\ r = 1,\ldots,R. \tag{13}$$

Constraints (14) guarantee that route r visits customer i in just one period t, only if customer i belongs to route r:

$$\sum_{t\in\mathcal{T}} \phi_{irt} = \sum_{\substack{j\in\{\bar{\mathcal{C}},n+1\}\\ j\neq i}} w_{ijr}, \quad i \in \bar{\mathcal{C}},\ r = 1,\ldots,R. \tag{14}$$

The sets of constraints (15) and (16) ensure that each non-empty route starts and finishes in just one period (not necessarily the same), respectively. Note that $w_{0(n+1)r} = 1$ implies that route r was not used, therefore $\phi_{0rt} = \phi_{(n+1)rt} = 0$ for all $t \in \mathcal{T}$:

$$\sum_{t\in\mathcal{T}} \phi_{0rt} = 1 - w_{0(n+1)r}, \quad r = 1,\ldots,R, \tag{15}$$

$$\sum_{t\in\mathcal{T}} \phi_{0rt} = \sum_{t\in\mathcal{T}} \phi_{(n+1)rt}, \quad r = 1,\ldots,R. \tag{16}$$

We do not impose any limit on the duration of the routes. Therefore, routes can take any time, as long as the vehicle returns to the depot before the end of the planning horizon ($\bar{\delta}_{(n+1)|\mathcal{T}|}$) and all the customers' time windows and deadlines are respected. To do so, we keep track of the periods when each route departs from and returns to the depot (ϕ_{0rt} and $\phi_{(n+1)rt}$, respectively), as well as the time instant when each node is served (μ_{ir}), through constraints (17)–(21).

Equation (17) represents the time windows constraints. These constraints are active only if node i is visited by route r in period t, i.e., $\phi_{irt} = 1$. Since each customer is visited in just one period, it follows that each customer is served in just one time window. If node i is a customer, then $\tilde{M}_i = \Delta_i$, otherwise $\tilde{M}_i = \bar{\delta}_{i|\mathcal{T}|}$:

$$\delta_{it} - \delta_{it}(1-\phi_{irt}) \le \mu_{ir} \le \bar{\delta}_{it} + (\tilde{M}_i - \bar{\delta}_{it})(1-\phi_{irt}), \quad i \in \mathcal{N}, t \in \mathcal{T},\ r = 1,\ldots,R. \tag{17}$$

Lower bounds on the time instant when node j is served by route r are imposed by inequalities (18) and (19). Constraints (18) are active only for the first customer served by route r, and consider the time required to load the vehicle at the depot. For the remaining nodes j, including node $n+1$, constraints (19) consider the unloading time at the customer previously served by the route, i:

$$\mu_{jr} \geq \mu_{0r} + \sum_{i \in \bar{\mathcal{C}}} s_i \sum_{\substack{k \in \{\bar{\mathcal{C}}, n+1\} \\ k \neq i}} w_{ikr} + \tau_{0j} - M_{0j}(1 - w_{0jr}), \quad j \in \bar{\mathcal{C}}, r = 1, \ldots, R, \tag{18}$$

$$\mu_{jr} \geq \mu_{ir} + s_i + \tau_{ij} - M_{ij}(1 - w_{ijr}), \quad \begin{array}{l} i \in \bar{\mathcal{C}},\ j \in \{\bar{\mathcal{C}}, n+1\}, \\ r = 1, \ldots, R : i \neq j. \end{array} \tag{19}$$

Each customer i must be served before its due date, as stated by constraints (20). Observe that μ_{ir} is set to zero if route r does not visit customer i. Since we are dealing with a single vehicle case, routes cannot overlap in time. Therefore, constraints (21) ensure that route $r+1$ can only depart from the depot after route r has ended:

$$\mu_{ir} \leq \Delta_i \sum_{\substack{j \in \{\bar{\mathcal{C}}, n+1\} \\ j \neq i}} w_{ijr}, \quad i \in \bar{\mathcal{C}},\ r = 1, \ldots, R, \tag{20}$$

$$\mu_{0(r+1)} \geq \mu_{(n+1)r}, \quad r = 1, \ldots, R-1. \tag{21}$$

Constraints (22) ensure that the vehicle capacity is not exceeded:

$$\sum_{p \in \mathcal{P}} \varphi_p \sum_{i \in \bar{\mathcal{C}}} d_{pi} \sum_{\substack{j=1 \\ j \neq i}}^{n+1} w_{ijr} \leq \theta, \quad r = 1, \ldots, R. \tag{22}$$

Finally, the domain of the variables is given by (23)–(28):

$$x_{ct}, I_{ct} \geq 0, \quad c \in \mathcal{C},\ t \in \mathcal{T}, \tag{23}$$

$$y_{ct} \in \{0, 1\}, \quad c \in \mathcal{C},\ t \in \mathcal{T}, \tag{24}$$

$$Q_{prt} \geq 0, \quad p \in \mathcal{P}, r = 1, \ldots, R,\ t \in \mathcal{T}, \tag{25}$$

$$\phi_{irt} \in \{0, 1\}, \quad i \in \mathcal{N}, r = 1, \ldots, R,\ t \in \mathcal{T}, \tag{26}$$

$$w_{ijr} \in \{0, 1\}, \quad i, j \in \mathcal{N},\ r = 1, \ldots, R, \tag{27}$$

$$\mu_{ir} \geq 0, \quad i \in \mathcal{N},\ r = 1, \ldots, R. \tag{28}$$

It is worth mentioning that we also implemented two alternative formulations by redefining the Q variables. In the first one, instead of the non-negative continuous variable Q_{prt}, we defined Q_{irt} as a binary variable, which is equal to one if the demand of node i is served by route r that starts in period t, 0 otherwise. Thus, $Q_{irt} = 1$ means that route r serves node i and departs from the depot in period t. Observe that Q_{prt} corresponds to $\sum_{i \in \bar{\mathcal{C}}} d_{pi} Q_{irt}$ and therefore, some adjustments should be made in the model (1)–(28).

The expression above should be replaced in constraints (2) and (3), and some straightforward modifications in constraints (12) and (13) are required in order to establish the proper relationship between the binary variables Q_{irt}, ϕ_{0rt} and w_{ijr}, respectively. These adjustments result in the following sets of constraints:

$$I_{c,t-1} + x_{ct} = \sum_{r=1}^{R} \sum_{p \in \mathcal{P}} \eta_{cp} \sum_{i \in \bar{\mathcal{C}}} d_{pi} Q_{irt} + I_{ct}, \quad c \in \mathcal{C},\ t \in \mathcal{T}, \tag{29}$$

$$I_{c,t-1} \geq \sum_{r=1}^{R} \sum_{p \in \mathcal{P}} \eta_{cp} \sum_{i \in \bar{\mathcal{C}}} d_{pi} Q_{irt}, \quad c \in \mathcal{C},\ t \in \mathcal{T}, \tag{30}$$

$$\sum_{i \in \bar{\mathcal{C}}} Q_{irt} \leq |\mathcal{N}| \phi_{0rt}, \quad t \in \mathcal{T},\ r = 1, \ldots, R, \tag{31}$$

$$\sum_{t \in \mathcal{T}} Q_{irt} = \sum_{\substack{j \in \{\bar{\mathcal{C}}, n+1\} \\ j \neq i}} w_{ijr}, \quad i \in \bar{\mathcal{C}},\ r = 1, \ldots, R, \tag{32}$$

$$Q_{irt} \in \{0, 1\}, \quad i \in \bar{\mathcal{C}},\ r = 1, \ldots, R,\ t \in \mathcal{T}. \tag{33}$$

Therefore, the first alternative formulation corresponds to minimize (1), subject to (4)–(11), (14)–(24), (26)–(33).

In the second alternative formulation, we define Q_{it} as a binary variable, which is equal to one if there is a shipment to node i in period t, 0 otherwise. Thus, Q_{it} is equal to 1 only if some route r starts in period t and node i is served by r. Here, Q_{it} corresponds to $\sum_{r=1}^{R} Q_{irt}$ and therefore, constraints (2) and (3) should be modified accordingly:

$$I_{c,t-1} + x_{ct} = \sum_{p \in \mathcal{P}} \eta_{cp} \sum_{i \in \bar{\mathcal{C}}} d_{pi} Q_{it} + I_{ct}, \quad c \in \mathcal{C},\ t \in \mathcal{T}, \tag{34}$$

$$I_{c,t-1} \geq \sum_{p \in \mathcal{P}} \eta_{cp} \sum_{i \in \bar{\mathcal{C}}} d_{pi} Q_{it}, \quad c \in \mathcal{C},\ t \in \mathcal{T}. \tag{35}$$

The new set of constraints (36) must be added to the formulation in order to avoid splitting deliveries. Constraints (12) and (13) are both replaced by the inequalities (37), which assure that a shipment to customer i in period t only takes place if a route r starts in period t (i.e., $\phi_{0rt} = 1$) and customer i is served by route r (i.e., $\sum_{\substack{j=1 \\ j \neq i}}^{n+1} w_{ijr} = 1$):

$$\sum_{t \in \mathcal{T}} Q_{it} = 1, \quad i \in \bar{\mathcal{C}}, \tag{36}$$

$$Q_{it} \geq \phi_{0rt} + \sum_{\substack{j=1 \\ j \neq i}}^{n+1} w_{ijr} - 1, \quad i \in \bar{\mathcal{C}},\ r = 1, \ldots, R,\ t \in \mathcal{T}, \tag{37}$$

$$Q_{it} \in \{0, 1\}, \quad i \in \bar{\mathcal{C}},\ t \in \mathcal{T}. \tag{38}$$

The second alternative formulation reads as follows: minimize (1) subject to (4)–(11), (14)–(24), (26)–(28), (34)–(38).

Preliminary computational tests showed that model (1)–(28) outperformed both of these alternative formulations, as it returned better solutions in less CPU time.

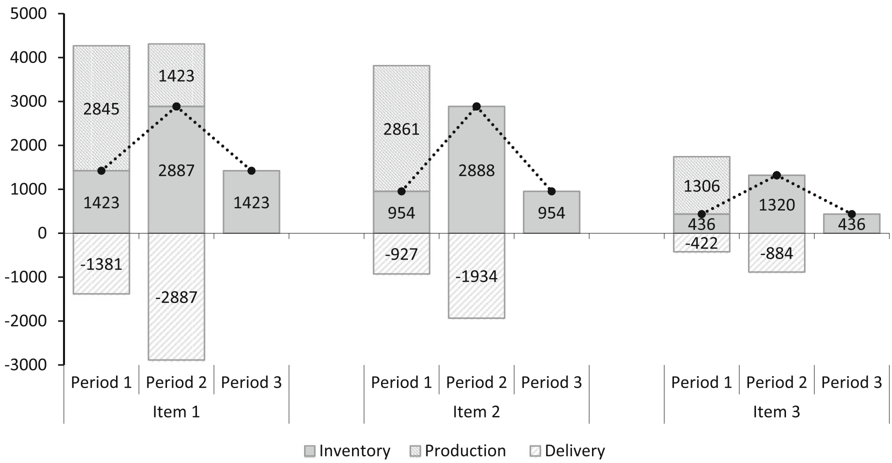

Fig. 2 Production plan of the illustrative example

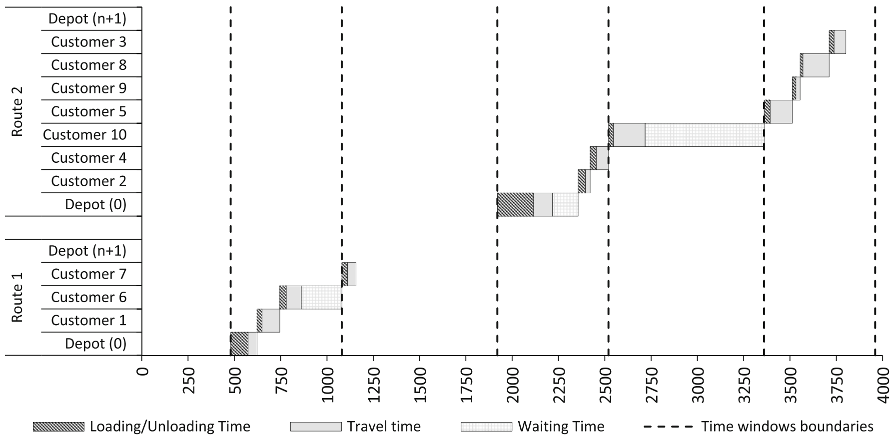

Fig. 3 Routing plan of the illustrative example

4.1 Illustrative example

In order to illustrate some key features of the model, Figs. 2 and 3 show a solution of an illustrative example with $n = 10$, $|\mathcal{P}| = 2$, $|\mathcal{C}| = 3$ and $|\mathcal{T}| = 3$. During each period, corresponding to 1 day, the vehicle is always available to travel and/or wait at different locations (customers or depot). However, loading/unloading operations can only take place within the preset time windows. The remaining parameters were generated based on Table 3 (see Sect. 6.1).

Figure 2 shows, for each item, inventory levels, lot-sizes and shipment quantities for each period. For example, for item one, the inventory at the beginning of period one is 1423 units. The lot-size and the shipment quantity in this period are 2845 and 1381 units, respectively. Thus, the inventory level of item one at the end of period one is $1423 + 2845 - 1381 = 2887$ units, which corresponds to the initial inventory of item one in period two. By following the same calculation, we can obtain the inventory levels, lot-sizes and shipment quantities for the remaining periods. Note that there is no production or shipments of item one in period three, and therefore the initial and final inventory levels are the same.

Figure 3 shows the distribution routes used over the planning horizon (3 days). The vertical dashed lines represent the time windows for each day. We assumed that all the customers have the same time window in each period, corresponding to regular business hours (from 8:00 a.m. to 6:00 p.m. every day). The depot is open 24 h, but loading operations must occur during the regular working hours. For this reason, in Fig. 3, there is no activity during the first 8 h (480 min).

The first route starts immediately after the time window in period 1 opens and visits customers 1, 6 and 7 before returning to the depot, still in period 1. Note that customer 7 is served right at the end of the time window. When route 1 finishes, the depot is not available for loading operations, and therefore the vehicle should stay at the depot until the next time window opens. Route 2 starts in period 2, and serves customers 2, 4, 10, 5, 9, 8 and 3, before returning to the depot in period 3. Therefore, route 2 extends over more than one period. After serving customer 10, at the end of the time window of period 2, the vehicle travels to customer 5, where it has to wait until the time window of period 3 opens. Thus, customers 5, 9, 8 and 3 are served in period 3.

5 Solution methods

In this section, we propose different relax-and-fix strategies (RF) to solve the MIP model presented in Sect. 4. Each RF heuristic uses a different criterion for partitioning and fixing variables. The idea is to explore the structure of the model, which includes several sets of integer variables, to obtain various sub-models that may lead to different feasible
solutions.

5.1 Relax-and-fix heuristics

The RF heuristic, described by Dillenberger et al. (1994), is a decomposition algorithm widely and most notably used in lot-sizing and scheduling problems (Araujo et al. 2007; Beraldi et al. 2008; Ferreira et al. 2009, 2010; James and Almada-Lobo 2011).

In this procedure, the set of integer variables is partitioned into disjunctive sets according to a given criterion. At each iteration, variables of only one of these sets are defined as integers, while the remaining integer variables are relaxed. The resulting sub-model is then solved. If a feasible solution is obtained, then the integer variables of the sub-model, or part of them, are fixed to their current values and the process is repeated for all the remaining sets.

Besides the variable set partition, a criterion should be defined to fix variables at each iteration. The main feature of this procedure is that we solve smaller, and possibly

Table 2 Relax-and-fix strategies

Strategy	Partition	Fixing
RFX1	Period	y_{ct}
RFX2	Period	y_{ct} if $x_{ct} > 0$; w_{ijr} if $\phi_{0rt} = 1$
RFX3	Period	y_{ct} if $x_{ct} > 0$; w_{ijr}, ϕ_{0rt} if $\phi_{0rt} = 1$
RFX4	Route (w_{ijr})	w_{ijr}
RFX5	Lot-sizing then routing	y_{ct}
RFX6	Routing then lot-sizing	w_{ijr}, ϕ_{irt}

easier problems than the original one. However, we observed that the RF heuristic does not guarantee that it will find a feasible solution, even if one exists.

The model proposed in Sect. 4 has various possibilities to build different subsets of binary variables. In the lot-sizing part, the setup variables (y_{ct}) are indexed by item and by period. In the routing part, the vehicle flow variables (w_{ijr}) and the period visit variables (ϕ_{irt}) are indexed by nodes, routes and periods. Some of these indices were used to define strategies for partitioning variables. Moreover, different criteria for fixing variables were implemented. Table 2 shows the six different RF strategies proposed in this paper.

The RFX1–RFX3 strategies use the classic criterion of partitioning variables per time periods. They differ from each other by the criterion used to fix variables at each iteration. RFX1 fixes the binary variable y_{ct}, regardless of whether there is production of item c in period t. The RFX2 strategy fixes y_{ct} to one, if there is production of item c in period t (i.e., if $x_{ct} > 0$), and w_{ijr} to one, if route r starts in period t (i.e., if $\phi_{0rt} = 1$). This criterion for fixing variables is more flexible, since less variables are fixed at each iteration and some decisions may be re-evaluated at later iterations. As an alternative, the RFX3 strategy also fixes the binary variable ϕ_{0rt} if it is equal to 1. Observe that RFX2 never fixes ϕ_{0rt}, so that at each iteration it is possible to redefine the period when route r departs from the depot. In RFX3, on the other hand, this re-definition is not possible because ϕ_{0rt} is fixed as soon as it is equal to one.

In RFX4, we relaxed the integrality constraint on the w_{ijr} variables and kept ϕ_{irt} as binary variables. In order to gain some flexibility, the ϕ_{irt} variables are not fixed and, therefore, they can be updated at each iteration. In other words, once route r is determined, the order of the visits cannot be changed, but only the periods when the customers are visited. In some preliminary tests, we also tried relaxing/fixing the ϕ_{irt} variables, so that less binary variables need to be determined at each iteration. However, this strategy led to poor results as the heuristic failed to return feasible solutions for many instances.

Strategies RFX5 and RFX6 are based on the idea of decomposing the problem into two parts. In RFX5, we first relaxed all the routing variables and then solved the remaining lot-sizing subproblem. If a feasible solution is found, the setup variables (y_{ct}) are fixed to their current values, and the routing subproblem is then solved with integrality constraints. Observe that we gained some flexibility by fixing only the setup variables, as the remaining production decisions may be still re-evaluated in the routing phase.

In RFX6 we followed the opposite path to RFX5. First, we relaxed all the lot-sizing variables and then solved the remaining routing subproblem. If a feasible solution is found, the binary variables w_{ijr} and ϕ_{irt} are fixed to their current values, and the lot-sizing subproblem is then solved with integrality constraints. Similarly to RFX5, in RFX6 we allowed some of the routing decisions to be updated when solving the lot-sizing subproblem.

Preliminary tests showed that when RFX5 and RFX6 could not modify previous decisions, it was difficult to find feasible solutions. This is because the subproblem solved at the first stage is myopic; meanwhile the one solved at the second stage cannot reset any decision to achieve feasibility. Therefore, the idea of letting the heuristics update some decisions at the second stage aims to overcome this issue.

6 Computational tests

In this section, we present the computational results of CPLEX and the RF strategies on the set of randomly generated instances described in Sect. 6.1. The MIP model and the RF algorithms were coded in AMPL and solved by CPLEX 12.6 on a personal computer with an Intel Core i7 3.40 GHz processor and 16 GB of RAM. At each iteration of the RF heuristics, a MIP needs to be solved. Although smaller than the full integrated model, these sub-models may be still difficult to solve. Therefore, if they are not solved to optimality in a pre-specified amount of time, their execution is stopped and the best solution found so far is analyzed. A computational time limit of 1 h is arbitrarily set for each instance. This limit seems to be acceptable to support decisions in practical settings of small furniture companies.

It is worth mentioning that we performed extensive preliminary experiments with different CPLEX parameters in order to identify choices that may improve the solver performance. We tested different selection nodes and variables strategies, as well as analyzing the impact of using cutting planes and some primal heuristics, namely RINS, feasibility pump and local branching heuristics. The tests were conducted using a different set of randomly generated instances, which consist of problems with three products, three items, five periods and $n \in \{5, 10, 15\}$. For each value of n, ten instances were generated following the patterns shown in Table 3, totaling 30 instances. Based on the results, we decided to use the best estimate search as a strategy to select nodes, strong branching to select variables and set on the local branching heuristic. All the remaining parameters were set up to their default values. Details on the meaning of these and other CPLEX parameters are found in the CPLEX user's manual (IBM ILOG CPLEX 2013).

6.1 Instances generation

We generated a set of random instances using some information provided by a Brazilian furniture company. However, part of the data were randomly generated based on the

Table 3 Parameters used to generate random instances

Units of item c required per unit of product p	$\eta_{cp} \in U[0, 5]$
Width (W_c) and length (L_c) of item c	$W_c, L_c \in U[20, 50]$
Unit inventory holding cost of item c	$h_c = 0.001 \cdot W_c \cdot L_c$
Fixed production setup cost of item c	$f_c = 1000 \cdot h_c$
Unit processing time of item c	$\rho_c = \frac{W_c \cdot L_c}{2500}$
Demand of product p at customer i	$d_{pi} \in U[10, 100]$
Production capacity in period t	$K_t = 3.5 \left\lceil \frac{\sum_{c\in\mathcal{C}} \rho_c \left(\sum_{p\in\mathcal{P}} \sum_{i\in\mathcal{N}} d_{pi}\eta_{cp}\right)}{\lvert\mathcal{T}\rvert} \right\rceil$
Initial (minimum) inventory of item c	$I_{c0} = I_c^{\min} = \left\lceil \frac{\left(\sum_{p\in\mathcal{P}} \sum_{i\in\mathcal{N}} d_{pi}\eta_{cp}\right)}{\lvert\mathcal{T}\rvert} \right\rceil$
Coordinates of node i	$(X_i, Y_i) \in [0, 500]$
Travel time from node i to node j	$\tau_{ij} = \left\lceil \frac{60}{100} \left(\sqrt{\left(X_i - X_j\right)^2 + \left(Y_i - Y_j\right)^2} \right) \right\rceil$
Travel cost from node i to node j	$c_{ij} = \left\lfloor \sqrt{\left(X_i - X_j\right)^2 + \left(Y_i - Y_j\right)^2} \right\rfloor$
Unit weight of product p	$\varphi_p = \left\lceil 0.001 \cdot \sum_{c\in\mathcal{C}} W_c \cdot L_c \cdot \eta_{cp} \right\rceil$
Vehicle capacity	$\theta = \sum_{p\in\mathcal{P}} \sum_{i\in\mathcal{N}} d_{pi} \cdot \varphi_p$
Time window of node $i \in \{0, \bar{\mathcal{C}}\}$ in period t	$\left[\delta_{it}, \bar{\delta}_{it}\right] = [480 + 1440\,(t-1)\,,\, 1080 + 1440\,(t-1)]$
Time window of node $n+1$ in period t	$\left[\delta_{(n+1)t}, \bar{\delta}_{(n+1)t}\right] = [1440\,(t-1)\,,\, 1440t]$
Time to load/unload per unit of weight	$\lambda = 1/50$
Due date of customer $i \in \bar{\mathcal{C}}$	$\Delta_i = \bar{\delta}_{i\lvert\mathcal{T}\rvert}$
Maximum number of routes over the planning horizon	$R = n$

literature practice. We noticed that the size of the instances is already realistic in the context of small furniture companies, which produce a few final products to meet the demand of a set of customers over a short planning horizon (usually a week).

We relied on the procedures proposed by Gramani et al. (2009) for a cutting stock problem in furniture companies, as well as Armentano et al. (2011) for a multi-product PIDRP. The number of final products ($|\mathcal{P}|$), items ($|\mathcal{C}|$) and customers (n) are $\{2, 3\}$, $\{3, 5\}$ and $\{5, 10, 15\}$, respectively. The planning horizon is fixed to five periods for all the instances. For each combination of final products, items and customers, we generated five instances, totaling 60 instances. Table 3 shows how the remaining parameters were generated.

Notice that we defined width and length for each item c in order to estimate inventory holding costs, fixed setup costs, unit processing times and unit weights as a function of the size of the items. It means that the larger the items, the more expensive and heavier they are. Factor 3.5, used to estimate the production capacity, adjusts the capacity tightness as in Trigeiro et al. (1989), Toledo and Armentano (2006) and Armentano et al. (2011). Regarding the time windows, we adopted a common practice in the Brazilian furniture industry: customers may be visited at any moment within the regular business hours. Similarly, the vehicle must leave the depot within the regular working hours, but it may return at any time. Furthermore, as we considered a relatively short planning horizon (5 days), we supposed that the customers' due dates matched the end of the last time window. The time to load/unload per unit of weight was estimated as 1/50, which means that 50 kg is loaded/unloaded per minute. Finally, we assumed that the maximum number of routes over the planning horizon is equal to the number of customers, n.

6.2 Numerical results

Table 4 shows the results obtained using CPLEX. Columns 2–5 show the number of customers, products, items and periods, respectively. Column 6 shows the total cost. Columns 7 and 8 show the optimality gap, in percentage, and the CPU time, in seconds, respectively. Finally, columns 9–11 indicate the number of constraints, total and binary variables, respectively. The symbol "–" marks instances where CPLEX failed to return a feasible solution. The optimality gap corresponds to the relative difference between the best upper bound (UB) and the best lower bound (LB) found by CPLEX after 1 h of computing time, and is calculated as follows:

$$\text{Gap}\,(\%) = \left(\frac{\text{UB} - \text{LB}}{\text{UB}}\right) \times 100. \tag{39}$$

As observed, CPLEX solved to optimality all the instances with five customers ($n = 5$) in short computational times. However, for ten customers ($n = 10$), CPLEX only proved the optimality of one instance. For the remaining 19 instances, it only returned feasible solutions within the preset computational time limit. The optimality gaps were small, except for instances 26, 28, 32 and 33, whereby the optimality gap was more than 5%.

As expected, the most difficult instances were those with 15 customers ($n = 15$). In this group, CPLEX solved 17 out of 20 instances, but none of them to optimality. For instances 56, 57 and 59, CPLEX did not find a feasible solution within the computational time limit. The average optimality gap for this group (5.42%) is larger than the one for $n = 10$ (3.46%); however, some near-optimal solutions were obtained within 1 h of CPU time.

In order to show the importance of integrating production and distribution decisions, we compared our model with a sequential approach that mimics real practice in furniture companies. The routing schedule (routes and timing) is normally decided first with the aim of minimizing only the routing costs. Then, once the routing decisions are made, the amount of final products carried in each route is used by the production planner to calculate the demand of each item in each period. These demands are then used as input to determine a production plan which minimizes setups and inventory holding costs over the planning horizon. Therefore, first we solved a multi-trip traveling salesman problem (multi-TSP) in order to obtain the route schedules, and then proceeded to solve a multi-item lot-sizing problem to determine lot-sizes and inventory levels.

Table 4 Result of the CPLEX solver: 1 h CPU time

Instance	n	$\lvert\mathcal{P}\rvert$	$\lvert\mathcal{C}\rvert$	$\lvert\mathcal{T}\rvert$	Total cost	Gap (%)	CPU (s)	#Constraints	#Variables	#Binaries
1	5	2	3	5	12,382.36	0.00	22.72	741	553	345
2	5	2	3	5	16,540.51	0.00	5.60	741	553	345
3	5	2	3	5	14,848.74	0.01	8.52	741	553	345
4	5	2	3	5	18,313.76	0.00	10.59	741	553	345
5	5	2	3	5	22,356.50	0.01	11.40	741	553	345
6	5	2	5	5	30,410.35	0.01	15.78	783	585	355
7	5	2	5	5	27,106.13	0.00	52.57	783	585	355
8	5	2	5	5	28,632.54	0.00	12.97	783	585	355
9	5	2	5	5	36,794.33	0.01	130.29	783	585	355
10	5	2	5	5	44,480.04	0.01	225.90	783	585	355
11	5	3	3	5	32,251.10	0.01	62.79	771	578	345
12	5	3	3	5	28,966.26	0.01	57.03	771	578	345
13	5	3	3	5	25,535.43	0.00	13.15	771	578	345
14	5	3	3	5	28,523.15	0.01	55.66	771	578	345
15	5	3	3	5	21,009.33	0.01	41.61	771	578	345
16	5	3	5	5	47,639.32	0.01	111.01	813	610	355
17	5	3	5	5	35,868.20	0.01	13.79	813	610	355
18	5	3	5	5	46,326.40	0.01	101.07	813	610	355
19	5	3	5	5	40,218.10	0.01	83.99	813	610	355
20	5	3	5	5	28,536.27	0.01	66.94	813	610	355

Table 4 continued

Instance	n	$\|\mathcal{P}\|$	$\|\mathcal{C}\|$	$\|\mathcal{T}\|$	Total cost	Gap (%)	CPU (s)	#Constraints	#Variables	#Binaries
21	10	2	3	5	32,472.89	1.44	3600.00	2866	2308	1725
22	10	2	3	5	34,385.96	1.86	3600.00	2866	2308	1725
23	10	2	3	5	39,592.76	2.00	3600.00	2866	2308	1725
24	10	2	3	5	29,437.46	0.01	2547.53	2866	2308	1725
25	10	2	3	5	31,008.98	1.99	3600.00	2866	2308	1725
26	10	2	5	5	44,665.86	6.84	3600.00	2908	2340	1735
27	10	2	5	5	47,533.48	2.67	3600.00	2908	2340	1735
28	10	2	5	5	41,129.61	7.43	3600.00	2908	2340	1735
29	10	2	5	5	42,741.94	1.74	3600.00	2908	2340	1735
30	10	2	5	5	52,878.70	1.95	3600.00	2908	2340	1735
31	10	3	3	5	51,793.46	2.60	3600.00	2926	2358	1725
32	10	3	3	5	40,027.89	6.62	3600.00	2926	2358	1725
33	10	3	3	5	27,563.59	8.98	3600.00	2926	2358	1725
34	10	3	3	5	42,281.83	3.39	3600.00	2926	2358	1725
35	10	3	3	5	39,922.32	3.47	3600.00	2926	2358	1725
36	10	3	5	5	54,605.62	3.42	3600.00	2968	2390	1735
37	10	3	5	5	57,149.86	0.70	3600.00	2968	2390	1735
38	10	3	5	5	44,365.74	4.53	3600.00	2968	2390	1735
39	10	3	5	5	68,986.06	2.83	3600.00	2968	2390	1735
40	10	3	5	5	58,064.13	4.65	3600.00	2968	2390	1735

Table 4 continued

Instance	n	$\|\mathcal{P}\|$	$\|\mathcal{C}\|$	$\|\mathcal{T}\|$	Total cost	Gap (%)	CPU (s)	#Constraints	#Variables	#Binaries
41	15	2	3	5	57,212.94	4.21	3600.00	7191	6063	4905
42	15	2	3	5	48,753.51	4.25	3600.00	7191	6063	4905
43	15	2	3	5	43,428.13	3.53	3600.00	7191	6063	4905
44	15	2	3	5	45,299.84	9.97	3600.00	7191	6063	4905
45	15	2	3	5	41,302.28	1.87	3600.00	7191	6063	4905
46	15	2	5	5	76,015.51	5.06	3600.00	7233	6095	4915
47	15	2	5	5	95,251.30	4.88	3600.00	7233	6095	4915
48	15	2	5	5	65,589.68	4.01	3600.00	7233	6095	4915
49	15	2	5	5	63,170.14	1.59	3600.00	7233	6095	4915
50	15	2	5	5	76,022.14	4.82	3600.00	7233	6095	4915
51	15	3	3	5	59,550.04	9.53	3600.00	7281	6138	4905
52	15	3	3	5	49,086.32	8.12	3600.00	7281	6138	4905
53	15	3	3	5	63,638.54	5.72	3600.00	7281	6138	4905
54	15	3	3	5	31,820.79	7.12	3600.00	7281	6138	4905
55	15	3	3	5	32,068.38	4.89	3600.00	7281	6138	4905
56	15	3	5	5	–	–	3600.00	7323	6170	4915
57	15	3	5	5	–	–	3600.00	7323	6170	4915
58	15	3	5	5	81,438.07	7.66	3600.00	7323	6170	4915
59	15	3	5	5	–	–	3600.00	7323	6170	4915
60	15	3	5	5	71,353.81	5.46	3600.00	7323	6170	4915

Table 5 Benefits of integrating production and routing decisions: $n = 5$ customers

Instance	n	$\|\mathcal{P}\|$	$\|\mathcal{C}\|$	$\|\mathcal{T}\|$	Sequential strategy		Integrated strategy		Savings (%)
					Total cost	CPU (s)	Total cost	CPU (s)	
1	5	2	3	5	13,055.19	0.38	12,382.36	22.72	5.15
2	5	2	3	5	17,386.57	0.34	16,540.51	5.60	4.87
3	5	2	3	5	15,291.56	0.27	14,848.74	8.52	2.90
4	5	2	3	5	19,634.48	0.39	18,313.76	10.59	6.73
5	5	2	3	5	23,572.30	0.33	22,356.50	11.40	5.16
6	5	2	5	5	31,855.60	0.34	30,410.35	15.78	4.54
7	5	2	5	5	28,328.57	0.30	27,106.13	52.57	4.32
8	5	2	5	5	29,704.99	0.39	28,632.54	12.97	3.61
9	5	2	5	5	39,219.30	0.52	36,794.33	130.29	6.18
10	5	2	5	5	48,061.05	0.36	44,480.04	225.90	7.45
11	5	3	3	5	36,928.96	0.27	32,251.10	62.79	12.67
12	5	3	3	5	31,981.21	0.30	28,966.26	57.03	9.43
13	5	3	3	5	27,324.65	0.25	25,535.43	13.15	6.55
14	5	3	3	5	31,638.52	0.25	28,523.15	55.66	9.85
15	5	3	3	5	22,215.18	0.20	21,009.33	41.61	5.43
16	5	3	5	5	52,210.08	0.34	47,639.32	111.01	8.75
17	5	3	5	5	37,883.87	0.44	35,868.20	13.79	5.32
18	5	3	5	5	50,936.68	0.50	46,326.40	101.07	9.05
19	5	3	5	5	42,469.30	0.47	40,218.10	83.99	5.30
20	5	3	5	5	28,813.06	0.34	28,536.27	66.94	0.96
Mean					31,425.56	0.35	29,336.94	55.17	6.21

This procedure may fail to return a feasible solution as the production planning might be unable to meet the demand imposed by the routing (for example, because there is not enough capacity to meet the demand on time). In practice, this situation triggers a tedious and time-consuming adjustment process in which the routing schedule is modified until a feasible production plan is achieved. We did not implement this adjustment procedure as our main goal was to show that a sequential approach either provides a solution with a higher overall cost or fails to find a feasible solution.

Tables 5, 6 and 7 compare the solution quality and CPU time of both the integrated model and the sequential strategy described above. All the columns are self-explanatory, except for column "Savings (%)", which shows the percentage of savings, or cost reduction, that the integrated strategy achieved when compared with the sequential one. Symbol "†" marks instances where the sequential approach failed (i.e., the lot-sizing problem was infeasible), while symbol "–" indicates instances where the integrated approach did not find a feasible solution after 1 h (i.e., CPLEX stopped with no feasible solution). Of course, in the "Savings (%)" column, we only report values for instances where both strategies returned a feasible solution.

Table 6 Benefits of integrating production and routing decisions: $n = 10$ customers

Instance	n	$\|\mathcal{P}\|$	$\|\mathcal{C}\|$	$\|\mathcal{T}\|$	Sequential strategy		Integrated strategy		Savings (%)
					Total cost	CPU (s)	Total cost	CPU (s)	
21	10	2	3	5	36,793.70	422.51	32,472.89	3600.00	11.74
22	10	2	3	5	39,532.18	17.22	34,385.96	3600.00	13.02
23	10	2	3	5	44,820.54	20.72	39,592.76	3600.00	11.66
24	10	2	3	5	32,295.69	7.53	29,437.46	2547.53	8.85
25	10	2	3	5	34,884.56	4.34	31,008.98	3600.00	11.11
26	10	2	5	5	48,681.10	5.88	44,665.86	3600.00	8.25
27	10	2	5	5	52,357.10	55.26	47,533.48	3600.00	9.21
28	10	2	5	5	45,558.32	7.72	41,129.61	3600.00	9.72
29	10	2	5	5	47,812.04	12.21	42,741.94	3600.00	10.60
30	10	2	5	5	59,431.28	24.78	52,878.70	3600.00	11.03
31	10	3	3	5	59,786.25	137.79	51,793.46	3600.00	13.37
32	10	3	3	5	44,729.33	113.81	40,027.89	3600.00	10.51
33	10	3	3	5	30,153.09	95.59	27,563.59	3600.00	8.59
34	10	3	3	5	47,952.65	85.51	42,281.83	3600.00	11.83
35	10	3	3	5	45,035.91	32.98	39,922.32	3600.00	11.35
36	10	3	5	5	62,075.59	8.83	54,605.62	3600.00	12.03
37	10	3	5	5	64,477.84	3.99	57,149.86	3600.00	11.37
38	10	3	5	5	50,122.22	120.82	44,365.74	3600.00	11.48
39	10	3	5	5	†	6.61	68,986.06	3600.00	–
40	10	3	5	5	65,544.54	46.57	58,064.13	3600.00	11.41
Mean					48,002.31	61.53	44,030.41	3547.38	10.90

Based on the results reported in Tables 5, 6 and 7, it is clear that solving the full integrated model requires a higher computational effort than using a sequential approach. However, as expected, solutions returned after solving the integrated model are cheaper than the ones obtained using the sequential procedure. On average, we achieved savings of 6.21, 10.90 and 11.33% for instances with 5, 10 and 15 customers, respectively, using the integrated model. A reason for these results is that the integrated model tries to achieve the lowest possible overall cost by considering at the same time both production and routing costs, while in the sequential approach each part of the overall cost is optimized regardless of one another.

Furthermore, by solving the integrated model a feasible solution can be found for most of the instances, while the sequential approach fails to solve many instances with $n = 15$ customers. This is because in the sequential approach, the routing decisions are made no matter the impact they may cause on the production decisions. Thus, an optimal or near-optimal solution for the routing problem may lead to an infeasible problem in the production side. This situation is unlikely to happen in the integrated model as all the constraints related to production and routing are considered together in a single framework.

Table 7 Benefits of integrating production and routing decisions: $n = 15$ customers

Instance	n	$\|\mathcal{P}\|$	$\|\mathcal{C}\|$	$\|\mathcal{T}\|$	Sequential strategy		Integrated strategy		Savings (%)
					Total cost	CPU (s)	Total cost	CPU (s)	
41	15	2	3	5	†	1590.84	57,212.94	3600.00	–
42	15	2	3	5	54,886.02	3246.64	48,753.51	3600.00	11.17
43	15	2	3	5	49,608.04	1813.12	43,428.13	3600.00	12.46
44	15	2	3	5	48,740.84	273.78	45,299.84	3600.00	7.06
45	15	2	3	5	47,342.09	67.34	41,302.28	3600.00	12.76
46	15	2	5	5	†	646.55	76,015.51	3600.00	–
47	15	2	5	5	†	3600.00	95,251.30	3600.00	–
48	15	2	5	5	†	2447.16	65,589.68	3600.00	–
49	15	2	5	5	†	719.76	63,170.14	3600.00	–
50	15	2	5	5	†	522.85	76,022.14	3600.00	–
51	15	3	3	5	†	3600.00	59,550.04	3600.00	–
52	15	3	3	5	57,522.20	3600.00	49,086.32	3600.00	14.67
53	15	3	3	5	†	3600.00	63,638.54	3600.00	–
54	15	3	3	5	35,206.36	3600.00	31,820.79	3600.00	9.62
55	15	3	3	5	36,260.16	3600.00	32,068.38	3600.00	11.56
56	15	3	5	5	†	3600.00	–	3600.00	–
57	15	3	5	5	†	1887.36	–	3600.00	–
58	15	3	5	5	†	3600.00	81,438.07	3600.00	–
59	15	3	5	5	†	246.11	–	3600.00	–
60	15	3	5	5	†	57.69	71,353.81	3600.00	–
Mean					47,080.82	2115.96	58,882.44	3600.00	11.33

6.3 Relax-and-fix heuristics results

In this section, we present the results of the RF heuristic proposed in Sect. 5. Results for $n = 5$ customers are not reported, since none of the RF was competitive with CPLEX. This result was expected, as CPLEX found optimal solutions for those instances in very short computational times. The best performance was achieved by RFX4 and RFX1, which were able to find solutions, on average, 0.07 and 1.22% far from the optimal, respectively.

Tables 8, 9, 10 and 11 present the results for instances with $n = 10$ and $n = 15$ customers. Column "CPLEX" shows the best solution found by CPLEX after 1 h of computational time. Then, for each heuristic, columns "Total cost", "Deviation (%)" and "CPU (s)" report the objective value of the solution found by the heuristic (a "–" indicates that the heuristic fails to return a feasible solution), the relative deviation from the CPLEX solutions and the total CPU time, respectively. For a given RF heuristic, each entry of the Deviation (%) column is calculated according to (40) and negative values correspond to instances where the heuristic found a better solution than CPLEX:

Table 8 Relax-and-fix results: $n = 10$ customers

Instance	CPLEX	RFX1			RFX2			RFX3		
		Total cost	Deviation (%)	CPU (s)	Total cost	Deviation (%)	CPU (s)	Total cost	Deviation (%)	CPU (s)
21	32,472.89	32,472.89	0.00	3383.76	32,735.85	0.81	109.56	34,092.31	4.99	104.99
22	34,385.96	34,449.84	0.19	3600.00	35,364.94	2.85	250.67	34,668.04	0.82	180.27
23	39,592.76	39,592.76	0.00	3600.00	39,592.76	0.00	844.95	42,359.26	6.99	3600.00
24	29,437.46	29,551.46	0.39	2931.84	29,700.46	0.89	80.93	29,700.46	0.89	64.15
25	31,008.98	31,008.98	0.00	3600.00	31,829.41	2.65	1644.00	31,829.41	2.65	3286.90
26	44,665.86	44,665.86	0.00	3600.00	46,636.46	4.41	1143.22	49,507.40	10.84	3600.00
27	47,533.48	47,634.07	0.21	3600.00	48,784.88	2.63	1660.83	50,278.75	5.78	2880.15
28	41,129.61	41,129.61	0.00	3600.00	41,798.07	1.63	2169.13	41,719.92	1.44	3600.00
29	42,741.94	42,741.94	0.00	3600.00	42,779.94	0.09	224.94	42,779.94	0.09	369.06
30	52,878.70	52,953.70	0.14	3600.00	53,415.76	1.02	864.64	56,167.86	6.22	2880.60
31	51,793.46	51,860.46	0.13	3600.00	52,132.66	0.65	620.66	52,517.46	1.40	482.33
32	40,027.89	40,347.42	0.80	3600.00	40,453.89	1.06	1449.75	40,453.89	1.06	2511.52
33	27,563.59	27,563.59	0.00	3600.00	28,534.35	3.52	1639.05	28,534.35	3.52	3515.72
34	42,281.83	42,281.83	0.00	3600.00	42,362.42	0.19	174.70	42,362.42	0.19	170.13
35	39,922.32	39,922.32	0.00	3600.00	39,961.16	0.10	1593.71	42,558.01	6.60	3600.00
36	54,605.62	54,497.21	−0.20	3600.00	55,082.50	0.87	2435.39	54,878.68	0.50	3600.00
37	57,149.86	57,149.86	0.00	3600.00	57,396.86	0.43	231.74	58,372.02	2.14	188.04
38	44,365.74	44,365.74	0.00	3600.00	46,406.67	4.60	2173.00	47,366.50	6.76	3600.00
39	68,986.06	68,949.62	−0.05	3600.00	69,654.10	0.97	1260.36	70,952.12	2.85	1652.69
40	58,064.13	57,859.25	−0.35	3600.00	59,182.19	1.93	832.92	59,182.19	1.93	1460.81
Mean	44,030.41	44,049.92	0.06	3555.78	44,690.27	1.56	1070.21	45,514.05	3.38	2067.37

Table 9 Relax-and-fix results: $n = 10$ customers (continuation)

Instance	CPLEX	RFX4			RFX5			RFX6		
		Total cost	Deviation (%)	CPU (s)	Total cost	Deviation (%)	CPU (s)	Total cost	Deviation (%)	CPU (s)
21	32,472.89	32,730.85	0.79	489.86	33,472.59	3.08	3600.00	35,246.72	8.54	1804.18
22	34,385.96	34,923.40	1.56	588.16	34,934.35	1.59	3600.00	35,816.65	4.16	1805.98
23	39,592.76	40,319.26	1.83	427.85	41,021.30	3.61	3057.73	42,250.90	6.71	1804.87
24	29,437.46	29,876.46	1.49	410.43	29,437.46	0.00	381.82	32,256.20	9.58	1808.05
25	31,008.98	31,445.98	1.41	425.92	31,241.84	0.75	1066.08	33,877.33	9.25	1800.25
26	44,665.86	44,665.86	0.00	722.21	48,019.50	7.51	52.55	50,754.38	13.63	1802.40
27	47,533.48	47,735.41	0.42	410.89	–	–	3600.00	52,503.46	10.46	1804.23
28	41,129.61	42,368.56	3.01	1526.73	44,023.56	7.04	168.83	45,437.75	10.47	1804.28
29	42,741.94	43,056.22	0.74	722.01	42,741.94	0.00	1772.87	47,061.21	10.11	1802.38
30	52,878.70	53,167.70	0.55	636.94	52,878.70	0.00	2583.58	56,776.18	7.37	1803.43
31	51,793.46	52,124.10	0.64	744.15	53,032.42	2.39	3600.00	53,012.66	2.35	1800.85
32	40,027.89	40,455.46	1.07	1136.49	41,449.90	3.55	3600.00	43,747.86	9.29	1806.50
33	27,563.59	27,563.59	0.00	721.02	–	–	3600.00	30,840.85	11.89	1804.52
34	42,281.83	42,382.20	0.24	441.62	43,067.00	1.86	3600.00	43,402.16	2.65	1803.46
35	39,922.32	39,922.32	0.00	391.71	40,859.34	2.35	1944.67	42,723.62	7.02	1802.83
36	54,605.62	55,117.54	0.94	1238.97	55,246.71	1.17	3600.00	56,794.16	4.01	1802.97
37	57,149.86	57,643.38	0.86	424.34	–	–	3600.00	59,818.79	4.67	1803.22
38	44,365.74	44,759.28	0.89	566.93	44,557.60	0.43	3600.00	47,856.76	7.87	1807.18
39	68,986.06	68,986.06	0.00	723.19	–	–	3600.00	71,370.77	3.46	1802.36
40	58,064.13	58,000.53	−0.11	724.24	58,656.43	1.02	3600.00	60,402.39	4.03	1801.13
Mean	44,030.41	44,362.21	0.82	673.68	43,415.04	2.27	2711.41	47,097.54	7.38	1803.75

Table 10 Relax-and-fix results: $n = 15$ customers

Instance	CPLEX	RFX1			RFX2			RFX3		
		Total cost	Deviation (%)	CPU (s)	Total cost	Deviation (%)	CPU (s)	Total cost	Deviation (%)	CPU (s)
41	57,212.94	59,230.64	3.53	3600.00	57,093.02	−0.21	2352.26	57,103.94	−0.19	1486.98
42	48,753.51	49,490.24	1.51	3600.00	48,989.59	0.48	2073.21	48,967.97	0.44	2881.37
43	43,428.13	–	–	3600.00	46,217.20	6.42	2077.05	46,217.20	6.42	2351.91
44	45,299.84	–	–	3600.00	42,814.20	−5.49	1518.71	43,580.70	−3.80	2433.62
45	41,302.28	42,745.04	3.49	3600.00	41,515.44	0.52	1538.12	41,515.44	0.52	815.43
46	76,015.51	–	–	3600.00	72,227.45	−4.98	2895.24	76,982.48	1.27	3600.00
47	95,251.30	†	–	2161.10	–	–	3600.00	–	–	3600.00
48	65,589.68	67,750.16	3.29	3600.00	65,929.16	0.52	2356.09	65,929.16	0.52	1845.46
49	63,170.14	64,483.37	2.08	3600.00	64,863.77	2.68	1508.84	–	–	3600.00
50	76,022.14	†	–	2161.10	74,562.32	−1.92	2890.13	–	–	3600.00
51	59,550.04	57,196.39	−3.95	3600.00	56,635.09	−4.89	1702.60	55,231.25	−7.25	771.07
52	49,086.32	49,473.11	0.79	3600.00	47,316.00	−3.61	1978.05	47,255.84	−3.73	1589.21
53	63,638.54	61,911.51	−2.71	3600.00	63,991.22	0.55	2886.19	63,991.22	0.55	1950.02
54	31,820.79	–	–	3600.00	31,546.58	−0.86	2900.78	33,432.67	5.07	2882.25
55	32,068.38	–	–	3600.00	32,042.38	−0.08	2896.81	–	–	3600.00
56	–	†	–	2161.20	–	–	3600.00	–	–	3600.00
57	–	†	–	2161.30	–	–	3600.00	–	–	3600.00
58	81,438.07	78,939.16	−3.07	3600.00	77,828.30	−4.43	2888.04	77,828.30	−4.43	2881.89
59	–	93,346.95	–	3600.00	92,237.07	–	2202.23	92,237.07	–	1618.41
60	71,353.81	72,063.95	1.00	3600.00	70,103.17	−1.75	2881.56	70,159.55	−1.67	2882.71
Mean	58,882.44	63,330.05	0.60	3312.24	57,994.82	−1.07	2517.30	58,602.34	−0.48	2579.52

Table 11 Relax-and-fix results: $n = 15$ customers (continuation)

Instance	CPLEX	RFX4			RFX5			RFX6		
		Total cost	Deviation (%)	CPU (s)	Total cost	Deviation (%)	CPU (s)	Total cost	Deviation (%)	CPU (s)
41	57,212.94	58,168.80	1.67	1082.71	67,043.50	17.18	3600.00	57,916.08	1.23	1808.98
42	48,753.51	49,929.80	2.41	1272.08	51,734.91	6.12	3600.00	50,046.60	2.65	1809.79
43	43,428.13	43,907.45	1.10	751.91	–	–	3600.00	44,809.17	3.18	1804.06
44	45,299.84	45,513.70	0.47	1289.58	43,122.66	−4.81	3600.00	45,000.82	−0.66	1807.04
45	41,302.28	†	–	1132.84	42,144.04	2.04	3600.00	43,048.60	4.23	1811.34
46	76,015.51	76,896.81	1.16	1444.66	77,167.42	1.52	3600.00	78,676.67	3.50	1802.96
47	95,251.30	†	–	729.27	109,039.30	14.48	3600.00	98,277.16	3.18	1807.01
48	65,589.68	65,416.44	−0.26	1000.65	66,249.56	1.01	3600.00	66,037.98	0.68	1802.71
49	63,170.14	63,817.84	1.03	736.22	67,919.94	7.52	3600.00	67,394.23	6.69	1802.12
50	76,022.14	76,403.08	0.50	1296.30	81,051.24	6.62	3600.00	77,237.24	1.60	1803.11
51	59,550.04	†	–	757.94	62,806.47	5.47	3600.00	57,751.39	−3.02	1804.78
52	49,086.32	†	–	974.10	47,745.04	−2.73	3600.00	47,450.45	−3.33	1802.38
53	63,638.54	†	–	1322.78	78,460.32	23.29	3600.00	64,121.36	0.76	1806.70
54	31,820.79	31,829.57	0.03	970.80	31,732.11	−0.28	3600.00	33,723.00	5.98	1803.71
55	32,068.38	35,740.79	11.45	886.25	–	–	3600.00	34,533.12	7.69	1800.20
56	–	†	–	1200.54	13,2,225.31	–	3600.00	†	–	1800.05
57	–	†	–	1684.12	106,605.37	–	3600.00	100,875.63	–	1800.20
58	81,438.07	†	–	960.60	86,572.32	6.30	3600.00	78,207.95	−3.97	1806.98
59	–	†	–	977.49	100,270.64	–	3600.00	97,815.84	–	1802.19
60	71,353.81	†	–	1040.75	75,149.93	5.32	3600.00	71,581.29	0.32	1805.13
Mean	58,882.44	54,762.43	1.96	1075.58	73,724.45	5.94	3600.00	63,921.29	1.81	1804.57

Table 12 Hybrid relax-and-fix results: $n = 10$ customers

Instance	CPLEX	Hybrid RFX5			Hybrid RX6		
		Total cost	Deviation (%)	CPU (s)	Total cost	Deviation (%)	CPU (s)
21	32,472.89	32,472.89	0.00	3600.00	32,472.89	0.00	3600.00
22	34,385.96	34,385.96	0.00	3600.00	34,385.96	0.00	3600.00
23	39,592.76	39,592.76	0.00	3600.00	39,592.76	0.00	3600.00
24	29,437.46	29,437.46	0.00	1684.06	29,437.46	0.00	3600.00
25	31,008.98	31,008.98	0.00	3600.00	31,008.98	0.00	3600.00
26	44,665.86	44,888.86	0.50	3600.00	44,665.86	0.00	3600.00
27	47,533.48	47,533.48	0.00	3600.00	47,533.48	0.00	3600.00
28	41,129.61	41,058.66	−0.17	3600.00	41,495.61	0.89	3600.00
29	42,741.94	42,741.94	0.00	3600.00	42,741.94	0.00	3600.00
30	52,878.70	52,878.70	0.00	3600.00	52,878.70	0.00	3600.00
31	51,793.46	52,029.24	0.46	3600.00	51,860.46	0.13	3600.00
32	40,027.89	40,251.27	0.56	3600.00	40,121.89	0.23	3600.00
33	27,563.59	27,563.59	0.00	3600.00	27,563.59	0.00	3600.00
34	42,281.83	42,281.83	0.00	3600.00	42,281.83	0.00	3600.00
35	39,922.32	39,922.32	0.00	3600.00	39,922.32	0.00	3600.00
36	54,605.62	54,537.47	−0.12	3600.00	54,586.02	−0.04	3600.00
37	57,149.86	57,149.86	0.00	3600.00	57,149.86	0.00	3600.00
38	44,365.74	44,365.74	0.00	3600.00	44,365.74	0.00	3600.00
39	68,986.06	69,044.08	0.08	3600.00	68,970.62	−0.02	3600.00
40	58,064.13	57,824.53	−0.41	3600.00	59,231.97	2.01	3600.00
Mean	44,030.41	44,048.48	0.04	3504.20	44,113.40	0.16	3600.00

$$\text{Deviation (\%)} = \left(\frac{\text{Total cost} - \text{CPLEX}}{\text{CPLEX}}\right) \times 100. \tag{40}$$

Although smaller than the full integrated model, the MIP sub-models at each iteration of the RF heuristics might be difficult to solve, and therefore for some instances the heuristics failed to find a feasible solution within the preset computational time limit. In other cases, decisions made in early iterations led to infeasible subproblems in latter iterations. We use the following criterion to report these results: in the column "Total cost", we used the symbol "–" to mark instances where no solution was obtained because of the time limit, while the symbol "†" was used when the heuristic stopped due to infeasibility.

For instances with $n = 10$ customers, RFX1 clearly achieved the best results in terms of total cost. This heuristic found slightly better solutions than CPLEX for three instances (ins. 36, 39 and 40), and for 11 other instances it returned the same solution as CPLEX. For the remaining six instances the Deviation (%) was less than 1%. RFX4 also presented good results, improving one instance and returning the same solution as CPLEX for four other instances. Except for instance 28, the Deviation (%) is less than 2%, which shows that the heuristic found solutions of similar quality to those

Table 13 Hybrid relax-and-fix results: $n = 15$ customers

Instance	CPLEX	Hybrid RFX5			Hybrid RX6		
		Total cost	Deviation (%)	CPU (s)	Total cost	Deviation (%)	CPU (s)
41	57,212.94	56,831.26	−0.67	3600.00	56,810.06	−0.70	3600.00
42	48,753.51	48,904.50	0.31	3600.00	49,367.30	1.26	3600.00
43	43,428.13	43,748.75	0.74	3600.00	43,359.13	−0.16	3600.00
44	45,299.84	43,122.66	−4.81	3600.00	43,709.17	−3.51	3600.00
45	41,302.28	41,573.40	0.66	3600.00	41,619.28	0.77	3600.00
46	76,015.51	75,820.01	−0.26	3600.00	76,610.56	0.78	3600.00
47	95,251.30	97,048.11	1.89	3600.00	95,034.09	−0.23	3600.00
48	65,589.68	65,858.22	0.41	3600.00	65,942.94	0.54	3600.00
49	63,170.14	63,595.64	0.67	3600.00	65,425.40	3.57	3600.00
50	76,022.14	74,902.96	−1.47	3600.00	75,255.98	−1.01	3600.00
51	59,550.04	57,452.28	−3.52	3600.00	56,123.89	−5.75	3600.00
52	49,086.32	47,323.30	−3.59	3600.00	47,185.79	−3.87	3600.00
53	63,638.54	71,987.31	13.12	3600.00	63,546.34	−0.14	3600.00
54	31,820.79	31,590.79	−0.72	3600.00	31,748.79	−0.23	3600.00
55	32,068.38	32,846.22	2.43	3600.00	32,217.82	0.47	3600.00
56	–	116,363.57	–	3600.00	–	–	3600.00
57	–	95,179.39	–	3600.00	94,773.76	–	3600.00
58	81,438.07	78,934.66	−3.07	3600.00	79,603.95	−2.25	3600.00
59	–	95,962.62	–	3600.00	97,766.84	–	3600.00
60	71,353.81	70,582.02	−1.08	3600.00	70,200.87	−1.62	3600.00
Mean	58,882.44	65,481.38	0.06	3600.00	62,436.95	−0.71	3600.00

achieved by CPLEX. Moreover, RFX4 was able to find such high-quality solutions in less CPU time than both CPLEX and RFX1. RFX2 appeared a bit less effective than RFX1 and RFX4 in terms of solution quality. It found the same solution as CPLEX for one instance, while for the remaining ones, the Deviation (%) was between 0.09 and 4.60%.

All the remaining heuristics found solutions with larger Deviation (%) and, in general, performed worse than RFX1, RFX2 and RFX4. RFX5 found the same solution as CPLEX in three cases, but it failed to return feasible solutions for three other instances. The Deviation (%) of RFX3 and RFX6 was between 0.09–10.84% and 2.35–13.63%, respectively.

Results for $n = 15$ confirm that those instances are the most difficult to solve. The best results were achieved by RFX2, which found better solutions than CPLEX in 10 out of 20 instances, although it could not return feasible solutions for 3 other instances. The percentage of improvement concerning the CPLEX solutions was between 0.08 and 5.49% for these 10 instances. Heuristics RFX1, RFX3 and RFX4 showed a poor performance, failing to solve many of the instances. We observed that RFX3 obtained better solutions than CPLEX for six instances. However, we claim it performed poorly because it also failed to solve six other instances.

Heuristic RFX5 found feasible solutions for most of the instances (18 out of 20). The Deviation (%) suggests that these solutions are, in general, far from the best solution found by CPLEX. However, we noticed that RFX5 found feasible solutions for instances 56, 57 and 59, which were not solved by CPLEX. Heuristic RFX6 found feasible solutions for all but one instance (instance 56). Contrarily to RFX5, RFX6 provided solutions very close to the best ones returned by CPLEX (1.81% deviation on average) and, in addition, it required less computational time than CPLEX (1804.81 s on average).

We further explored the potential of heuristics RFX5 and RFX6 using the solutions they returned as an initial solution for CPLEX. In order to make a fair comparison, we ran RFX5 and RFX6 again for at most 1800 s. Then, we fed the solution returned by the heuristics, if any, into CPLEX and let it run for another 1800 s. Tables 12 and 13 show the results of these new strategies, called "Hybrid RFX5" and "Hybrid RFX6", respectively.

Tables 12 and 13 show that the new hybrid strategies are effective and competitive with CPLEX. For instances with $n = 10$, the hybrid RFX5 slightly improved 3 out of the 20 instances and had an average deviation of 0.04%. Similarly, the hybrid RFX6 improved 2 out of the 20 instances and had an average deviation of 0.16%. Roughly speaking, for $n = 10$ customers, the hybrid RFX5 obtained the best results among all the heuristics, outperforming the previous results of RFX1, while the hybrid RFX6 was ranked as the third best strategy (Tables 8, 9, 10, 11).

For $n = 15$ customers, both strategies outperformed CPLEX in terms of the number of feasible solutions found (20 and 19 feasible solutions, respectively). The hybrid RFX5 was the only strategy able to find a feasible solution for all the instances, which shows that the combination of RFX5 and CPLEX is a good alternative to solve the problem. In this combination, RFX5 may find feasible solutions for instances that CPLEX cannot solve by itself (e.g., instances 56, 57 and 59). Conversely, when RFX5 fails to provide an initial feasible solution (e.g., instances 43 and 55), CPLEX is able to find very good solutions from scratch. In addition, it is worth mentioning that the hybrid RFX5 found better solutions than CPLEX for nine instances, with an average deviation of 0.06%.

The hybrid RFX6 only failed to find a feasible solution for instance 56. Among the remaining 19 instances, this hybrid heuristic obtained better solutions than CPLEX for 11 of them and, except for instance 49, deviations were less than 1.30%. On average, the hybrid RFX6 achieved solutions 0.71% better than the ones obtained by CPLEX.

Since the number of instances solved by each heuristic was different, the mean values reported in Tables 10, 11 and 13 were not comparable. Therefore, in order to identify the best performing strategies on the set of instances with $n = 15$ customers, we plotted the performance profile of the RF strategies and the CPLEX solver in Fig. 4. The value of $P(s, q)$ indicates how often a given algorithm s finds solutions with at most q% deviation from the best solution known. Therefore, the value of $P(s, q)$ when $q = 0\%$ indicates the fraction of instances whereby algorithm s finds the best solution. The value of q when $P(s, q) = 1$ indicates that, for instances where algorithm s does not find the best results, it obtains solutions with at most q% of deviation from the best known. The best algorithm is the one with the lowest value of q when $P(s, q) = 1$.

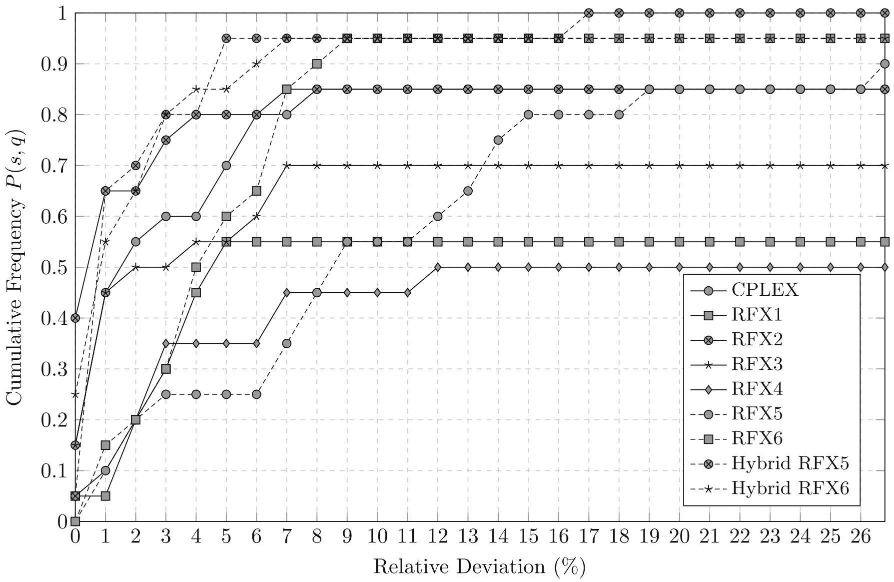

Fig. 4 Performance profiles of the relax-and-fix heuristics for instances with $n = 15$ customers

For more details about the performance profile tool, readers may consult the work of Dolan and Moré (2002).

In Fig. 4, only the performance profile of the hybrid RFX5 converged to one as this was the only strategy that solved all the instances with $n = 15$ customers. A drawback of this strategy is that it only found the best solution 5% of the times. However, for 65% of the instances, it found solutions with at most 1% deviation from the best known and, in general, it solved 95% of the instances with a relative deviation less than or equal to 5%. Another high-performing strategy was the hybrid RFX6, which found the best solution among all the strategies 25% of the times. Roughly speaking, this strategy provided feasible solutions with up to 5% of deviation for 85% of the instances, just behind the performance of the hybrid RFX5.

Among the pure RF heuristics, only RFX2 seemed to have a competitive performance. In fact, this heuristic found the best solution for 40% of the instances and provided good feasible solutions (at most 7% deviation) for all the other instances that it solved. Regarding CPLEX, results pointed out that it was outperformed by the hybrids RFX5 and RFX6, as well as by the heuristic RFX2. In general, it found the best solution 15% of the times, less than both the RFX2 and the hybrid RFX6. Moreover, it returned solutions with deviation values below 5% only 70% of the times.

7 Concluding remarks

In this paper, we proposed a MIP model to represent and solve a PIDRP in small furniture companies. The model considered some features typically found in small furniture companies in Brazil, such as producing and stocking parts and sub-assemblies, limited production and distribution capacities, routes extending over several production periods, multiple time windows and customers' due dates. We used the solver CPLEX to solve a set of 60 randomly generated instances. Results showed that CPLEX solved to optimality instances with $n = 5$ customers, and near-optimal solutions were found for most of the instances with $n \geq 10$ customers.

We also proposed six relax-and-fix heuristics, which used different criteria for partitioning and fixing variables, in order to identify good-quality solutions for the largest instances ($n = 15$ customers). Among those heuristics, the best performance was achieved by RFX2, which found better solutions than CPLEX for 10 out of 20 instances, with a percentage of improvement between 0.08 and 5.49%.

Based on the results of heuristics RFX5 and RFX6, which found feasible solutions for most of the instances, we decided to further explore them by setting the solution that they returned as an initial solution for CPLEX. These two new strategies were very effective in solving the problem, outperforming both CPLEX and RFX2. In general, they provided better solutions than CPLEX for 9 and 11 instances, respectively. They also returned good feasible solutions for most of the remaining instances (at most 5% deviation for 95 and 85% of the instances, respectively).

An interesting perspective for future research is to develop more specific solution strategies, combining mathematical programming and heuristic methods (i.e., hybrid or matheuristic methods) in order to find better solutions than those obtained by the CPLEX solver and the proposed RF heuristics. Another interesting line of research is to extend the formulation in order to include sequence-dependent setup times and costs, as well as heterogeneous fleet of vehicles. Considering these two new features would result in a more complex scenario, which is often found in larger furniture companies.

Acknowledgements The authors would like to thank the two anonymous reviewers for their useful comments and suggestions of revision.

References

Absi N, Archetti C, Dauzère-Pérès S, Feillet D (2014) A two-phase iterative heuristic approach for the production routing problem. Transp Sci 49(4):784–795. doi:10.1287/trsc.2014.0523

Adulyasak Y, Cordeau JF, Jans R (2014a) Formulations and branch-and-cut algorithms for multivehicle production and inventory routing problems. INFORMS J Comput 26:103–120

Adulyasak Y, Cordeau JF, Jans R (2014b) Optimization-based adaptive large neighborhood search for the production routing problem. Transp Sci 48(1):20–45

Adulyasak Y, Cordeau JF, Jans R (2015) Benders decomposition for production routing under demand uncertainty. Oper Res 63(4):851–867. doi:10.1287/opre.2015.1401

Adulyasak Y, Cordeau JF, Jans R (2015b) The production routing problem: a review of formulations and solution algorithms. Comput Oper Res 55:141–152. doi:10.1016/j.cor.2014.01.011

Amorim P, Belo-Filho M, Toledo F, Almeder C, Almada-Lobo B (2013) Lot sizing versus batching in the production and distribution planning of perishable goods. Int J Prod Econ 146(1):208–218. doi:10.1016/j.ijpe.2013.07.001

Amorim P, Parragh SN, Sperandio F, Almada-Lobo B (2014) A rich vehicle routing problem dealing with perishable food: a case study. TOP 22(2):489–508. doi:10.1007/s11750-012-0266-4

Araujo S, Arenales M, Clark A (2007) Joint rolling-horizon scheduling of materials processing and lot-sizing with sequence-dependent setups. J Heuristics 13(4):337–358. doi:10.1007/s10732-007-9011-9

Archetti C, Bertazzi L, Paletta G, Speranza MG (2011) Analysis of the maximum level policy in a production–distribution system. Comput Oper Res 38(12):1731–1746. doi:10.1016/j.cor.2011.03.002

Armentano V, Shiguemoto A, Løkketangen A (2011) Tabu search with path relinking for an integrated production–distribution problem. Comput Oper Res 38(8):1199–1209. doi:10.1016/j.cor.2010.10.026
Armstrong R, Gao S, Lei L (2007) A zero-inventory production and distribution problem with a fixed customer sequence. Ann Oper Res 159(1):395–414. doi:10.1007/s10479-007-0272-3
Azi N, Gendreau M, Potvin JY (2007) An exact algorithm for a single-vehicle routing problem with time windows and multiple routes. Eur J Oper Res 178(3):755–766. doi:10.1016/j.ejor.2006.02.019
Bard JF, Nananukul N (2009) Heuristics for a multiperiod inventory routing problem with production decisions. Comput Ind Eng 57(3):713–723. doi:10.1016/j.cie.2009.01.020
Bard JF, Nananukul N (2009b) The integrated production–inventory–distribution–routing problem. J Sched 12(3):257–280. doi:10.1007/s10951-008-0081-9
Bard JF, Nananukul N (2010) A branch-and-price algorithm for an integrated production and inventory routing problem. Comput Oper Res 37(12):2202–2217. doi:10.1016/j.cor.2010.03.010
Beamon BM (1998) Supply chain design and analysis: models and methods. Int J Prod Econ 55:281–294
Belhaiza S, Hansen P, Laporte G (2014) A hybrid variable neighborhood tabu search heuristic for the vehicle routing problem with multiple time windows. Comput Oper Eng 52(Part B):269–281. doi:10.1016/j.cor.2013.08.010
Beraldi P, Ghiani G, Grieco A, Guerriero E (2008) Rolling-horizon and fix-and-relax heuristics for the parallel machine lot-sizing and scheduling problem with sequence-dependent set-up costs. Comput Oper Res 35(11):3644–3656. doi:10.1016/j.cor.2007.04.003
Bertazzi L, Paletta G, Speranza MG (2005) Minimizing the total cost in an integrated vendor-managed inventory system. J Heuristics 11:393–419
Boudia M, Prins C (2009) A memetic algorithm with dynamic population management for an integrated production–distribution problem. Eur J Oper Res 195(3):703–715. doi:10.1016/j.ejor.2007.07.034
Boudia M, Louly M, Prins C (2007) A reactive GRASP and path relinking for a combined production–distribution problem. Comput Oper Res 34(11):3402–3419. doi:10.1016/j.cor.2006.02.005
Brahimi N, Aouam T (2016) Multi-item production routing problem with backordering: a MILP approach. Int J Prod Res 54(4):1076–1093. doi:10.1080/00207543.2015.1047971
Chandra P, Fisher ML (1994) Coordination of production and distribution planning. Eur J Oper Res 72:503–517
Chen HK, Hsueh CF, Chang MS (2009) Production scheduling and vehicle routing with time windows for perishable food products. Comput Oper Res 36(7):2311–2319. doi:10.1016/j.cor.2008.09.010
Chen ZL (2004) Integrated production and distribution operations: taxonomy, models, and review. In: Simchi-Levi D, Wu S, Shen ZJ (eds) Handbook of quantitative supply chain analysis: modeling in the e-business era, chap 17. Kluwer Academic Publishers, Boston, pp 711–745
Chen ZL (2010) Integrated production and outbound distribution scheduling: review and extensions. Oper Res 58(1):130–148. doi:10.1287/opre.1080.0688
Chen ZL, Pundoor G (2006) Order assignment and scheduling in a supply chain. Oper Res 54(3):555–572. doi:10.1287/opre.1060.0280
Chen ZL, Vairaktarakis GL (2005) Integrated scheduling of production and distribution operations. Manag Sci 51(4):614–628. doi:10.1287/mnsc.1040.0325
Chiang WC, Russell R, Xu X, Zepeda D (2009) A simulation/metaheuristic approach to newspaper production and distribution supply chain problems. Int J Prod Econ 121(2):752–767. doi:10.1016/j.ijpe.2009.03.001
Díaz-Madroñero M, Peidro D, Mula J (2015) A review of tactical optimization models for integrated production and transport routing planning decisions. Comput Ind Eng 88:518–535. doi:10.1016/j.cie.2015.06.010
Dillenberger C, Escudero LF, Wollensak A, Zhang W (1994) On practical resource allocation for production planning and scheduling with period overlapping setups. Eur J Oper Res 75(2):275–286. doi:10.1016/0377-2217(94)90074-4
Dolan ED, Moré JJ (2002) Benchmarking optimization software with performance profiles. Math Progr 91(2):201–213. doi:10.1007/s101070100263
Farahani P, Grunow M, Günther HO (2012) Integrated production and distribution planning for perishable food products. Flex Serv Manuf J 24(1):28–51. doi:10.1007/s10696-011-9125-0
Ferreira D, Morabito R, Rangel S (2009) Solution approaches for the soft drink integrated production lot sizing and scheduling problem. Eur J Oper Res 196(2):697–706. doi:10.1016/j.ejor.2008.03.035

Ferreira D, Morabito R, Rangel S (2010) Relax and fix heuristics to solve one-stage one-machine lot-scheduling models for small-scale soft drink plants. Comput Oper Res 37(4):684–691. doi:10.1016/j.cor.2009.06.007

Fumero F, Vercellis C (1999) Synchronized development of production, inventory, and distribution schedules. Transp Sci 33(3):330–340. doi:10.1287/trsc.33.3.330

Geismar HN, Laporte G, Lei L, Sriskandarajah C (2008) The integrated production and transportation scheduling problem for a product with a short lifespan. INFORMS J Comput 20(1):21–33. doi:10.1287/ijoc.1060.0208

Goel A (2012) The minimum duration truck driver scheduling problem. EURO J Transp Logist 1(4):285–306. doi:10.1007/s13676-012-0014-9

Gramani M, França P, Arenales M (2009) A lagrangian relaxation approach to a coupled lot-sizing and cutting stock problem. Int J Prod Econ 119(2):219–227. doi:10.1016/j.ijpe.2009.02.011

IBM ILOG CPLEX (2013) IBM ILOG CPLEX Optimization Studio: CPLEX user's manual v.12.6, IBM Corporation

James R, Almada-Lobo B (2011) Single and parallel machine capacitated lotsizing and scheduling: new iterative MIP-based neighborhood search heuristics. Comput Oper Res 38(12):1816–1825. doi:10.1016/j.cor.2011.02.005

Jans R, Degraeve Z (2008) Modeling industrial lot sizing problems: a review. Int J Prod Res 46(6):1619–1643. doi:10.1080/00207540600902262

Lei L, Liu S, Ruszczynski A, Park S (2006) On the integrated production, inventory, and distribution routing problem. IIE Trans 38(11):955–970. doi:10.1080/07408170600862688

Molina F, Morabito R, Araujo S (2016) MIP models for production lot sizing problems with distribution costs and cargo arrangement. J Oper Res Soc 67(11):1395–1407. doi:10.1057/jors.2016.12

Moons S, Ramaekers K, Caris A, Arda Y (2017) Integrating production scheduling and vehicle routing decisions at the operational decision level: a review and discussion. Comput Ind Eng 104:224–245

Mula J, Peidro D, Díaz-Madroñero M, Vicens E (2010) Mathematical programming models for supply chain production and transport planning. Eur J Oper Res 204(3):377–390. doi:10.1016/j.ejor.2009.09.008

Quadt D, Kuhn H (2007) Capacitated lot-sizing with extensions: a review. 4OR 6(1):61–83. doi:10.1007/s10288-007-0057-1

Rancourt ME, Cordeau JF, Laporte G (2013) Long-haul vehicle routing and scheduling with working hour rules. Transp Sci 47(1):81–107

Russell R, Chiang WC, Zepeda D (2008) Integrating multi-product production and distribution in newspaper logistics. Comput Oper Res 35(5):1576–1588. doi:10.1016/j.cor.2006.09.002

Sarmiento AM, Nagi R (1999) A review of integrated analysis of production–distribution systems. IIE Trans 31(11):1061–1074

Savelsbergh M, Song JH (2008) An optimization algorithm for the inventory routing problem with continuous moves. Comput Oper Res 35(7):2266–2282. doi:10.1016/j.cor.2006.10.020

Shiguemoto AL, Armentano V (2010) A tabu search procedure for coordinating production, inventory and distribution routing problems. Int Trans Oper Res 17(2):179–195. doi:10.1111/j.1475-3995.2009.00741.x

Toledo FMB, Armentano V (2006) A lagrangian-based heuristic for the capacitated lot-sizing problem in parallel machines. Eur J Oper Res 175(2):1070–1083. doi:10.1016/j.ejor.2005.06.029

Trigeiro WW, Thomas LJ, McClain JO (1989) Capacitated lot sizing with setup times. Manag Sci 35(3):353–366. doi:10.1287/mnsc.35.3.353

Vidal CJ, Goetschalckx M (1997) Strategic production–distribution models: a critical review with emphasis on global supply chain models. Eur J Oper Res 98(1):1–18