Transportation Science

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

The Fixed-Partition Policy Inventory Routing Problem

Ali Diabat, Claudia Archetti, Waleed Najy

informs
http://pubsonline.informs.org/journal/trsc

TRANSPORTATION SCIENCE
Vol. 55, No. 2, March–April 2021, pp. 353–370
ISSN 0041-1655 (print), ISSN 1526-5447 (online)

The Fixed-Partition Policy Inventory Routing Problem

Ali Diabat,[a,b] Claudia Archetti,[c] Waleed Najy[a]

[a] Division of Engineering, New York University Abu Dhabi, Abu Dhabi, United Arab Emirates; [b] Tandon School of Engineering, New York University, Brooklyn, New York 11201; [c] Department of Information Systems, Decision Sciences and Statistics, ESSEC Business School, 95000 Cergy, France

Contact: diabat@nyu.edu, https://orcid.org/0000-0002-0902-0011 (AD); archetti@essec.edu, https://orcid.org/0000-0002-3524-1600 (CA); wn11@nyu.edu, https://orcid.org/0000-0003-0639-1949 (WN)

Received: October 23, 2019
Revised: March 26, 2020; August 31, 2020
Accepted: September 13, 2020
Published Online in Articles in Advance: February 15, 2021

https://doi.org/10.1287/trsc.2020.1019

Abstract. In this paper, we formally introduce a variant of the inventory routing problem (IRP) that we call the fixed-partition policy IRP (FPP-IRP). In contrast to the classical IRP in which delivery routes are arbitrary, the FPP-IRP partitions customers into mutually exclusive clusters that are fixed throughout the optimization horizon, and distribution is performed separately for each cluster. By restricting the flexibility inherent in the classical IRP, the FPP-IRP attains many potential advantages. First, partitioning reduces the operational complexity of the system and allows a simpler organization of the distribution service. Second, it improves the robustness of the system by isolating disruptions to affected clusters. Third, it can fit the needs and requirements of specific applications in which consistency in the distribution policy, such as familiarity between customers and drivers and route invariance, is required. We present two fixed-partition policies for the IRP together with mathematical formulations and valid inequalities. We also present a worst-case analysis on the performance of these policies. Extensive computational results are presented to show the behavior of these policies and glean insights into their potential benefits.

Keywords: inventory routing problem • fixed-partition policies • consistent routing

1. Introduction

The inventory routing problem (IRP) is a classical logistics problem that models systems involving both transportation planning to and inventory management at various sites in a distribution network. In particular, the IRP deals with determining a routing plan for a given fleet of vehicles that minimizes costs while satisfying the demands of all customers for a single product throughout a given planning horizon. The cost function is given by the sum of transportation (or routing) costs and inventory holding costs. This problem has attracted a lot of attention from the research community in the past few years. We refer the reader to a recent survey by Coelho, Cordeau, and Laporte (2014) and two tutorials by Bertazzi and Speranza (2012, 2013) for a comprehensive review of the literature and applications. The main reason for this rising interest is the many practical contexts in which this problem finds applications and the potential economic benefits arising when combining distribution and inventory management. These two operations have traditionally tended to be planned separately, mainly because of the complexity of the integrated approach. However, recent contributions to the IRP literature have improved the state of the art on dealing with this difficult problem class in terms of both modeling and solution methodologies. This has been coupled with studies aiming to demonstrate the economic benefits attainable from an integrated strategy (Archetti and Speranza 2016).

Despite its advantages, however, there are potential weaknesses to the IRP model as it stands. For one, the flexibility of the IRP's routing can be a disadvantage: because routes are allowed to be arbitrary, the resulting distribution plan can be overly complex, dispatching vehicles to ever-changing routes over time and making concise bookkeeping difficult.

Second, the routes produced by the general IRP are often not robust to unforeseen disruptions. For example, suppose a delivery truck breaks down unexpectedly. Then, the effects of this disruption are not restricted to the customers in that route or to that time period. Rather, a failure of a vehicle can potentially affect many parts of the network because the vehicle may have been scheduled to visit a large fraction of customers over all subsequent time periods. This can lead to a system-wide propagation of backlogs and chronic disruptions, which can make the task of reoptimizing for new routes difficult. This, in turn, is detrimental to the vendor's perceived reliability in the eyes of customers, which can be a deal-breaker for them especially in industries that operate on thin margins.

IRP solutions also lack what has generically become known in the routing literature as "consistency."

In this context, consistency loosely means that some aspect of the solution remains invariant over the planning horizon; for example, if, whenever a customer is visited, the customer is visited by the same driver or if the driver is assigned a fixed delivery route. In the past few years, consistency has been identified as a key feature in customer-oriented routing because some degree of familiarity between driver and customer is viewed as an important competitive advantage in some industries, such as small-package shipping (Groër, Golden, and Wasil 2009; Kovacs et al. 2014b). In addition, familiarity with the route itself can increase the efficiency of delivery and reduce transit times, and it is also linked to increased job satisfaction on the part of the driver as demonstrated by the fact that many job postings for drivers list consistent routes as an incentive. Another application in which the notion of consistency is important is in home healthcare delivery, in which nurses or health practitioners periodically visit patients to deliver treatments and medicines (the routing of nurse visitations while maintaining sufficient supplies of medicine can readily be modeled as an IRP). The medical literature provides strong evidence for the fact that patient–nurse familiarity is a key factor in achieving therapeutic outcomes (Minick and Harvey 2003; Mok and Chiu 2004; Fleischer et al. 2009; Massey, Chaboyer, and Aitken 2014). This consistency feature is missing in the assumptions of the IRP, however, because the model is allowed to construct arbitrary routes, and these routes tend to bear little resemblance to each other from one time period to the next. As a result, vehicles are deployed to ever-changing subsets of customers, invariably involving new routes every time period, and hence, the driver fails to build familiarity with either customers or routes. Routing consistency has largely been studied in the context of the vehicle routing problem (Macdonald, Dörner, and Gandibleux 2009; Feillet et al. 2014; Kovacs, Parragh, and Hartl 2014a; Rodríguez-Martín, Salazar-González, and Yaman 2019). However, and to the best of our knowledge, only one work has, thus far, dealt with the concept of consistency in the context of the IRP, namely Coelho, Cordeau, and Laporte (2012). We note that another type of consistency is one in which it is required that a distribution plan is defined over a finite horizon such that it can be reproduced without changes in the following planning periods. This is the objective of the cyclic IRP (see Raa and Dullaert (2017) for a recent reference).

One way to address these shortcomings of the classical IRP is to impose a *fixed-partition policy* (FPP) on the solution structure. Under an FPP, customers are partitioned into disjoint and collectively exhaustive clusters that are served independently (Anily and Federgruen 1990). Applying FPPs to the IRP has the potential to provide substantial practical benefits. By partitioning customers into separate clusters that remain fixed in time, the organizational structure of the distribution plan is immensely simplified so that each cluster is managed as if it were an independent network. Also, this partitioning helps make the network more robust because it isolates the effects of these disruptions only to the affected cluster. In addition, partitioning allows the network operator to assign drivers to clusters in as consistent a fashion as possible over the planning horizon, thus helping the driver develop familiarity with the geographical region that the driver has to serve as well as with customers. This assignment can be done a posteriori after clustering in an efficient manner as we show later.

In order to achieve this, we formally introduce, in this paper, a variant of the IRP that we call the fixed-partition policy inventory routing problem (FPP-IRP). The FPP has been studied before in the context of inventory routing problems. Over time, however, the assumptions under which inventory routing problems are solved have changed significantly since these early attempts (see Coelho, Cordeau, and Laporte 2014). Currently, the standard (and more challenging) form of the IRP has the following features:

- Variable and nonhomogenous demand rates.
- Capacitated storage facilities at customer sites.
- Bounded supply from the depot.
- Bounded vehicle capacities.
- No bounds on the number of visits to customers over the planning horizon (apart from allowing only one visit per day).
- No stock-outs.
- No split deliveries (i.e., each customer is served by at most one vehicle in each time period).

These settings result in a problem that is highly combinatorial in nature on both the routing and inventory management fronts and is, therefore, very difficult to solve. Most previous works on the IRP under FPP assume away some of these complicating conditions to make the problem more analytically and computationally tractable. For example, by assuming a constant demand rate for each customer, some aspects of the problem can be reduced to economic order quantity models that are easy to solve (Anily and Federgruen 1990; Anily and Bramel 2004b, a; Raa and Aghezzaf 2009; Zhao, Wang, and Lai 2007). Also, almost every previous article assumes that an infinite supply of product is available at the depot, which eliminates a critical knapsack-like combinatorial element from inventory decisions in addition to decoupling the problem for a given partition (see the previous references in addition to Chan, Federgruen, and Simchi-Levi 1998; Gaur and Fisher 2004; and Li, Chu, and Chen 2011). To the best of our knowledge, this current paper is the first to investigate FPPs for the IRP as defined herein.

In this work, we propose FPP-IRP formulations and analyze them computationally using a widely used IRP benchmark instance set. The contributions of our paper can be summarized as follows:

- We introduce two FPP-IRP paradigms: the *rigid FPP-IRP* (rFPP-IRP), in which both clusters and the route within each cluster are fixed, and the *flexible FPP-IRP* (fFPP-IRP), in which only the clusters are fixed, but routing can be arbitrary within them.
- We define different classes of valid inequalities for both problems.
- We develop a branch-and-cut algorithm for both problems.
- We provide a worst-case analysis of FPP policies with respect to the optimal solution of the IRP.
- We perform extensive computational tests with the aim of discerning the performance of the branch-and-cut algorithm.
- We analyze the differences in solution structure between the IRP and the FPP-IRP.
- We compare the FPP-IRPs to formulations that impose strict consistency conditions and demonstrate the nature of the trade-off between cost and consistency.

The rest of this paper is organized as follows. Section 2 provides a formal description of the problem and of the FPPs proposed. Formulations for the different policies are provided in Section 3 together with different classes of valid inequalities. A worst-case analysis is presented in Section 4, and the branch-and-cut algorithm used to solve the formulations presented in Section 3 is described in Section 5. Computational results are presented in Section 6, and Section 7 contains an extensive discussion of the consistency aspects of the various FPPs considered. Finally, conclusions are drawn in Section 8.

2. Problem Definition

We now describe the IRP and then introduce the FPPs considered, namely the rFPP and the fFPP.

The IRP is defined over an undirected graph $G = (N, E)$ with $N = \{0\} \cup N'$, where node 0 is the depot and nodes in $N' = \{1, \ldots, n\}$ represent the customers. The elements of E are denoted (i, j) with $i < j$ (as the graph is undirected). A planning horizon $T = \{1, \ldots, H\}$ of H time periods is considered. Each customer $i \in N'$ is associated with an initial inventory level I_i^0, an inventory capacity U_i, and a consumption rate r_i^t of a single product for each $t \in T$. The depot has no inventory capacity limit and is associated with a production rate r_0^t for each $t \in T$. Each edge $(i, j) \in E$ represents a link between locations i and j and is associated with a nonnegative cost c_{ij}. We assume that costs c_{ij} satisfy the triangle inequality. A cost of h_0 and h_i is associated with holding a unit of inventory at the depot and at customer i, respectively. Distribution from the depot to customers is performed through a fleet $K = \{1, \ldots, m\}$ of m homogeneous vehicles, each with capacity Q. The objective of the IRP is to determine a distribution plan defining, for each time period, which customers are served, what the quantity delivered to each of them is, and what the routes performed by the vehicles are, in such a way that

- Vehicle capacity is not exceeded.
- Customer inventory capacity is not exceeded.
- No stock-out occurs at the customers or the depot.
- The total cost is minimized, and the total cost includes transportation costs and inventory holding costs at the customers and at the depot.

This setting of the problem is also known in the literature as the IRP with a maximum level policy; that is, a replenishment policy that establishes that any quantity can be delivered on any visit provided that the customer's inventory capacity is not exceeded.

The idea of FPPs is to create fixed clusters of customers that are served independently during the entire planning horizon. We consider two FPPs. The first policy, which we call rFPP, is such that, not only are the clusters fixed, but so is the route within each cluster. Consequently, if a customer in a cluster is visited in a particular time period, then so are all other customers in that cluster. Note that because costs c_{ij} do not depend on time, once the cluster is fixed, the route performed by the vehicle is always the same, that is, the cost of the traveling salesman problem (TSP) on these customers and the depot. We call the corresponding problem the rFPP-IRP. The second policy, which we call fFPP, is such that the clusters are fixed but routes within each cluster are allowed to be arbitrary; that is, they may contain any subset of customers as needed and may vary with time. We call the corresponding problem the fFPP-IRP.

3. Mathematical Formulations

In the following, we provide mathematical formulations for the rFPP-IRP and the fFPP-IRP, respectively. We wll use the notation $E' = \{(i,j) \in E : i, j \in N'\}$, $E(S) = \{(i,j) \in E : i \in S, j \in S\}$, and $\delta(S) = \{(i,j) \in E : (i \in S, j \notin S) \text{ or } (i \notin S, j \in S)\}$. In the rest of the text, we refer to variables via both individually indexed notation (e.g., y_{ij}^{kt}) and unindexed aggregated notation (e.g., y) as appropriate.

3.1. Basic IRP Formulation

We begin by presenting the mathematical formulation of the classical IRP problem. This serves as a basis for all other formulations we consider in this paper.

The basic IRP is formulated using the following variables:

- I_i^t: Continuous variable indicating the inventory level at node $i \in N$ at the end of time period $t \in T$.

- z_i^{kt}: Binary variable equal to one if node $i \in N$ is visited at time period $t \in T$ by vehicle $k \in K$ and zero otherwise.
- q_i^{kt}: Continuous variable representing the quantity delivered to customer $i \in N'$ in time period $t \in T$ by vehicle $k \in K$.
- y_{ij}^{kt}: Integer variable representing the number of times the edge $(i,j) \in E$ is traversed by vehicle $k \in K$ in time period $t \in T$; y_{ij}^{kt} can take a value of two only if edge (i,j) is incident to the depot and forms a one-customer route, meaning that it is traversed twice.

The formulation is as follows:

$$\min \sum_{t\in T} h_0 I_0^t + \sum_{i\in N'}\sum_{t\in T} h_i I_i^t + \sum_{k\in K}\sum_{(i,j)\in E}\sum_{t\in T} c_{ij} y_{ij}^{kt}, \tag{1a}$$

$$\text{s.t. } I_0^t = I_0^{t-1} + r_0^t - \sum_{k\in K}\sum_{i\in N'} q_i^{kt} \quad t\in T, \tag{1b}$$

$$I_i^t = I_i^{t-1} - r_i^t + \sum_{k\in K} q_i^{kt} \qquad i\in N',\ t\in T, \tag{1c}$$

$$I_i^t \geqslant 0 \qquad i\in N,\ t\in T, \tag{1d}$$

$$\sum_{k\in K} q_i^{kt} \leqslant U_i - I_i^{t-1} \qquad i\in N',\ t\in T, \tag{1e}$$

$$q_i^{kt} \leqslant U_i z_i^{kt} \qquad i\in N',\ k\in K,\ t\in T, \tag{1f}$$

$$\sum_{i\in N'} q_i^{kt} \leqslant Q z_0^{kt} \qquad k\in K,\ t\in T, \tag{1g}$$

$$\sum_{k\in K} z_i^{kt} \leqslant 1 \qquad i\in N',\ t\in T, \tag{1h}$$

$$\sum_{j\in N:(i,j)\in E} y_{ij}^{kt} + y_{ji}^{kt} = 2 z_i^{kt} \qquad i\in N,\ k\in K,\ t\in T, \tag{1i}$$

$$\sum_{(i,j)\in E(S)} y_{ij}^{kt} \leqslant \sum_{i\in S} z_i^{kt} - z_s^{kt} \quad s\in S\subseteq N',\ k\in K, t\in T, \tag{1j}$$

$$z_i^{kt} \in \{0,1\} \qquad i\in N,\ k\in K,\ t\in T, \tag{1k}$$

$$q_i^{kt} \geqslant 0 \qquad i\in N',\ k\in K,\ t\in T, \tag{1l}$$

$$y_{ij}^{kt} \in \{0,1\} \qquad (i,j)\in E',\ k\in K,\ t\in T, \tag{1m}$$

$$y_{0j}^{kt} \in \{0,1,2\} \qquad (0,j)\in E,\ k\in K,\ t\in T. \tag{1n}$$

The objective function (1a) calls for the minimization of the total cost, which is the sum of inventory holding costs at the depot, inventory holding costs at the customers, and transportation costs of routes over the time horizon. Constraints (1b)–(1d) determine the evolution of inventory levels over time and force the absence of stock-out situations at the supplier and at the customers. Constraints (1e) and (1f) ensure the policy imposing that, if a customer is visited, the quantity delivered is such that the maximum inventory level is not exceeded. Constraints (1g) are vehicle capacity constraints. Constraints (1h)–(1j) are routing constraints: constraints (1h) impose visiting each customer at most once in each time period, constraints (1i) are degree constraints for each node and each vehicle in each time period (i.e., if a node is visited, then there must be an entry edge and an exit edge), and (1j) are the subtour elimination constraints for each vehicle route and each time period. Finally, (1k)–(1n) define the domains of the variables.

3.2. rFPP-IRP Formulations

With the basic IRP defined, we can now present the various fixed-partition policy formulations, beginning with the rFPP-IRP. In addition to the preceding variables, the formulation for the rFPP-IRP makes use of the following variables:

- x_{ij}: Integer variable equal to one or two if edge (i,j) is "open"—that is, can be traversed by vehicles—and zero otherwise; x_{ij} can take a value of two only if edge (i,j) is incident to the depot and forms a one-customer route.

The rFPP-IRP can be formulated as follows:

$$\min \ (1a)$$

$$\text{s.t. } (1b) \text{ to } (1n)$$

$$y_{ij}^{kt} \leqslant x_{ij} \qquad (i,j)\in E, k\in K, t\in T, \tag{2a}$$

$$\sum_{j\in N':(i,j)\in E} x_{ij} + x_{ji} = 2 \qquad i\in N', \tag{2b}$$

$$x_{ij} \in \{0,1\} \qquad (i,j)\in E', \tag{2c}$$

$$x_{0j} \in \{0,1,2\} \qquad (0,j)\in E. \tag{2d}$$

Constraints (2a) and (2b) define the rFPP policy. In particular, constraints (2a) stipulate that no vehicle may be routed on edge (i,j) in any time period unless the edge is open; that is, variables x define edges that are available for routing, and a y_{ij}^{kt} variable can be positive (i.e., vehicle k can traverse edge (i,j) at time t) only if x_{ij} is positive. Constraints (2b) mandate that, for every customer, there is one open entering edge and one open leaving edge, meaning that every customer must be involved in some route. Note that constraints (2b) coupled with (1j) and (2a) imply the fact that x_{ij} forms routes in addition to defining clusters.

Note that, because x fully defines all routes in this formulation, subtour elimination constraints (1j) may be substituted by the following:

$$\sum_{(i,j)\in\delta(S)} x_{ij} \geqslant 2 \quad S\subseteq N'. \tag{2e}$$

These are subtour elimination constraints defined on the x variables and dictate that, for any two-set partition of some subset of customers, there must be at least two open edges linking the two sets so that a set may be entered and exited. Note that edges incident to the depot can perform both functions simultaneously.

3.3. fFPP-IRP Formulations

Two compact formulations for the fFPP-IRP are proposed.

3.3.1. First Formulation. Let us define W as the set of clusters created. Note that an upper bound on $|W|$ is $\min\{mH, n\}$. To see this, first note that n customers can be partitioned into, at most, n nonempty clusters. Also, because the fleet size is m, at most m clusters may have vehicles deployed to them in any given time period. Thus, by the pigeonhole principle, at most mH different clusters can be used over the planning horizon.

We define the following variables:

- v_{iw}: Binary variable equal to one if customer $i \in N'$ is inserted into cluster $w \in W$. These variables define the clustering of customers.

The formulation is the following:

min (1a)

s.t. (1b) to (1n)

$$y_{ij}^{kt} \leqslant 1 - v_{iw} + v_{jw} \quad (i,j) \in E,\ k \in K,\ t \in T \in W, \tag{3a}$$

$$y_{ij}^{kt} \leqslant 1 + v_{iw} - v_{jw} \quad (i,j) \in E,\ k \in K,\ t \in T,\ w \in W, \tag{3b}$$

$$v_{iw} + v_{jw} \leqslant 3 - \left(z_i^{kt} + \sum_{k' \in K \setminus \{k\}} z_j^{k't}\right) i,j \in N',\quad k \in K,\ t \in T,\ w \in W, \tag{3c}$$

$$\sum_{w \in W} v_{iw} = 1 \quad i \in N', \tag{3d}$$

$$v_{iw} \in \{0,1\} \quad i \in N',\ w \in W. \tag{3e}$$

Constraints (3a)–(3d) define the flexible clustering policy. In particular, a customer is assigned to a single cluster through (3d). Constraints (3a) and (3b) ensure that edge (i,j) is traversable only when i and j are both assigned to the same cluster. Constraints (3c) impose that customers who are assigned to the same cluster are visited by the same vehicle.

3.3.2. Second Formulation. This formulation is similar to the rFPP-IRP formulation presented in Section 3.2. It makes use of exactly the same sets of variables.

The formulation is the following:

min (1a)

s.t. (1b) to (1n)

$$y_{ij}^{kt} \leqslant x_{ij} \quad (i,j) \in E,\ k \in K,\ t \in T, \tag{4a}$$

$$x_{ij} \leqslant 2 - \left(z_i^{kt} + \sum_{k' \in K: k' \neq k} z_j^{k't}\right) \quad (i,j) \in E,\ k \in K,\ t \in T, \tag{4b}$$

$$x_{ij} \in \{0,1\} \quad (i,j) \in E', \tag{4c}$$

$$x_{0j} \in \{0,1,2\} \quad (i,j) \in E. \tag{4d}$$

Constraints (4a) are identical to (2a). Constraints (4b) substitute (2b) and allow a flexible clustering. In particular, variables x again identify open edges but, contrary to formulation (2), variables x in formulation (4a) do not identify vehicle routes. Rather, constraints (4b) state that, if edge (i,j) is open, then i and j must be served by the same vehicle in time period t. Because x_{ij} does not contain a time index, this, in turn, means that i and j belong to the same (fixed) cluster. Thus, the routes are identified by the y variables and can visit any subset of customers belonging to a cluster. Note that there is no restriction on the number of open edges incident to each node.

3.4. Valid Inequalities

Different valid inequalities can be inherited from IRP (see Archetti et al. 2007, 2014). We now focus on new ones, either related to FPPs or not previously proposed for the IRP.

3.4.1. Number of Routes. The following inequalities fix a lower bound on the minimum number of routes that need to be performed in order to serve all customers. They are valid for the general IRP. They are formulated as follows:

$$\sum_{j \in N': (0,j) \in E} \sum_{k \in K} \sum_{t \in T} y_{0j}^{kt} \geqslant \left\lceil \frac{\sum_{i \in N}\left(\sum_{t \in T} r_i^t - I_i^0\right)}{Q} \right\rceil. \tag{5}$$

This inequality establishes that the number of routes is at least equal to the ratio of the total demand to the capacity of a vehicle.

3.4.2. Number of Clusters. The following inequalities establish an upper bound on the maximum number of clusters:

$$\sum_{j \in N': (0,j) \in E} x_{0j} \leqslant mH. \tag{6}$$

This bound is useful when mH is far smaller than n, the number of customers, which is a trivial upper on the number clusters.

3.4.3. Infeasible Clusters. The following two classes of valid inequalities are related to subsets of customers that cannot be part of the same cluster. Let $\tilde{r}_i^t = (\sum_{t \in T} r_i^t - I_i^0)/H$, the average daily demand of customer i. Consider a subset of customers $S \subseteq N'$ such that $\sum_{i \in S} \tilde{r}_i^t > Q$. Then,

$$\sum_{(i,j) \in E(S)} x_{ij} \leqslant |S| - 2. \tag{7}$$

This is because, if $\sum_{i \in S} \tilde{r}_i^t > Q$, then even visiting customers in S every day of the planning horizon would not be sufficient to deliver the quantity they need if they are all inserted in the same cluster (and, thus, visited by a single vehicle in each time period).

Thus, we exclude any path linking them. Note that inequalities (7) may be also written as

$$\sum_{(i,j)\in\delta(S\cup\{0\})} x_{ij} \geqslant 2. \tag{8}$$

Let us now consider a subset $S \subseteq N'$ such that $\sum_{i\in N'\setminus S} \tilde{r}_i^t > (m-1)Q$. In this case, the daily demand of customers $N'\setminus S$ is larger than $(m-1)Q$; thus, customers in S cannot absorb a cluster and must be connected to other customers. We have

$$\sum_{(i,j)\in\delta(S)\setminus\{0\}} x_{ij} \geqslant 2. \tag{9}$$

Note that inequalities (7)–(9) may be seen as capacity cuts for the classical vehicle routing problem (see Ralphs et al. (2003) for further details).

3.4.4. Coherence Inequalities. The following inequalities link z and x variables:

$$x_{ij} \leqslant 1 - \left(z_i^{kt} - z_j^{kt}\right) \quad (i,j) \in E', t \in T, k \in K, \tag{10}$$

$$x_{ij} \leqslant 1 - \left(z_j^{kt} - z_i^{kt}\right) \quad (i,j) \in E', t \in T, k \in K. \tag{11}$$

These inequalities state that, if a vehicle visits i but not j in time period t, then the link between them cannot be traversed. Note that these inequalities are consistent only with the rFPP-IRP formulation because fFPP allows customers within a cluster to be visited selectively in different time periods.

3.4.5. Symmetry-Breaking Constraints. The following inequalities prioritize the use of vehicles with smaller indices (inequalities (12)), and assign the currently unassigned customer with the smallest index to the lowest-index vehicle (inequalities (13)):

$$z_0^{kt} \geqslant z_0^{k+1,t} \qquad k \in K\setminus\{m\},\ t \in T. \tag{12}$$

$$\sum_{i=1}^{j} 2^{j-i} z_i^{kt} \geqslant \sum_{i=1}^{j} 2^{j-i} z_i^{k+1,t} \qquad j \in N', k \in K\setminus\{m\}, t \in T. \tag{13}$$

The same inequalities are also used in Archetti et al. (2014).

4. Worst-Case Analysis

In this section, we present a worst-case analysis comparing the solution of the IRP with either the rFPP-IRP or the fFPP-IRP. Moreover, we compare the rFPP-IRP with the fFPP-IRP.

Let $z(P)$ be the value of the optimal solution of a given problem P.

Theorem 1. *Given an instance $\mathcal{I}$ that is feasible for the IRP, the rFPP-IRP and fFPP-IRP may not admit a feasible solution for instance $\mathcal{I}$.*

Proof. Consider the following instance $\mathcal{I}$: $n = 3$, $H = 2$, $m = 2$, $I_i^t = 0$, and $U_i = Q$ for $i \in N'$. Suppose that customer demands are as follows: $r_{i1} = Q$ for $i = 1,2$; $r_{31} = 0$; $r_{i2} = Q/2$ for $i = 1,2$; and $r_{32} = Q$. The only feasible solution for the IRP is visiting customers 1 and 2 on day 1 with two vehicles, and on day 2, one vehicle visits customers 1 and 2 in a single route and the second vehicle visits customer 3. Looking at the clustering on day 2, we notice that customers 1 and 2 are visited together, so they form a cluster. Note that no other clustering is feasible for day 2. However, cluster $\{1,2\}$ is not feasible for serving customers 1 and 2 on day 1 as each of them needs to receive a quantity equal to Q, and they, thus, must be visited separately. Thus, there exists no feasible solution for the rFPP-IRP or fFPP-IRP for instance $\mathcal{I}$. □

Theorem 2. *The optimal value of the rFPP-IRP relative to the fFPP-IRP is bounded by the inequality z(rFPP-IRP)/z(fFPP-IRP)$\leqslant H$, and the bound is tight.*

Proof. Any cluster C generated in the fFPP-IRP is a feasible cluster for the rFPP-IRP. Note that clusters defined in the fFPP-IRP only determine customers that are visited in the rFPP-IRP. However, visiting a customer does not imply that a positive quantity is delivered. Each cluster can be used at most H times. The cost of visiting all customers in the cluster corresponds to the value of the optimal solution of the TSP over all customers in the cluster (including the depot). Let us call this $z(\mathrm{TSP}(C))$. Let us also denote as $c(C)$ the cost related to cluster C in the fFPP-IRP solution; that is, the sum of the transportation costs of the routes visiting customers in C over the entire planning horizon. Note that $c(C)$ is greater than or equal to $z(\mathrm{TSP}(C))$. Let us now construct a feasible solution for the rFPP-IRP that mirrors that of fFPP-IRP by visiting all customers in cluster C when at least one customer in C is visited in fFPP-IRP for all C. Quantities delivered are the same, so the inventory costs are the same in both policies. Suppose inventory costs are zero. Let $\mathcal{C}$ be the set of all clusters formed in the fFPP-IRP solution. Now, $z(\text{fFPP-IRP}) = \sum_{C\in\mathcal{C}} c(C) \geqslant \sum_{C\in\mathcal{C}} z(\mathrm{TSP}(C))$. The cost of the feasible rFPP-IRP solution is not greater than $H\sum_{C\in\mathcal{C}} z(\mathrm{TSP}(C))$. Thus, $z(\text{rFPP-IRP})/z(\text{fFPP-IRP}) \leqslant H$.

To show that the bound is tight, let us consider the following instance $\mathcal{I}$: $m = 2$, $n = Q$, $I_i^t = 0$, $h_i = 0$ for $i \in N'$, $U_i = 1$ for $i \in N'\setminus\{1,2\}$, and $U_i = Q/2 + 1$ for $i \in \{1,2\}$. Customer demands are as follows: $r_{1t} = r_{2t} = Q/2 + 1$ for $t \in T$, $r_{i1} = 1$ and $r_i^t = 0$ for $i \in N'\setminus\{1,2\}$ and $t \in T$, $c_{ij} = 1$ for all $i, j \in N\setminus\{1,2\}$, $c_{01} = c_{02} = c_{12} = \epsilon$, $c_{1i} = c_{2i} = 1$, for all $i \in N'\setminus\{1,2\}$. All customers need to be visited on day 1. Thus, two fully loaded vehicle routes visit all customers at day 1. One route visits customer 1, delivering $Q/2 + 1$ units, and $Q/2 - 1$ customers in $N'\setminus\{1,2\}$, delivering one unit each. The second route visits customer 2, delivering $Q/2+1$ units, and the remaining $Q/2-1$ customers in $N'\backslash\{1,2\}$, delivering one unit each. The cost of these two routes is $n+2\epsilon$. This forces the creation of two clusters including all customers. In the following days, the fFPP-IRP visits only customers 1 and 2 in two different routes with out-and-back routes of cost 2ϵ each. Thus, $z(\text{fFPP-IRP}) = n+2\epsilon+4\epsilon(H-1)$. Instead, the rFPP-IRP is also forced to visit all customers in days $2\ldots H$. Thus, $z(\text{rFPP-IRP}) = H(n+2\epsilon)$. The ratio is then $(n+2\epsilon)H/(n+2\epsilon+4\epsilon(H-1))$ which tends to H for $\epsilon \to 0$. □

5. Branch-and-Cut Algorithm

All formulations are solved through a branch-and-cut algorithm. In particular, subtour elimination constraints (1j) and (2e) are initially relaxed and inserted only once violated through the min-cut algorithm proposed in Padberg and Rinaldi (1991) in the classical fashion implemented in TSP formulations. We note quickly that the right-hand sides of constraints (1j) have become standard in the IRP literature, and they replace the set cardinality term usually found in generalized subtour elimination constraints because the inequalities are delineated by vehicles, and the node counts in a set only if it is visited by that vehicle in its route. Inequalities (5) and (6) and (10)–(13) are directly added to any applicable formulation as they are polynomial in number. Inequalities (7)–(9) are dynamically inserted once violated. In particular, we use the extended and greedy heuristics proposed in Ralphs et al. (2003). To quickly summarize these,

- The *extended shrinking heuristic* isolates the graph supported by the linear relaxation solution, and flows on the edges of this graph are induced by this solution. For each pair of customer nodes i and j on this graph, if a cut that isolates them violates (7)–(9), then the inequality is added to the formulation. Then an edge associated with the maximum flow between i and j is identified, and the two nodes linked by this edge are collapsed into one supernode, and the flow between them is aggregated. Then, the process is repeated on this shrunk graph.
- The *extended greedy heuristic* starts with some subset S of customers on the supported graph (chosen either at random or according to some rule), and then, this set is grown by adding customers in a greedy fashion maximizing the flow into S defined by the linear program solution. If the cut between S and the rest of the graph violates the capacity constraints, it is added to the formulation.

6. Computational Results and Analysis of Performance

In this section, we present comprehensive computational results for each of the formulations considered based on simulations performed using the instance set introduced in Archetti et al. (2007). The set consists of 160 combinations of categories formed by the Cartesian product of (i) 10 categories for the number of customers given by $n \in \{5, 10, \ldots, 50\}$, (ii) four categories for the number of vehicles given by $m \in \{2,3,4,5\}$, (iii) two categories for the length of the planning horizon given by $H \in \{3,6\}$, and (iv) two categories for the size of the holding costs given by H and L (indicating high and low). For each combination, five random instances are generated for a total of 800 instances. All formulations considered were coded in IBM's CPLEX 12.9 in C++ on an Intel Core i7-7700HQ clocked at 2.8 GHz with 24 GB of RAM. All runs are executed single-threaded, and all running times are capped at two hours.

The rFPP-IRP formulation admits two implementations of the subtour elimination constraints: one in which they are applied to every vehicle in every time period individually (i.e., to the variables y_{ij}^{kt}) as in (1j) and another in which they are applied to the route of open edges (i.e., to the variables x_{ij}) as in (2e). As we note earlier, this choice is not available for the fFPP-IRP. We investigate all these options. We also describe some features of the optimal solutions reached by each of the considered formulations to shed light on differences among them. Finally, we discuss the effectiveness of the valid inequalities used.

6.1. Computational Performance

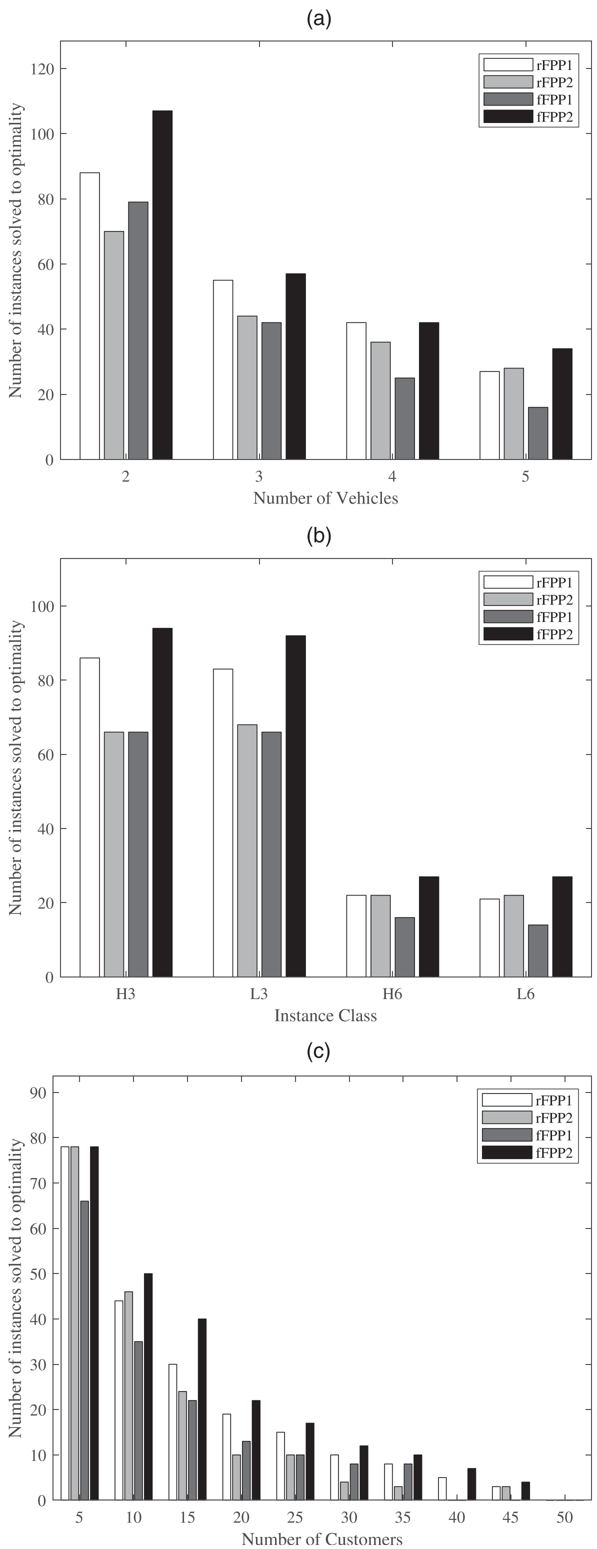
In this section, we compare and contrast how the considered formulations perform based on various computational metrics. Specifically, for each formulation, we look at three areas: (i) the number of instances solved to optimality within the two-hour running time limit, (ii) the degree of suboptimality that the fixed-partition policies introduce relative to the pure IRP, and (iii) the average computational time needed to solve instances to optimality. For each of these, we present three sets of results corresponding to average results that are separated by the number of vehicles used, the class of the instance, and the number of customers in the network. In some cases, further results are included. Four implementations are considered, code named in the following manner:

- rFPP1: The rFPP-IRP formulation in which subtour elimination constraints are applied to the y variables as in (1j).
- rFPP2: The rFPP-IRP formulation in which subtour elimination constraints are applied to the x variables as in (2e).
- fFPP1: The fFPP-IRP formulation in which clusters are defined using the v variables as in Section 3.3.1.
- fFPP2: The fFPP-IRP formulation in which clusters are defined using the x variables as in Section 3.3.2.

When results are presented without identifying the specific implementation, we simply say "rFPP" and "fFPP."

6.1.1. Number of Instances Solved to Optimality. Figure 1 presents results for the number of instances solved to optimality by each formulation within the two-hour limit. In general, instances with more vehicles, more customers, and a larger optimization horizon are harder to solve, which is readily explained by the increase in the size of the solution space. Further, the results show that the fFPP-IRP with clusters on x solves more instances, both in total and in every individual category. As for the rFPP-IRP, the results indicate that defining subtour elimination constraints on y is the superior option based on this metric with an additional 34 instances in total being solved compared with rFPP2. This may seem counterintuitive because one would expect that the quick resolution of the x variables via (2e) would aid in resolving a large number of y variables via (2a). A plausible explanation is that, contrary to x variables, the y variables are part of the objective function, and so any variation in their value has a direct impact on the solution value and the lower bound.

Figure 1. Number of Instances Solved to Optimality by (a) Number of Vehicles, (b) Instance Class, (c) Number of Customers



Another interesting observation is that, despite instances of size $n = 5$ being trivial, no formulation solves all 80 instances available, indicating that there might be feasibility issues when the additional constraint of fixed partitions is imposed. Thus, it is instructive to look into the ability of each formulation to find a feasible solution for each instance. In fact, it turns out that a large number of instances yield no feasible solution at the end of the two-hour limit as seen in Table 1. This might be because no feasible solution exists (see Theorem 1). However, determining whether the rFPP-IRP or fFPP-IRP admit feasible solutions for a given instance is a difficult task. In fact, the corresponding problem is NP-hard as it can be reduced to the single-item capacitated lot-sizing problem, which is NP-hard (see Florian, Lenstra, and Rinnooy Kan 1980; Bitran and Yanasse 1982).

6.1.2. Percentage Increase in Optimal Values Relative to the Pure IRP. For the purposes of this study, one of the most important metrics by which to evaluate the considered FPPs is the degree of suboptimality they introduce relative to the pure IRP formulation. As a reminder, an FPP is an additional constraint introduced into the pure IRP restricting the structure of allowed routes. Thus, the suboptimality metric quantifies the cost of imposing some structure on the operational logistics of the network. We also note that the degree of suboptimality is expected to be much lower for the fFPP compared with the rFPP. This is because the fFPP can itself be thought of as a relaxation of the

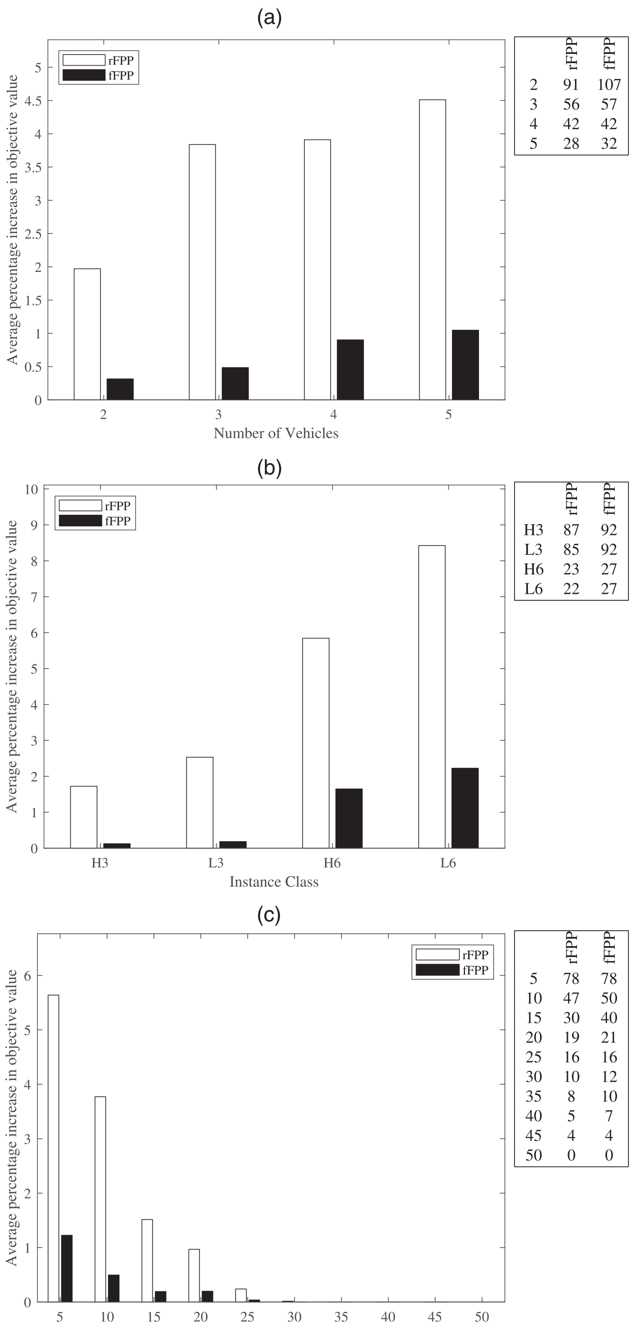
Table 1. Number of Instances in Which No Feasible Solution Was Found at Termination

rFPP1	rFPP2	fFPP1	fFPP2
413	389	514	304

rFPP because, once customers are partitioned, fFPPs may route arbitrarily on the cluster, and the rFPP must commit to a single unchanging route on each cluster.

Figure 2 shows averages for the percentage increase in the optimal objective value per instance of the considered FPP relative to that of the pure IRP. In order to keep the test fair, we only consider cases in which optimal objective values are available for both the pure IRP and the considered FPP (the pure IRP is solved independently). For each FPP, we use whichever version achieves an optimal solution (if either does). The number of such instances is shown in the table next to each graph. When no such instances are found, no average is presented.

Figure 2. Percentage Increase in Objective Values of Each Formulation Relative to the Pure IRP for Instances in Which Both Formulations Are Solved to Optimality by (a) Number of Vehicles, (b) Instance Class, (c) Number of Customers



(a)	rFPP	fFPP
2	91	107
3	56	57
4	42	42
5	28	32

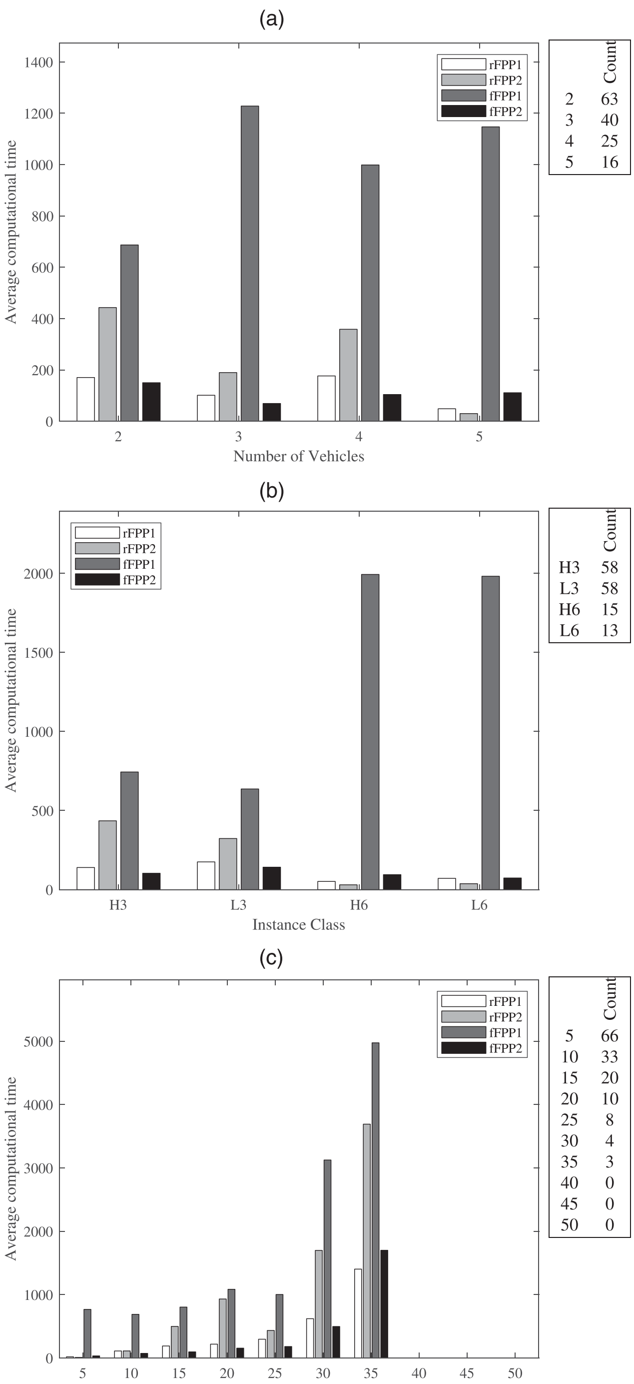
(b)	rFPP	fFPP
H3	87	92
L3	85	92
H6	23	27
L6	22	27

(c)	rFPP	fFPP
5	78	78
10	47	50
15	30	40
20	19	21
25	16	16
30	10	12
35	8	10
40	5	7
45	4	4
50	0	0

We do indeed see that the average percentage increases are far larger for the rFPPs relative to the fFPPs. An interesting observation, however, is that the degree of suboptimality is small across the board with a maximum increase of 9% for the rFPP and a mere 2% for the fFPP. In addition, we see the degree of suboptimality falling very sharply to nearly zero as the size of the network gets large. From a practical point of view, this is a very encouraging sign because it suggests that imposing some structure that simplifies operations is not punitive in terms of added cost and might, therefore, be a worthwhile choice for operators. Interestingly, we observe also that the cost increase is large when the optimization horizon is long, perhaps indicating that the IRP's flexibility advantage is better realized when there are more time periods with which to work.

6.1.3. Average Computational Time. Finally, we evaluate average running times for each FPP. As noted, the vast majority of instances terminate prior to reaching optimality, meaning that looking at a simple running time average would not be very helpful because it would be very close to two hours always and it would be difficult to discern any useful trends from these results. Instead, we look at averages only for instances solved to optimality. To keep the averages meaningful, we restrict the comparison with instances solved to optimality by all four FPPs simultaneously. These results are shown in Figure 3 with the sample size in each category shown in the accompanying tables. We notice straightaway that fFPP2 is, by far, the better fFPP implementation, and rFPP1 is the faster implementation for the rFPP in most bins. Formulation fFPP1 is clearly the slowest of the four implementations, indicating that the open-edge encoding of a cluster via the x variables is a better modeling choice than clustering via variables v. We also see that rFPP1 and fFPP2 perform comparably relative to each other, which, together with their lower running times, might suggest that their shared constraint set (i.e., (2a) and (4a)) provides strong inequalities. Not surprisingly, instances with more customers and longer horizons are harder to solve, but the trend does not carry to the fleet size with $m = 5$ instances solving slightly faster than ones with $m = 4$ although this might not be a robust conclusion given the sample size.

Figure 3. Average Computational Time for Instances Solved to Optimality by All Four Formulations by (a) Number of Vehicles, (b) Instance Class, (c) Number of Customers

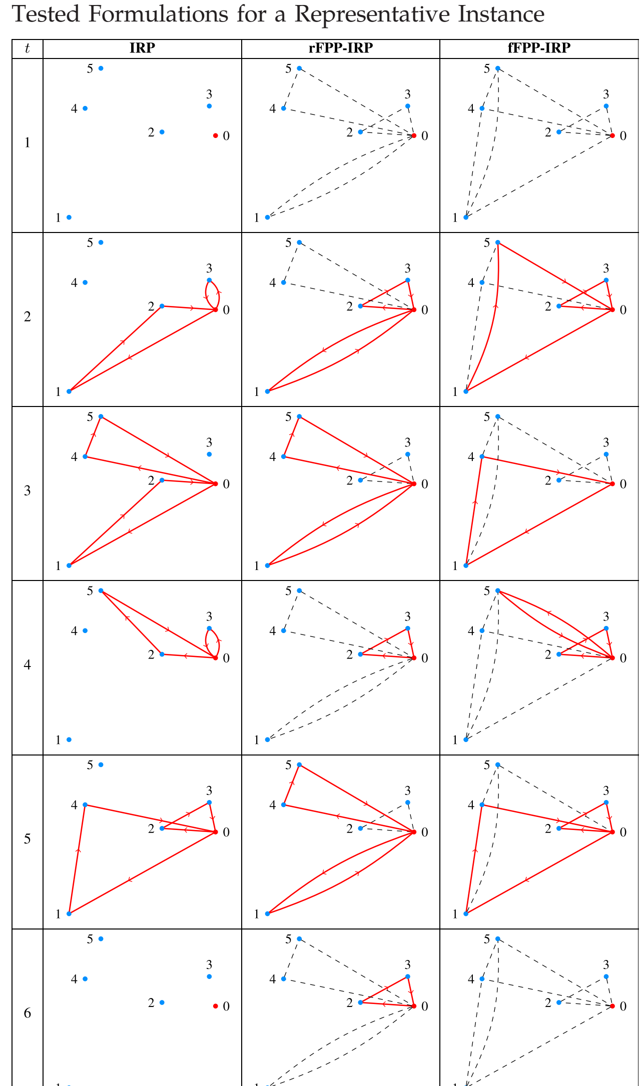


6.2. Differences in Route Structures

Figure 4 shows an illustration of the type of solutions produced by each formulation. The output here is based on the instance abs3n5 with five customers, two vehicles, and type L6 (low costs and a horizon of six time periods). The figure shows the routes (based on variables y) that the solution decides upon in each time period between the depot (node 0) and customers (nodes 1 onward), which are shown as solid lines. In the case of rFPP-IRP and fFPP-IRP, we also show the links that the solution decides to open but not necessarily use in any given time period (variables x). These are depicted as dashed lines. For simplicity, we do not distinguish routes by the vehicles assigned to them.

The IRP is free to pick arbitrary routes in different times, and this is indeed what we observe in the left-hand column of panels. On the other hand, the rFPP picks three rigid potential routes that remain unchanged in time—namely $R_1 = 0 \to 1 \to 0$, $R_2 = 0 \to 2 \to 3 \to 0$, and $R_3 = 0 \to 4 \to 5 \to 0$—and then decides in each time period whether a route is used. In this case, the rFPP uses R_1 in periods 2, 3, and 5; R_2 in periods 2, 4, and 6; and R_3 in periods 3 and 5. Finally, the fFPP opens the set of links for which $x_{ij} = 1$, and implicit in this is the clustering ($\{1, 4, 5\}$ and $\{2, 3\}$ in

Figure 4. (Color online) Routing Outputs of the Three Tested Formulations for a Representative Instance



this example), which remains unchanged in time. The fFPP is then free to pick arbitrary routes on these clusters. For example, cluster $\{1, 4, 5\}$ deploys route $0 \rightarrow 1 \rightarrow 5 \rightarrow 0$ in time period 2, route $0 \rightarrow 1 \rightarrow 4 \rightarrow 0$ in time periods 3 and 5, and route $0 \rightarrow 5 \rightarrow 0$ in time period 4.

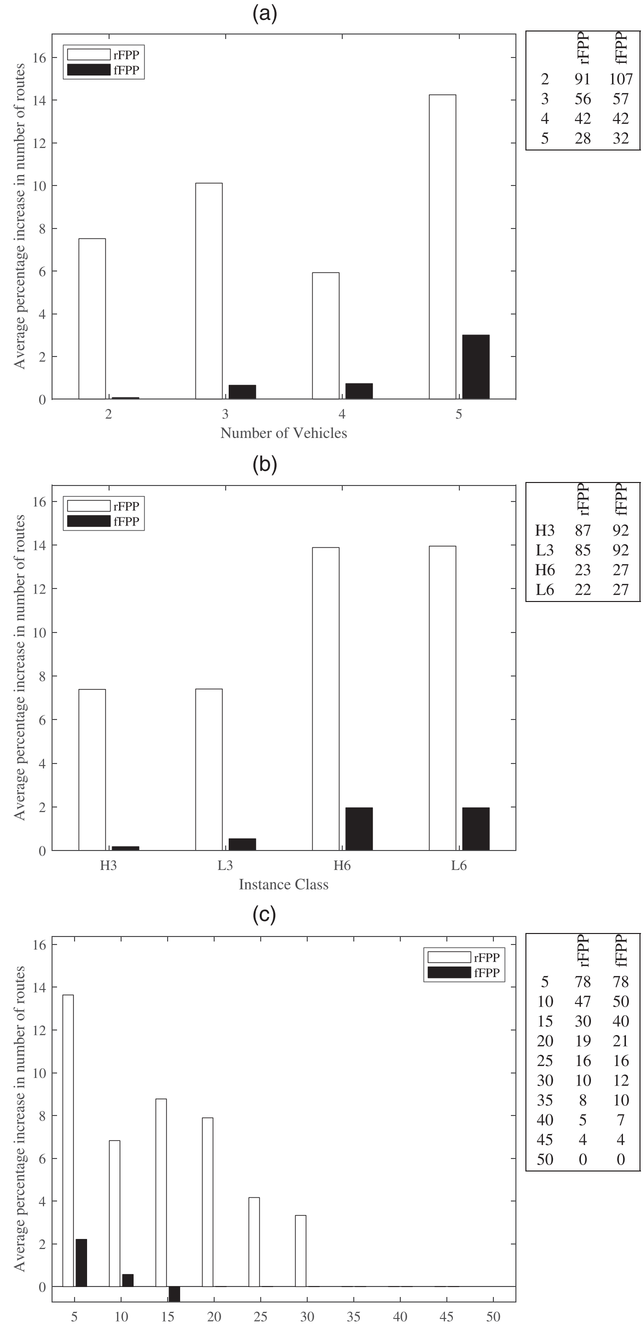
In order to further investigate the differences in underlying structure between formulations, we attempt to look at various statistics and indicators that can potentially shed light on this, including the total number of routes taken over the planning horizon, the average and variance of the number of routes taken by each vehicle, the average and variance of the number of routes taken on each day, the average and variance of the frequency of visitation of a customer, and the average customer inventory level in the system. With the exception of the total number of routes, none of these metrics reveals any discernible trends, which indicates that, apart from the basic structural change that the FPP introduces, not much else about the solution is different from the basic IRP in any significant way at least in terms of the indicators at which we look.

Figure 5 shows results for the percentage increase in the total number of routes used over the planning horizon by each FPP relative to the basic IRP. Results are compiled for cases in which both the IRP and the considered FPP are solved to optimality. For each FPP, we use whichever version achieves an optimal solution (if one is found). The results show that the rFPP in particular exhibits a clear and consistent increase in the number of routes used relative to the IRP. It is reasonable to attribute this to the rigid structure of the rFPP because, if two customers in different (fixed) clusters both require stock replenishment in a given time period, the formulation forces the deployment of two routes, which under the more lax IRP might have been consolidated into one route. This reasoning is further corroborated by the fact that the increase in route count of the fFPP is weak at best, perhaps owing to its less distant structure relative to the IRP.

6.3. Efficacy of Valid Inequalities

In this section, we discuss the effectiveness of the valid inequalities introduced in Section 3.4 in aiding better computational performance when solving the FPP-IRP. In the interest of reducing clutter, these results are presented in an aggregated form here; that is, we test the formulations once with and once without its respective complete set of applicable inequalities (recall that not every set of valid inequalities is applicable to every formulation). A total of 3,200 instances are tested—800 for each of the four considered formulations—and the results are presented in Table 2. After first stating the total number of instances solved to optimality under each condition, we present comparative results for performance indicators

Figure 5. Average Percentage Increase in Number of Routes Relative to the IRP for Instances Solved to Optimality by Both Formulations by (a) Number of Vehicles, (b) Instance Class, (c) Number of Customers



such as gap, best lower and upper bounds at termination, and running time when optimality is reached. In each case, the attribute "superior" used in the table is relative to the alternative condition; for example, the table indicates that 803 instances terminated with a better gap when valid inequalities were included than when the same instances were attempted without valid inequalities. Conversely, 238 instances terminated with a better gap when valid inequalities were

Table 2. Summary of Effects of Valid Inequalities on Performance

	Valid inequalities included	Valid inequalities excluded
Number of instances solved to optimality	792	642
Number of instances with superior gap at termination	803	238
Average percentage improvement in termination gap for instances with better gap	61.0	40.1
Number of instances with superior lower bound at termination	329	293
Average percentage improvement in lower bound for instances with superior lower bound	27.0	29.5
Number of instances with superior upper bound at termination	1,684	419
Average percentage improvement in lower bound for instances with superior upper bound (relative to the best upper bound)	7.2	3.3
Number of instances with superior running time when instances are solved to optimality under both conditions	342	173
Percentage improvement in running time when instances are solved to optimality under both conditions	57.6	22.3

excluded than when they were included. The percentage listed for each indicator is an average over the sample set in the row above; for example, in the case of the gap at termination, we compute the percentage reduction in termination gap with valid inequalities included relative to when they are excluded for each of the 803 cases in which this is the superior performer, and then, we average over this set. In each row, all instances not accounted for in the complete set of 3,200 instances exhibit identical performance on the given metric (instances in which no feasible solution is found are assigned an upper bound of ∞ and a gap of 100%).

Looking at the results, we see that the set of valid inequalities included yields more instances solved to optimality and also improves performance on all four considered metrics. In particular, the improvement in termination gap and best lower bound is drastic with almost four times as many instances attaining better values with these inequalities than without. Although it might be striking that some instances exhibit better performance without valid inequalities, this is not surprising—it is often the case with large, complex formulations that the inclusion of too many constraints leads to an increase in the computational overhead of parsing and processing these constraints, and the spacial requirements of encoding them can also be a strain on the solver. Nevertheless, we can see here that the inclusion of these inequalities leads to a better set of results in the aggregate.

7. Consistency

As noted in the introduction, the issue of service consistency is of great importance to operators of certain sectors of the delivery industry and is a consideration that has been gaining significant traction in the routing literature. Consistency is desirable for many reasons related to customer and driver satisfaction and quality of service, but it comes at the expense of higher operation costs because of the added structure imposed on the solution. An important avenue of analysis in our current work is to investigate the degree of consistency the FPPs achieve in the resulting routing plans as well as the cost implications of this modeling choice.

We are concerned primarily with two measures of route consistency. *Driver–customer consistency* (DCC) is an indicator of the degree of invariance of the vehicle that visits a given customer over the planning horizon, and *driver–route consistency* (DRC) measures the degree of invariance of the routes on which a driver is dispatched. These two metrics are reflective of the practical considerations highlighted earlier as driver–customer familiarity is linked to a heightened sense of customer satisfaction, leading to potential repeat business, and route consistency is associated with greater efficiency of execution of deliveries as well as greater job satisfaction from the driver's point of view.

In the context of the various FPP models introduced thus far, we can see that no explicit association between a driver and a customer or a route is made anywhere in the formulations. Yet it is intuitively clear that these models introduce some amount of consistency to the solution. The degree of this consistency is important to quantify. As a first step in this direction, we can see that perfect consistency may not be attained: to illustrate this, consider a problem with n customers and a fleet of $m < n$ vehicles. Perfect DRC implies a one-to-one correspondence between a route and a vehicle, and so at most m distinct clusters of customers may be created under this condition. On the other hand, an FPP may create as many as n clusters as long as at most m of them are served in any particular time period. This, in turn, means that some vehicles potentially have to be dispatched to different routes on different days, violating perfect DRC.

Thus, we can think of the FPPs as formulations that introduce a soft notion of consistency, and in this sense, they can loosely be considered relaxations of formulations that impose hard consistency requirements. The potential advantage of FPP formulations is the reduction in cost that results from this relaxation. On the other extreme is the IRP, which is cheaper than the FPPs as it is less restrictive but is predicted to exhibit less consistency. The rest of this section is dedicated to analyzing where FPPs lie on this spectrum.

7.1. Consistency Metrics

Before proceeding, we must first make the definitions of the consistency metrics concrete. Let κ_c and κ_r denote the DCC and DRC metrics, respectively.

7.1.1. DCC Metric. Given the solution (if found) of an instance of any of the formulations introduced thus far, let Z_i^k be a binary parameter equal to one if customer i is visited by vehicle k at any point over the horizon. In other words,

$$Z_i^k = \max_t z_i^{kt}. \tag{14}$$

Then, the DCC metric is defined as

$$\kappa_c = \frac{\sum_{i\in N'} \sum_{k\in K} Z_i^k - n}{n} \times 100\%. \tag{15}$$

Under perfect DCC, the sum $\sum_{i\in N'} \sum_{k\in K} Z_i^k$ would be equal to n because each customer would see exactly one vehicle over the planning horizon, and thus, κ_c would be minimized to zero (κ_c cannot be negative as every customer must see some vehicle at some point in a nontrivial instance, i.e., one with no redundant customers). Thus, κ_c measures the percentage deviation away from this ideal.

7.1.2. DRC Metric. Given the output of an instance, let R^{kt} be a sequence containing the indices of customers in the route taken by vehicle k in time t, listed in the order in which they are visited, with $|R^{kt}|$ denoting the length of the sequence. If k is not dispatched on day t, R^{kt} is an empty sequence (and, thus, has length zero). Also, let R_i^{kt} denote the ith entry in the route. Given two lists R and R' not necessarily of the same length, let $L_{R,R'}(|R|,|R'|)$ be the *Levenshtein edit distance* between R and R', where $L_{R,R'}(\cdot,\cdot)$ is recursively defined as follows:

$$L_{R,R'}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0, \\ \min\begin{cases} L_{R,R'}(i-1,j)+1 \\ L_{R,R'}(i,j-1)+1 \\ L_{R,R'}(i-1,j-1)+\delta_{R_i\neq R'_j} \end{cases} & \text{otherwise,} \end{cases} \tag{16}$$

where $\delta_{R_i\neq R'_j}$ is one if $R_i \neq R'_j$ and zero otherwise. In words, the Levenshtein edit distance is the minimum number of single-entry edit operations (i.e., insertion, deletion, or substitution) needed to convert R into R' or vice versa. For example, the distance between $R = (5,9,4)$ and $R' = (5,7)$ is two because R can be changed into R' by deleting entry 9 and substituting 4 with 7, and this cannot be done in fewer steps. The Levenshtein edit distance is a commonly used tool for measuring distances between strings and has been applied in the routing literature in the past (see, for example, Sörensen 2006; Pillac, Gueret, and Medaglia 2012).

Let $\bar{K}$ be the set of vehicles that are deployed at any point over the planning horizon; that is, $\bar{K} = \{k \in K : \max_t |R^{kt}| > 0\}$. Also, let $\bar{T}^k$ be the set of time periods in which vehicle $k \in \bar{K}$ is deployed; that is, $\bar{T}^k = \{t \in T : |R^{kt}| > 0\}$. The DRC metric can then be defined as follows:

$$\kappa_r = \frac{1}{|\bar{K}|} \sum_{k\in\bar{K}} \sum_{t,t'\in\bar{T}^k : t>t'} L_{R^{kt},R^{kt'}}\left(|R^{kt}|, |R^{kt'}|\right). \tag{17}$$

Stated in words, κ_r computes the average edit distances of pairs of routes on which a vehicle is deployed, in which the average is taken over deployed vehicles. If every deployed vehicle has an invariant route, κ_r would be equal to zero. Note that κ_r does not consider pairs in which at least one route is empty because a nondeployment is not a violation of the consistency requirement, and so the metric should not penalize it.

7.2. Perfect Consistency Formulations

In this section, we present two formulations, each explicitly incorporating one of the two consistency requirements introduced. These serve as points of reference for our comparative analysis of FPP formulations. For each of the formulations, we solve the same set of instances solved earlier and compare the results obtained in terms of both cost and consistency.

7.2.1. DCC-IRP Formulation. We first consider the IRP with perfect DCC, which, in keeping with the nomenclature used in the paper so far, we term the DCC-IRP. In addition to those of the basic IRP, this formulation uses the following variables:

- u_i^k: Binary variable equal to one if customer i is assigned to vehicle k and zero otherwise.

The formulation is as follows. This same model also appears in Coelho, Cordeau, and Laporte (2012).

$$\begin{aligned} \min \quad & (1a) \\ \text{s.t.} \quad & (1b) \text{ to } (1c) \\ & \sum_{k\in K} u_i^k = 1 \quad i \in N', && (18a) \\ & z_i^{kt} \leqslant u_i^k \quad i \in N', k \in K, t \in T, && (18b) \\ & u_i^k \in \{0,1\} \quad i \in N', k \in K. && (18c) \end{aligned}$$

Constraints (18a) assign each customer to exactly one vehicle, and constraints (18b) allow vehicle k to visit customer i only if it is assigned to it.

7.2.2. DRC-IRP Formulation. The second model we consider imposes perfect DRC: the DRC-IRP. It uses the following extra variables:

- Integer variables x_{ij}^k equal to one or two if edge (i,j) is assigned to (i.e., can be traversed by) vehicle k and zero otherwise. x_{ij}^k can take a value of two only if edge (i,j) is incident to the depot and forms a one-customer route.

The formulation is as follows:

$$
\begin{aligned}
\min \quad & (1a) \\
\text{s.t.} \quad & (1b) \text{ to } (1c) \\
& y_{ij}^{kt} \leqslant x_{ij}^k && (i,j) \in E,\ k \in K,\ t \in T, && (19a)\\
& \sum_{k\in K} \sum_{j\in N':(i,j)\in E} x_{ij}^k + x_{ji}^k = 2 && i \in N', && (19b)\\
& x_{ij}^k \in \{0,1\} && (i,j) \in E',\ k \in K, && (19c)\\
& x_{0j}^k \in \{0,1,2\} && (0,j) \in E,\ k \in K. && (19d)
\end{aligned}
$$

The DRC formulation is essentially the rFPP-IRP with the added condition that route-defining variables are explicitly linked to vehicles. Constraints (19a) and (19b) serve similar purposes to those of (2a) and (2b), respectively.

7.3. Reassignment Mixed-Integer Programs (MIPs)

An analysis of the FPPs in relation to perfect consistency formulations requires us to compare results based on consistency metrics. However, prior to doing that, we must ensure that the comparison is fair by preventing artificially high (i.e., bad) consistency scores. Consider the following scenario: suppose an instance of the rFPP-IRP generates fewer routes than there are vehicles available. In principle, this allows a perfect one-to-one assignment of vehicle to route. However, the solver may arbitrarily switch the assignment in different time periods, yielding an artificial nonzero value for κ_r (and κ_c). Similarly, suppose an instance of the fFPP-IRP forms more clusters than the size of the fleet. In this case, perfect DCC is impossible as some vehicle has to be assigned to multiple clusters. However, we can minimize κ_c by maintaining to the fullest extent the invariance of the driver of the densest clusters and assigning the remaining vehicles to all other clusters as needed. If the assignment is arbitrary, however, then we can end up with a value of κ_c that is higher than the minimum possible.

For this reason, we need to preprocess the outputs of our models before extracting statistics from them. In particular, we run our outputs through an MIP that reassigns routes to vehicles in order to achieve the minimum possible consistency scores. In addition to achieving a fair comparison, this step simultaneously serves the secondary purpose of performing the a posteriori assignment step that we allude to in the introduction toward attaining optimal consistency.

In what follows, we describe each of the two MIPs that perform this task. Let y be the computed optimal solution of an instance of any of the main formulations described in Section 3 (if found). In both MIPs, we take y_{ij}^{kt} as input and compute reassigned variables $\bar{y}_{ij}^{kt}$. Note that it is straightforward to infer reassigned versions of all other variables from $\bar{y}$, and so we do not do so here.

We make use of the following parameters in both formulations: define β_{ij}^t to be the aggregation of y_{ij}^{kt} over vehicles k; that is,

$$\beta_{ij}^t = \sum_{k\in K} y_{ij}^{kt} \quad (i,j) \in E,\ t \in T. \tag{20}$$

7.3.1. Optimizing for DCC. The DCC reassignment MIP is as follows:

$$
\begin{aligned}
\text{min.} \quad & \sum_{i\in N'} \sum_{k} Z_i^k, && && (21a)\\
\text{s.t.} \quad & \bar{y}_{ij}^{kt} \leqslant 2Z_i^k && (i,j) \in E : i \in N',\ k \in K,\ t \in T, && (21b)\\
& \bar{y}_{ij}^{kt} \leqslant 2Z_j^k && (i,j) \in E : j \in N',\ k \in K,\ t \in T, && (21c)\\
& \sum_{k\in K} \bar{y}_{ij}^{kt} = \beta_{ij}^t && (i,j) \in E,\ t \in T, && (21d)\\
& \sum_{j\in N':(0,j)\in E} \bar{y}_{0j}^{kt} \leqslant 2 && t \in T,\ k \in K, && (21e)\\
& \bar{y}_{ij}^{kt} \in \{0,1\} && (i,j) \in E',\ k \in K,\ t \in T, && (21f)\\
& \bar{y}_{0j}^{kt} \in \{0,1,2\} && (0,j) \in E,\ k \in K,\ t \in T, && (21g)\\
& Z_i^k \in \{0,1\}, && \forall i \in N',\ k \in K,\ t \in T. && (21h)
\end{aligned}
$$

In these, Z_i^k has the same interpretation as that in Section 7.1.1. It is clear that minimizing objective (21a) is equivalent to minimizing κ_c. Constraints (21b) and (21c) ensure that Z_i^k and Z_j^k register the traversing of customers i and j by k at any point. Constraints (21d) replicate the structure of the original output stripped from the vehicle assignment. Finally, constraints (21e) prevent vehicle k from being dispatched to multiple routes in the same time period t after reassignment.

7.3.2. Optimizing for DRC. The task of minimizing κ_r is a more difficult one because of the involved structure of the metric. However, we can attempt to minimize a proxy of it. In particular, we can try to minimize the number of different routes in which a vehicle is involved over the planning horizon by minimizing the number of different customers that form the beginning and end of all of its routes (i.e., the first customer visited as it leaves the depot and the last customer visited as it arrives back at the depot). Although this does not guarantee route consistency (because intermediate customers might be different), it does ensure that, if a consistent routing strategy is available, it is attained. Also, from a DRC standpoint, it is preferable to assign a fixed vehicle to a cluster, as in the case of the fFPP-IRP, even if intermediate customers differ as this would yield a smaller value of κ_r than if the vehicle were to be dispatched to an entirely different cluster in another time period.

Define variables f_i^k as one if i is the first or last customer in a route by k at any point in the horizon and zero otherwise. In addition, let variables $\bar{f}_i$ be the aggregation of f_i^k over k, indicating whether i is a starting or leaving customer in some route at any point during the horizon. $\bar{f}_i$ takes a value of two if there is a route for which i is both starting and leaving (i.e., if the route is $0 \rightarrow i \rightarrow 0$). Similarly, g_i^{kt} is one if i is visited by k on day t, $\bar{g}^{kt}$ is one if k is dispatched on some route on day t, and $\bar{\bar{g}}^k$ is one if k is dispatched at any point during the horizon.

The DRC reassignment MIP is as follows:

$$\min. \sum_{i\in N'}\sum_k f_i^k - \sum_k \bar{\bar{g}}^k + \sum_{i\in N'} \epsilon \bar{f}_i, \tag{22a}$$

$$\text{s.t.} \sum_k \bar{y}_{ij}^{kt} = \beta_{ij}^t \qquad (i,j)\in E,\ t\in T, \tag{22b}$$

$$\bar{y}_{i0}^{kt} \leqslant 2f_i^k \qquad i\in N',\ t\in T,\ k\in K, \tag{22c}$$

$$\sum_{(i,j)\in E} \bar{y}_{ij}^{kt} = 2g_i^{kt} \qquad i\in N,\ t\in T,\ k\in K, \tag{22d}$$

$$\sum_{i\in N':(0,i)\in E} \bar{y}_{0i}^{kt} = 2\bar{g}^{kt} \quad k\in K,\ t\in T, \tag{22e}$$

$$(1/2)\bar{y}_{ij}^{kt} \leqslant \left\{g_i^{kt}, g_j^{kt}\right\} \leqslant \bar{g}^{kt} \leqslant \bar{\bar{g}}^k \leqslant \sum_t \bar{g}^{kt} \qquad (i,j)\in E',\ t\in T,\ k\in K, \tag{22f}$$

$$\bar{y}_{0i}^{kt} \leqslant \bar{f}_i \leqslant \sum_{i\in N'} f_i^k \qquad i\in N',\ t\in T,\ k\in K, \tag{22g}$$

$$\sum_k 2\bar{\bar{g}}^k \leqslant \sum_{i\in N'} \bar{f}_i, \tag{22h}$$

$$\bar{y}_{ij}^{kt} \in \{0,1\} \qquad (i,j)\in E',\ k\in K,\ t\in T, \tag{22i}$$

$$\bar{y}_{0j}^{kt} \in \{0,1,2\} \qquad (0,j)\in E,\ k\in K,\ t\in T, \tag{22j}$$

$$\bar{f}_i \in \{0,1,2\}, \qquad i\in N',\ k\in K,\ t\in T, \tag{22k}$$

$$f_i^k, g_i^{kt}, \bar{g}^{kt}, \bar{\bar{g}}^k \in \{0,1\}, i\in N', k\in K, t\in T. \tag{22l}$$

In these, ϵ is a small number. Constraints (22c) register whether i is the starting or ending customer on any of vehicle k's routes. Constraints (22d) ensure an entering and a leaving edge if there is a route. Constraints (22e) make sure $\bar{g}^{kt}$ takes its correct value. Constraints (22f) perform self-evident bounding roles on the variables given their definitions. The left-hand side constraints of (22g) help $\bar{f}_i$ detect a starting or leaving node, and the right-hand side ones limit $\bar{f}_i$ to one if it is only starting or only leaving. Constraints (22h) mandate that the number of vehicles deployed cannot be larger than the number of distinct starting–leaving pairs. The first term in the objective attempts to minimize the number of distinct routes in the solution. The second term puts upward pressure on $\bar{\bar{g}}^k$ so that a vehicle is not assigned to two different routes even if such an arrangement is feasible, and (22h) puts downward pressure on $\bar{\bar{g}}^k$ so that the number of dispatched vehicles is no more that is required. The third term puts downward pressure on $\bar{f}_i$ so that it is not assigned arbitrarily large values; (22g) cannot perform this function because f_i^k is not restricted from above anywhere.

7.4. Results

We present an analysis of all formulations based on the consistency metrics and on optimal costs compared with perfect consistency formulations. We investigate κ_c for these formulations in relation to the DCC-IRP, and κ_r is analyzed in relation to the DRC-IRP. The output of each instance is fed to the corresponding reassignment MIP before a consistency metric is computed for it. We note here that, upon running, the reassignment MIPs are found to be extremely efficient, requiring no more than a few milliseconds per instance to solve, and so we do not include a running time analysis for these MIPs here. All results are computed based on instances in which both the formulation under consideration and the perfect consistency formulation find optimal solutions. For each of the rFPP and the fFPP, we choose whichever of the two implementations of each problem (i.e., rFPP1 or rFPP2 and fFPP1 or fFPP2) finds an optimal solution.

Tables 3 and 4 summarize the results obtained for κ_c and κ_r, respectively, for various formulations. As in Section 6, the averages are separated by the number of vehicles, instance type, and number of customers. We observe that pure IRP solutions exhibit relatively large deviations from perfect DCC, reaching as high as 17% when $m = 5$ and 25% when $H = 6$. By contrast, the rFPP-IRP exhibits near-perfect DCC with no average exceeding 2%, and fFPP-IRP averages are also significantly lower than those of the IRP in every category. The difference does seem to decrease with larger n, but these results are based on small sample sizes and are, therefore, inconclusive as they stand. However, we do note that this is at least partially because of instances with $H = 3$ which, as Table 3 shows, are both easier to solve to optimality and easier to achieve consistency for. We do not include rows for either the DCC-IRP or the DRC-IRP in Table 3, but we do mention that both these rows are zero upon testing, demonstrating that the DCC reassignment MIP does indeed yield the expected result of $\kappa_c = 0$ for both these formulations.

Table 3. Average Driver–Customer Consistency Metric for Instances Solved to Optimality by (a) Number of Vehicles, (b) Instance Class, (c) Number of Customers

	Number of vehicles			
	2	3	4	5
IRP	5.07 (117)	9.26 (65)	11.09 (43)	17.29 (32)
rFPP	0.00 (91)	0.36 (56)	0.95 (42)	0.00 (28)
fFPP	1.80 (107)	6.04 (57)	5.08 (42)	3.33 (34)

	Instance class			
	H3	L3	H6	L6
IRP	2.87 (95)	2.89 (94)	25.16 (34)	24.27 (34)
rFPP	0.00 (87)	0.00 (85)	1.74 (23)	0.91 (22)
fFPP	1.79 (94)	1.80 (92)	9.26 (27)	10.37 (27)

	Number of customers									
	5	10	15	20	25	30	35	40	45	50
IRP	20.00 (78)	7.78 (54)	3.79 (44)	1.20 (25)	1.64 (22)	0.56 (12)	0.57 (10)	0.00 (7)	0.00 (5)	— (0)
rFPP	0.77 (78)	0.00 (47)	0.00 (30)	0.00 (19)	0.00 (16)	0.00 (10)	0.00 (8)	0.00 (5)	0.00 (4)	— (0)
fFPP	7.44 (78)	3.40 (50)	2.17 (40)	0.45 (22)	0.24 (17)	0.56 (12)	0.57 (10)	0.00 (7)	0.00 (4)	— (0)

Switching to Table 4, we again see that the rFPP-IRP achieves a high degree of DRC as demonstrated by the low edit distances of deployed vehicles, which also translates into significant improvements over the pure IRP. We see that average pairwise route differences for the rFPP-IRP are typically two single-character edits smaller than those of the IRP. The improvement is most pronounced for $m = 5$, which is on the order of 3.5 edits less than the IRP, and for $H = 6$, for which the decrease is close to 5.5 edits. Results for the fFPP-IRP are very close to those of the IRP, which is expected because of their similar structure. Once again, the DRC row is omitted in 4 but is indeed zero, showing the correctness of the DRC reassignment MIP.

As was mentioned at the beginning of the section, the FPPs offer softer implementations of the two consistency requirements we have discussed thus far. In particular, we can think of the fFPP as a relaxation of the DCC condition (because clusters are fixed but not explicitly linked to fixed vehicles), and the rFPP is a relaxation of the DRC condition (because routes are fixed but not explicitly linked to fixed vehicles). Thus, we expect to see reductions in the optimal costs

Table 4. Average Driver–Route Consistency Metric for Instances Solved to Optimality by (a) Number of Vehicles, (b) Instance Class, (c) Number of Customers

	Number of vehicles			
	2	3	4	5
IRP	3.98 (117)	3.11 (65)	2.84 (43)	3.73 (32)
rFPP	2.75 (91)	1.60 (56)	0.87 (42)	0.36 (28)
fFPP	3.35 (107)	2.42 (57)	2.97 (42)	3.49 (34)

	Instance class			
	H3	L3	H6	L6
IRP	1.18 (95)	1.19 (94)	10.30 (34)	9.87 (34)
rFPP	1.11 (87)	1.04 (85)	4.72 (23)	4.24 (22)
fFPP	1.20 (94)	1.09 (92)	10.25 (27)	9.20 (27)

	Number of customers									
	5	10	15	20	25	30	35	40	45	50
IRP	5.17 (78)	4.48 (54)	3.66 (44)	1.19 (25)	2.49 (22)	1.13 (12)	0.40 (10)	0.21 (7)	0.00 (5)	— (0)
rFPP	1.90 (78)	3.18 (47)	1.44 (30)	1.05 (19)	1.01 (16)	1.00 (10)	0.00 (8)	0.00 (5)	0.00 (4)	— (0)
fFPP	4.91 (78)	3.97 (50)	2.67 (40)	0.73 (22)	0.92 (17)	1.13 (12)	0.40 (10)	0.21 (7)	0.00 (4)	— (0)

Table 5. Average Percentage Reduction in Cost Relative to DCC-IRP for Instances in Which Both Formulations Are Solved to Optimality by (a) Number of Vehicles, (b) Instance Class, (c) Number of Customers

	Number of vehicles			
	2	3	4	5
IRP	1.13 (114)	1.95 (63)	2.05 (42)	3.49 (31)
fFPP	0.70 (107)	1.39 (57)	1.21 (42)	2.35 (33)

	Instance class			
	H3	L3	H6	L6
IRP	0.71 (93)	1.02 (92)	3.57 (33)	5.26 (32)
fFPP	0.57 (94)	0.86 (91)	2.26 (27)	3.32 (27)

	Number of customers									
	5	10	15	20	25	30	35	40	45	50
IRP	3.29 (78)	1.91 (54)	1.49 (43)	0.67 (24)	0.15 (18)	0.04 (12)	0.26 (10)	0.00 (7)	0.00 (4)	— (0)
fFPP	2.15 (78)	1.27 (50)	1.06 (39)	0.31 (22)	0.01 (17)	0.04 (12)	0.26 (10)	0.00 (7)	0.00 (4)	— (0)

of the fFPP-IRP and the rFPP-IRP relative to the DCC-IRP and the DRC-IRP, respectively. It is instructive to quantify these reductions. Tables 5 and 6 present these results augmented with corresponding results for the pure IRP for comparison. With respect to the DCC-IRP, we observe that the savings produced are small in all categories, ranging from 0% to 5%. As expected, the pure IRP realizes larger savings than the fFPP-IRP across the board (as it is itself a relaxation of FPPs) although the difference is not large because of the already modest nature of the reduction for the pure IRP. As for the performance of the rFPP-IRP in relation to the DRC-IRP, the savings realized are again small although, in this case, they differ not insubstantially from the pure IRP, which achieves decent savings—as high as 12% in some categories.

We note here that these cost reduction results mirror to a large extent the results for κ_c and κ_r; that is, cost and consistency trade off largely in a one-to-one fashion. This suggests that, if there are compelling practical and organizational reasons to adopt a fixed-partition policy, then this can be done without fear of compromising on solution quality in relation to the cost–consistency frontier.

8. Conclusions

In this work, we introduce formulations for a practical variant of the classical inventory routing problem that we call the fixed-partition policy IRP. In this problem,

Table 6. Average Percentage Reduction in Cost Relative to DRC-IRP for Instances in Which Both Formulations Are Solved to Optimality by (a) Number of Vehicles, (b) Instance Class, (c) Number of Customers

	Number of vehicles			
	2	3	4	5
IRP	5.86 (81)	6.04 (44)	5.88 (29)	4.58 (17)
rFPP	4.19 (79)	2.18 (43)	1.12 (29)	0.00 (17)

	Instance class			
	H3	L3	H6	L6
IRP	3.49 (64)	5.15 (63)	8.23 (23)	11.97 (21)
rFPP	1.81 (64)	2.92 (61)	3.06 (22)	4.59 (21)

	Number of customers									
	5	10	15	20	25	30	35	40	45	50
IRP	7.26 (77)	9.29 (35)	7.45 (14)	0.00 (10)	0.00 (9)	0.02 (8)	0.00 (8)	0.00 (6)	0.00 (4)	— (0)
rFPP	2.26 (77)	5.92 (35)	5.86 (13)	0.00 (10)	0.00 (8)	0.01 (8)	0.00 (8)	0.00 (5)	0.00 (4)	— (0)

customers are partitioned into clusters that do not change over the planning horizon, after which these clusters are serviced independently. The motivation behind this is threefold: to reduce the operational complexity of the system through independent planning of clusters; to improve the robustness of the system by preventing widespread shortages resulting from unforeseen disruptions in the network and isolating such problems to the affected cluster; and to achieve a degree of consistency in the routing plan, which is a desired feature in customer-oriented planning. In the rigid format of the fixed-partition policy, the clusters themselves define the routes in an all-or-nothing approach: in any time period, either the full route is taken, visiting all customers in the partition along the way, or none are visited. In the flexible version, routing can be done arbitrarily within the cluster, but the cluster still does not change with time. Both formulations are computationally tested and compared with the original IRP, and results show that the added cost of imposing the FPP structure is small. Finally, we conduct a very extensive study of the consistency features of the proposed FPPs and how they trade off with the cost of the solution. We observe that consistency comes at the expense of cost in a steady gradation with no standout options on the proverbial Pareto front. Thus, choices can be made purely based on practical considerations. In the future, more extensive testing is needed to parse out the performance of FPPs on larger test cases. In addition, a path can be envisaged to specialized algorithms that take advantage of the special structure introduced by the open-link x variables to achieve better running times.

References

Anily S, Bramel J (2004a) An asymptotic 98.5%-effective lower bound on fixed partition policies for the inventory-routing problem. *Discrete Appl. Math.* 145(1):22–39.

Anily S, Bramel J (2004b) A probabilistic analysis of a fixed partition policy for an inventory-routing models. *Naval Res. Logist.* 51(7): 925–948.

Anily S, Federgruen A (1990) One warehouse multiple retailer systems with vehicle routing costs. *Management Sci.* 36(1):92–114.

Archetti C, Speranza MG (2016) The inventory routing problem: The value of integration. *Internat. Trans. Oper. Res.* 23(3): 393–407.

Archetti C, Bertazzi L, Laporte G, Speranza MG (2007) A branch-and-cut algorithm for a vendor managed inventory routing problem. *Transportation Sci.* 41(3):382–391.

Archetti C, Bianchessi N, Irnich S, Speranza MG (2014) Formulations for an inventory routing problem. *Internat. Trans. Oper. Res.* 21(3):353–374.

Bertazzi L, Speranza M (2012) Inventory routing problems: An introduction. *EURO J. Transportation Logist*. 1(4):307–326.

Bertazzi L, Speranza M (2013) Inventory routing problems with multiple customers. *EURO J. Transportation Logist*. 2(3):255–275.

Bitran GR, Yanasse HH (1982) Computational complexity of the capacitated lot size problem. *Management Sci.* 28(10):1174–1186.

Chan L, Federgruen A, Simchi-Levi D (1998) Probabilistic analyses and practical algorithms for inventory-routing models. *Oper. Res.* 46(1):96–106.

Coelho L, Cordeau J-F, Laporte G (2014) Thirty years of inventory routing. *Transportation Sci.* 48(1):1–19.

Coelho LC, Cordeau J-F, Laporte G (2012) Consistency in multi-vehicle inventory-routing. *Transportation Res.: Part C Emerging Tech.* 24:270–287.

Feillet D, Garaix T, Lehuédé F, Péton O, Quadri D (2014) A new consistent vehicle routing problem for the transportation of people with disabilities. *Networks* 63(3):211–224.

Fleischer S, Berg A, Zimmermann M, Wüste K, Behrens J (2009) Nurse-patient interaction and communication: A systematic literature review. *J. Public Health* 17(5):339–353.

Florian M, Lenstra JK, Rinnooy Kan AHG (1980) Deterministic production planning: Algorithms and complexity. *Management Sci.* 26(7):669–679.

Gaur V, Fisher M (2004) A periodic inventory routing problem at a supermarket chain. *Oper. Res.* 52(6):813–822.

Groër C, Golden B, Wasil E (2009) The consistent vehicle routing problem. *Manufacturing Service Oper. Management* 11(4):630–643.

Kovacs AA, Parragh SN, Hartl RF (2014a) A template-based adaptive large neighborhood search for the consistent vehicle routing problem. *Networks* 63(1):60–81.

Kovacs AA, Golden BL, Hartl RF, Parragh SN (2014b) The generalized consistent vehicle routing problem. *Transportation Sci.* 49(4):796–816.

Li J, Chu F, Chen H (2011) A solution approach to the inventory routing problem in a three-level distribution system. *Eur. J. Oper. Res.* 210(3):736–744.

Macdonald T, Dörner K, Gandibleux X (2009) Metaheuristics for the consistent nurse scheduling and routing problem. MS thesis, Université de Nantes, Nantes, France.

Massey D, Chaboyer W, Aitken L (2014) Nurses' perceptions of accessing a medical emergency team: A qualitative study. *Australian Critical Care* 27(3):133–138.

Minick P, Harvey S (2003) The early recognition of patient problems among medical-surgical nurses. *Medsurgical Nursing* 12(5): 291–297.

Mok E, Chiu PC (2004) Nurse–patient relationships in palliative care. *J. Adv. Nursing* 48(5):475–483.

Padberg M, Rinaldi G (1991) A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. *SIAM Rev.* 33(1):60–100.

Pillac V, Gueret C, Medaglia A (2012) Route consistency vehicle routing: A bi-objective approach. *ROADEF 2012, Angers, France*.

Raa B, Aghezzaf E-H (2009) A practical solution approach for the cyclic inventory routing problem. *Eur. J. Oper. Res.* 192(2):429–441.

Raa B, Dullaert W (2017) Route and fleet design for cyclic inventory routing. *Eur. J. Oper. Res.* 256(2):404–411.

Ralphs T, Kopman L, Pulleyblank W, Trotter L (2003) On the capacitated vehicle routing problem. *Math. Programming* 94(2): 343–359.

Rodríguez-Martín I, Salazar-González J-J, Yaman H (2019) The periodic vehicle routing problem with driver consistency. *Eur. J. Oper. Res.* 273(2):575–584.

Sörensen K (2006) Route stability in vehicle routing decisions: A bi-objective approach using metaheuristics. *Central Eur. J. Oper. Res.* 14(2):193–207.

Zhao Q-H, Wang S-Y, Lai K (2007) A partition approach to the inventory/routing problem. *Eur. J. Oper. Res.* 177(2):786–802.