Wang, Kai, Lu Zhen, Jun Xia, Roberto Baldacci, and Shuaian Wang. "Routing Optimization with Generalized Consistency Requirements." *Transportation Science* 56, no. 1 (January 2022): 223–44.
https://doi.org/10.1287/trsc.2021.1072

Submitted to *Transportation Science*
manuscript TS-2020-0209

Routing Optimization with Generalized Consistency Requirements

Kai Wang
Sloan School of Management, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA; Email: cwangkai@mit.edu

Lu Zhen
School of Management, Shanghai University, Shanghai, China; Email: lzhen@shu.edu.cn

Jun Xia
Sino-US Global Logistics Institute, Antai College of Economics & Management, Shanghai Jiao Tong University, Shanghai, China; Email: lgtxiaj@sjtu.edu.cn

Roberto Baldacci
Department of Electrical, Electronic, and Information Engineering "Guglielmo Marconi", University of Bologna, Bologna, Italy; Email: r.baldacci@unibo.it

Shuaian Wang
Department of Logistics and Maritime Studies, The Hong Kong Polytechnic University, Hong Kong; Email: hans.wang@polyu.edu.hk

The consistent vehicle routing problem (ConVRP) aims to design synchronized routes on multiple days to serve a group of customers while minimizing the total travel cost. It stipulates that customers should be visited at roughly the same time (time consistency) by several familiar drivers (driver consistency). This paper generalizes the ConVRP for any level of driver consistency and additionally addresses route consistency, which means that each driver can traverse at most a certain proportion of different arcs of routes on planning days, which guarantees route familiarity. To solve this problem, we develop two set-partitioning-based formulations, one based on routes and the other based on schedules. We investigate valid lower bounds on the linear relaxations of both of the formulations that are used to derive a subset of columns (routes and schedules); within the subset are columns of an optimal solution for each formulation. We then solve the reduced problem of either one of the formulations to achieve an optimal solution.

Numerical results show that our exact method can effectively solve most of the medium-sized ConVRP instances in the literature, and can also solve some newly generated instances involving up to 50 customers. Our exact solutions explore some managerial findings with respect to the adoption of consistency measures in practice. First, maintaining reasonably high levels of consistency requirements does not necessarily always lead to a substantial increase in cost. Second, a high level of time consistency can potentially be guaranteed by adopting a high level of driver consistency. Third, maintaining high levels of time consistency and driver consistency may lead to lower levels of route consistency.

Key words: vehicle routing, generalized consistency requirements, exact method, service efficiency.
History: April 6, 2021

1. Introduction

Small-package shipping companies collect and distribute packages among diverse customers throughout the days of a given period (e.g., one week). The number of deliveries and pickups (or demands) associated with each customer varies from day to day. To fulfill these multi-day demands in a cost-efficient manner, service providers optimize their working routes for a given period such that total travel cost (or time) can be minimized. In practice, service providers offer synchronized services, meaning that the routes operated on different days are coordinated for customer convenience, because customers usually prefer to be served at roughly the same time (*time consistency*, TC) by the same driver or a small set of drivers (*driver consistency*, DC) on different days. Embedding this synchronization feature within the vehicle routing decision yields a new class of problems called consistent vehicle routing problems (ConVRPs) (e.g., see Groër et al. 2009, Kovacs et al. 2015a); a survey on ConVRPs can be found in Kovacs et al. (2014a).

Synchronization should not only aim to increase customer satisfaction but also to improve service reliability by designing consistent working routes for drivers. Unfamiliar routing plans and service tasks lead to costly operations of routes for drivers (Kovacs et al. 2014a). In world-class small-package shipping companies, such as United Parcel Service (UPS), drivers usually work on fixed routes for many years (Smilowitz et al. 2013), making them familiar with the road and traffic conditions and effectively reducing service delays. To formalize this route familiarity concept, we define a new consistency requirement, called *route consistency* (RC), that aims to design routes consisting of more common arcs for drivers. Specifically, we evaluate the RC level of a driver as the proportion (n_1/n_2) of the number of different arcs (n_1) traversed over the total number of all arcs (n_2) traversed over all days. Clearly, the value of n_1/n_2 measures the difference between routes operated by a driver in the context of the number of common arcs traversed over all days. With this RC definition, for a planning horizon of D days we have $n_1/n_2 \in [1/D, 1]$, where $n_1/n_2 = 1/D$ corresponds to the highest level of RC, where the driver operates the same route on all the D days, and $n_1/n_2 = 1$ suggests the lowest level of RC, where the routes operated on different days share no common arcs.

Nowadays, under the product return policy offered by online retailers, reverse flows of packages that need to be collected from customers are growing rapidly. In response, shipping companies adopt simultaneous distribution and collection operations, in order to increase vehicle utilizations. In practice, UPS has adopted the solution of simultaneous distribution and collection operations. Such operational flexibility has also been implemented in the ORION system, which is a newly developed smart system that optimizes routing decisions for delivery operations and in which consistency issues are also considered (UPS 2019, Holland et al. 2017).

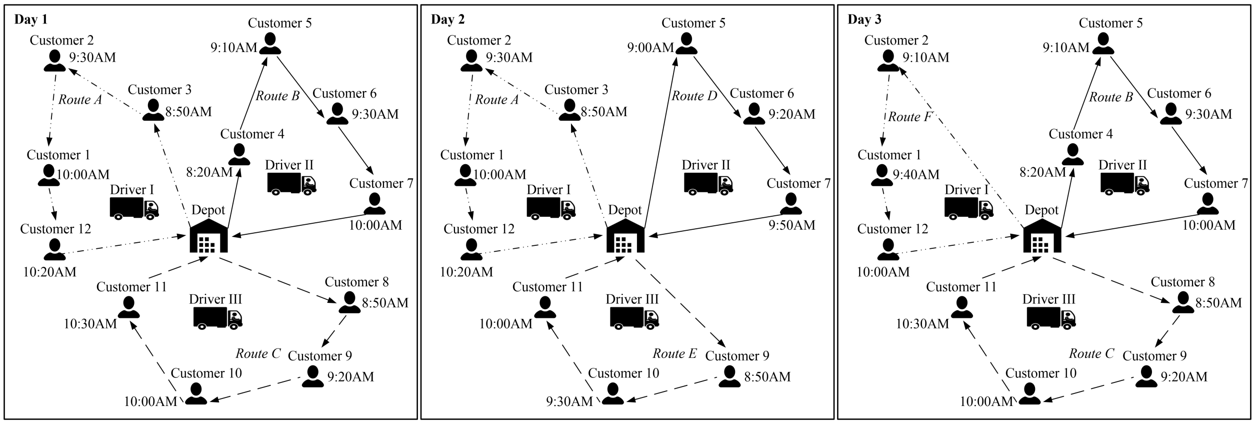

Figure 1 An example of consistent schedules

This paper studies a generalized ConVRP by jointly considering TC, DC, and RC, together with the simultaneous distribution and collection operations, defined as the vehicle routing problem with generalized consistency requirements and simultaneous distribution and collection (VRPGCR-SDC). VRPGCR-SDC aims to design routes on multiple days and combine them into "consistent schedules" for drivers such that demands are satisfied and total travel time is minimized. In our problem, a schedule is a driver's work sheet over planning days, which includes all route information on each day (i.e., visiting sequence and arrival timetable for customers). Operating schedules are consistent if the maximum arrival time difference at each customer on different days is restricted by $\mathbf{L}$ time units (TC level), each customer is served by at most $\mathbf{E}$ different drivers (DC level), and for each driver, the proportion of the number of different arcs traversed over the total number of all arcs traversed over all days is no greater than $\mathbf{F}$.

Figure 1 gives an example of consistent schedules in a three-day planning period. It maintains TC level $\mathbf{L} = 30$ minutes (customers 9, 10, and 11 have the maximum arrival time difference of 30 minutes), DC level $\mathbf{E} = 1$ (each customer is served by one driver), and RC level $\mathbf{F} = 3/7$ (each driver operates routes with 6 different arcs and 14 total arcs on the three days).

The VRPGCR-SDC is computationally challenging because it can be specialized into known NP-hard ConVRPs; for example, a special case of the VRPGCR-SDC, with $\mathbf{E} = 1$ and $\mathbf{F} = 1$ (for which RC is relaxed), is the problem proposed by Groër et al. (2009). This problem is further transformed into the version of Kovacs et al. (2015a) by slightly increasing $\mathbf{E}$ to any integer greater than one and by addressing TC as an objective penalization rather than a hard constraint. Table 1 summarizes the recent works on ConVRPs, as well as those on the traveling salesman problem (TSP) and periodic vehicle routing problem (PVRP), which only consider TC or DC.

As shown in Table 1, ConVRPs have been studied with many heuristic algorithms. Some of them treat the consistency constraints as soft constraints. For example, partial consistency requirements

Table 1 Summary of the contributed works on routing problems with consistency requirements

Problem	Reference	Consistency	Objective	Constraints	Algorithm
ConVRP	Groër et al. (2009); Tarantilis et al. (2012); Kovacs et al. (2014b)	TC, DC ($\mathbf{E}=1$)	Minimize total travel cost	Hard constraints on TC and DC in model, but no guarantee on TC by algorithm	Template route-based algorithms
	Goeke et al. (2019)		Minimize total travel cost	Hard constraints on TC and DC	Exact algorithm based on a set-partitioning formulation
	Kovacs et al. (2015a)	TC, DC	Minimize total travel cost and TC in a single objective	Soft constraints on TC, hard constraints on DC	Large neighborhood search
	Kovacs et al. (2015b); Lian et al. (2016)		Minimize total travel cost, TC, and DC in multi-objectives	Soft constraints on TC and DC	Multi-directional large neighborhood search or local search
TSP	Subramanyam and Gounaris (2016); Subramanyam and Gounaris (2018)	TC	Minimize total travel cost	Hard constraints on TC	Exact algorithms based on the branch-and-bound framework
PVRP	Smilowitz et al. (2013)	DC	Minimize total travel cost and optimize DC in separated phases	Soft constraints on DC	Tabu search
TWAVRP	Spliet and Gabor (2015); Subramanyam et al. (2018)	DC	Minimize total travel cost	Hard constraints on TC (based on time window assignment)	Exact algorithms based on the branch-and-bound framework

may be moved to the objectives (e.g., see Kovacs et al. 2015a,b, Lian et al. 2016), whereas some requirements are regulated using template routes in the algorithms (e.g., see Groër et al. 2009, Tarantilis et al. 2012, Kovacs et al. 2014b). The template routes are based on a simple precedence principle, which requires that the order of two customers visited by the same driver must be the same on all days when both customers require service.

Despite its practical benefits on improving service efficiency, RC has not been studied well in existing ConVRPs. To improve drivers' familiarity with their provided services, Smilowitz et al. (2013) take into account drivers' familiarity with working regions. Moreover, in the proposed heuristic algorithms for solving ConVRP (e.g., see Groër et al. 2009, Kovacs et al. 2015a), the idea of template routes has been used to maintain a high level of DC, which indeed improves driver's familiarity on customers at the same time. However, these efforts do not really capture driver's preferences on operating routes on days with similar routes to increase familiarity with road and traffic conditions—referring to the concept of RC that has not been addressed in any ConVRPs.

1.1. Literature Review

Groër et al. (2009) first defined the ConVRP by addressing both TC and DC. The proposed DC states that each customer must be served by the same driver over the planning days (i.e., $\mathbf{E}=1$). They developed a record-to-record algorithm using a given set of template routes to explore highly synchronized routes to meet TC and DC. But the algorithm had no guarantee of TC in the output solution. The authors provided a set of benchmark instances with up to 199 customers to analyze the efficiency of their algorithm. Based on template routes, a tabu search method (Tarantilis et al. 2012) and an adaptive large neighborhood search method (Kovacs et al. 2014b) were developed to iteratively improve the routing decisions. The concept of template routes benefits the TC. Because working routes on different days are generated from a template route for a driver (following DC level $\mathbf{E}=1$), the working routes will have small arrival-time differences for each customer served by the same driver. Kovacs et al. (2014b) allowed for an adjustment of departure times at the depot to synchronize the arrival times at each customer over the planning days. In a subsequent work, Kovacs et al. (2015a) further generalized the ConVRP by considering a relaxed DC: each customer can be serviced by a few drivers instead of just one considering that drivers could be absent during the planning horizon (e.g., on leave for a vacation). They penalized the arrival-time difference in the objective function. As an alternative to using template routes, they proposed exploring new routes with a large neighborhood search (LNS) method, which is embedded with a greedy time-adjustment procedure to improve TC. Multi-objective models and LNS-based methods for the ConVRPs have been investigated by Kovacs et al. (2015b) and Lian et al. (2016).

In contrast to heuristic methods, exact methods have seldom been considered for ConVRPs. To the best of our knowledge, Goeke et al. (2019) is the only work that has solved the ConVRP with $\mathbf{E}=1$ in an exact fashion. Considering the DC level $\mathbf{E}=1$, they posited that each driver serves a disjoint subset (i.e., a cluster) of customers over the planning period, based on which they proposed a set-partitioning formulation by enumerating all possible customer clusters (a special structure of $\mathbf{E}=1$). A column-and-cut generation procedure was developed to find a valid lower bound on the proposed formulation. Given an upper bound computed from an LNS-based algorithm, they found a subset of clusters that includes optimal ones for the problem. An optimal solution is returned by solving the reduced problem using the obtained customer clusters. Their algorithm solves the problem, with up to 5 days and 30 customers, to optimality. In addition to the practical importance of the RC level, our work is further motivated by the following reasons: (i) The solution framework of Goeke et al. (2019) is based on the special structure of $\mathbf{E}=1$ that is hardly extendable to generalized cases; (ii) a DC level $\mathbf{E}=1$ is restrictive in practice (Kovacs et al. 2015a) and the general case of $\mathbf{E}\geq 1$ studied by other ConVRP works all focused on heuristics; (iii) even if we slightly extend the DC level to $\mathbf{E}=2$, there could be significant operational benefits.

In addition to the exact algorithm for a special case of the ConVRP, Subramanyam and Gounaris (2016) and Subramanyam and Gounaris (2018) have developed exact algorithms based on the branch-and-bound framework to solve a TSP that considers TC for a multi-period setting (known as ConTSP), where one route per period (i.e., per day) is designed and, thus, DC is not involved.

Spliet and Gabor (2015) studied the time window assignment vehicle routing problem (TWAVRP) that determines a fixed length time window for each customer under different demand realization scenarios. This relates to time consistency in the ConVRP in the sense that the service time of each customer under any scenario must be within the proposed time window, thus ensuring a small arrival time difference. Different from time consistency in the ConVRP, the TWAVRP has an exogenous time window for each customer, and then narrows it down to a small one of fixed length. Meanwhile, waiting at a customer is allowed in the TWAVRP. In their model, the vehicle route and service time are modeled within a single column, being determined and generated simultaneously when solving a pricing problem. A branch-price-and-cut algorithm is developed to solve to optimality problem instances with up to 25 customers and 3 scenarios.

Subramanyam et al. (2018) proposed a branch-and-bound framework to solve the TWAVRP that is similar to the exact method proposed in Subramanyam and Gounaris (2018) for solving the ConTSP. In contrast to Spliet and Gabor (2015), the proposed solution method is able to tackle both continuous and discrete sets of time window assignments, and can solve more general scenario-based models capturing uncertainties. Their experimental results showed that the exact method solves to optimality the problem instances with up to 25 customers and 15 scenarios.

To improve drivers' efficiency in logistics operations, Smilowitz et al. (2013) enhanced workforce management by focusing on drivers' familiarity with both customers and regions. The degree of familiarity is evaluated using the number of repeated visits by each driver to each customer and region; they maximized the sum of these numbers in their model. Notably, the region familiarity considered in their work motivates drivers to work among roughly fixed regions each day. This is similar to our goal of using RC to improve drivers' familiarity with the road conditions. In our problem, the familiarity of roads for a driver depends on the traversed common arcs among the routes. This can be seen as a supplementary way of describing drivers' familiarity with the route, in contrast to the way used in Smilowitz et al. (2013) that minimized the number of different regions traversed by the drivers.

Allowing simultaneous distribution and collection in VRPs improves vehicle utilization and reduces operating costs. This consideration complicates the solution, mainly because the selection of customers to visit together with the visiting sequence will jointly determine the minimum capacity required by a route. Existing VRPs with simultaneous distribution and collection operations have mainly been studied heuristically (e.g., see Subramanian et al. 2010, Liu et al. 2013, Avci and Topaloglu 2016). However, Dell'Amico et al. (2006) designed an exact method for addressing it with a branch-and-price algorithm that optimally solves the problem with up to 40 customers.

1.2. Contributions of This Paper

In this paper, we study the VRPGCR-SDC, the most generalized version of the ConVRP to date, to investigate the influences of TC, DC, and RC on the multi-day routing optimization. Our work makes the following major contributions:

(i) The VRPGCR-SDC extends the scope of existing ConVRP studies, by generalizing the consistency requirements considered. From a managerial point of view, by introducing RC, the problem for the first time extends the utility of consistency to improve service reliability in addition to customer satisfaction. By exploring this problem, it is convenient to study the impacts of the consistency requirements and to look into the trade-offs behind customer satisfaction and service efficiency/reliability with respect to multi-day routing decisions.

(ii) Exploration of VRPGCR-SDC calls for an exact solution strategy for dealing with the generalized consistency requirements, while existing ConVRP studies have been shown to be very challenging. In this paper, we develop an exact algorithm that is indeed the first exact method for the ConVRP that considers any level of DC and also the requirement of RC. Extensive numerical experiments are conducted to attest the efficiency and effectiveness of the proposed exact method. Results show that our exact method performs effectively in finding optimal solutions for various ConVRPs, including the specialized problems studied in the literature and the generalized problem studied in this paper.

(iii) By leveraging the exact solutions of VRPGCR-SDC, we obtain the following managerial findings on imposing the consistency requirements. First, maintaining reasonable levels of consistency requirements, rather than very high levels of consistency requirements, does not necessarily lead to a substantial increase in cost. Second, a high level of TC can potentially be guaranteed by adopting a high level of DC. Third, maintaining high levels of TC and DC does not actually motivate the assignment of similar routes with more common arcs to drivers, thus showing the necessity of introducing RC in the multi-day routing optimization.

Our exact algorithm is based on a a route-based formulation F1 and a schedule-based formulation F2, both of which are developed in a set-partitioning fashion. We investigate valid lower bounds on both formulations by using a column-and-cut generation (CCG) algorithm. Given the obtained lower bounds, we generate a set of routes and a set of schedules that contain the routes and schedules in the optimal solutions for F1 and F2, respectively, and directly solve the reduced problem of F1 (using only the generated routes) or F2 (using only the generated schedules) by using a generic mixed integer programming (MIP) solver based on the sizes of the optimal sets computed. The generation of routes and schedules for F1 and F2 uses the enumeration-based solution framework proposed in Baldacci et al. (2008) for the capacitated VRP whereas F1 and F2 are combined in the exact method following the approach proposed by Mingozzi et al. (2013) for the multitrip VRP. We summarize other specific methodological contributions as follows:

- We develop new constraints to address TC in our formulations (similar to Spliet and Desaulniers (2015)), using Big-M-based constraints in contrast to existing formulations. The idea behind the new TC constraint reformulations can be adapted to many existing routing problems where the time of each customer visit has to be precisely captured in set-partitioning-based formulations.

- To address all generalized consistency requirements, the proposed set-partitioning-based formulations feature binary and continuous variables, as well as exponentially many *route-dependent constraints*. To our knowledge, our work is the first attempt at applying this route enumeration-based solution framework to solve such complicated set-partitioning-based models, especially on developing new relaxations for valid lower bounding and on implementing an efficient CCG algorithm for solving the relaxations.

The remainder of this paper is organized as follows. In §2, we formally define the VRPGCR-SDC and specify the two formulations F1 and F2. In §3, we describe the exact method by introducing optimality conditions and two relaxations R1 and R2 based on F1 and F2, respectively. §4 and §5 describe the algorithms used to solve R1 and R2, respectively. Numerical experiments are conducted in §6. Sensitivity analysis and managerial findings are explored in §7. Conclusions are drawn in §8. Supplementary material is provided in the e-companion (EC) to this paper, and all proofs of statements in this paper are presented in the EC.

2. Problem Description and Mathematical Formulations

In this section, we first define the VRPGCR-SDC in §2.1. Then, we give two formulations for the VRPGCR-SDC, these being a route-based formulation (F1) and a schedule-based formulation (F2) in §2.2 and §2.3, respectively.

2.1. Problem Description

In the VRPGCR-SDC, a set of customers $\mathcal{N} = \{1, ..., N\}$ is served by a set of homogeneous drivers $\mathcal{K} = \{1, ..., K\}$ using identical vehicles with capacity Q that are based at a single depot (denoted by 0). Drivers and vehicles share a one-to-one relationship, and hereafter we use them interchangeably.

Given the set of days $\mathcal{D} = \{1, ..., D\}$, a customer $i \in \mathcal{N}$ has p_{id} delivery demands and q_{id} pickup demands on day $d \in \mathcal{D}$. Let g_{id} be a binary parameter equal to 1 if customer $i \in \mathcal{N}$ on day $d \in \mathcal{D}$ needs a service (i.e., $p_{id} > 0$ or $q_{id} > 0$) and 0 otherwise. If $p_{id} = 0$ and $q_{id} = 0$, then customer i does not require service on day d. We define $\mathcal{N}_d$ as the subset of customers needing a service on day d, and $\mathcal{N}_d^+ = \mathcal{N}_d \cup \{0\}$. The problem can be defined on a graph $\mathcal{G} = (\mathcal{N}^+, \mathcal{A})$ with node set $\mathcal{N}^+ = \cup_{d\in\mathcal{D}} \mathcal{N}_d^+$ and arc set $\mathcal{A} = \cup_{d\in\mathcal{D}} \mathcal{A}_d$, where the set $\mathcal{A}_d = \{(0,j)|j \in \mathcal{N}_d\} \cup \{(i,0)|i \in \mathcal{N}_d\} \cup \{(i,j)|i,j \in \mathcal{N}_d : i \neq j\}$, $d \in \mathcal{D}$, represents the set of arcs associated with day d. We denote by $t_{ij} \geq 0$ the vehicle's travel time on arc $(i,j) \in \mathcal{A}$ and by $s_{id} \geq 0$ the service time for customer $i \in \mathcal{N}_d^+$ on day $d \in \mathcal{D}$ ($s_{0d} = 0$). Each vehicle has a driving time limit T.

The objective of the VRPGCR-SDC is to design a set of multi-day routes while minimizing the total travel time over the planning horizon, subject to the following constraints: (C.1) each customer is visited exactly once on each day when service is required; (C.2) each driver operates at most one route per day, where each route is subject to both capacity and maximum duration limits; (C.3) the arrival time difference at a customer is less than or equal to **L**; (C.4) each customer is served by at most **E** different drivers; and (C.5) each driver operates routes on D days with at most **F** proportion of different arcs (over the total number of all arcs traversed).

In the VRPGCR-SDC, we allow possible adjustments to the departure time of a vehicle from the depot, in order to achieve a high level of TC. Note that a further relaxation on adjusting the departure times from visited customers may bring extra flexibility to satisfy TC. However, as shown by Goeke et al. (2019), such benefits can be very marginal. In practice, vehicles waiting at customers need to occupy public or third-party parking areas outside the depot, and this may incur new costs to offset the potential revenues. Therefore, like most existing ConVRP studies (e.g., see Groër et al. 2009, Tarantilis et al. 2012, Kovacs et al. 2014b, 2015a, Goeke et al. 2019), we assume that vehicles are not allowed to wait at the customers.

For the VRPGCR-SDC, apart from models (F1) and (F2), we have also developed an arc-based model (F0) with a number of variables and constraints that are polynomial in the instance size. The details of F0 are given in the e-companion to this paper (see §EC.1). Due to the weak relaxation of F0, only a small-sized formulation can be solved by using a general-purpose mixed-integer programming (MIP) solver. In this paper, F0 is mainly used to provide benchmark results for some small-sized instances, which are later compared with the proposed exact method in §6.

2.2. Route-based Formulation F1

Let $\mathcal{R}_d$ be the index set of all routes on day d satisfying both capacity and maximum duration constraints, and define $\mathcal{R} = \cup_{d\in\mathcal{D}} \mathcal{R}_d$. Each route $r \in \mathcal{R}_d$ on day d has a duration denoted by t_{rd}^R.

Define a binary parameter a_{ri} that equals 1 if route r visits customer i and that equals 0 otherwise. Let e_{rij} be a binary parameter that equals 1 if route r passes the arc (i,j) and equals 0 otherwise. By defining the sets of decision variables: (i) u_i^k, a binary variable that equals 1 if customer $i \in \mathcal{N}$ is served by driver $k \in \mathcal{K}$ at least once during the planning horizon and 0 otherwise; and, (ii) v_{id}, a nonnegative continuous variable showing the arrival time at customer $i \in \mathcal{N}^+$ on day $d \in \mathcal{D}$; and, (iii) ξ_{rd}^k, a binary variable that equals 1 if route $r \in \mathcal{R}_d$ is operated by driver $k \in \mathcal{K}$ on day $d \in \mathcal{D}$ and 0 otherwise, and (iv) w_{ij}^k, a binary variable that equals 1 if arc $(i,j) \in \mathcal{A}$ is traversed by driver $k \in \mathcal{K}$ at least once on D days and 0 otherwise, the route-based formulation F1 is as follows:

$$z(\text{F1}) = \min \sum_{d\in\mathcal{D}} \sum_{k\in\mathcal{K}} \sum_{r\in\mathcal{R}_d} t_{rd}^R \xi_{rd}^k \tag{1a}$$

$$\text{s.t.} \sum_{k\in\mathcal{K}} \sum_{r\in\mathcal{R}_d} a_{ri}\xi_{rd}^k = g_{id} \quad i \in \mathcal{N}, d \in \mathcal{D}, \tag{1b}$$

$$\sum_{r\in\mathcal{R}_d} \xi_{rd}^k \leq 1 \quad k \in \mathcal{K}, d \in \mathcal{D}, \tag{1c}$$

$$\sum_{r\in\mathcal{R}_d} a_{ri}\xi_{rd}^k \leq u_i^k \quad k \in \mathcal{K}, i \in \mathcal{N}_d, d \in \mathcal{D}, \tag{1d}$$

$$\sum_{k\in\mathcal{K}} u_i^k \leq \mathbf{E} \quad i \in \mathcal{N}, \tag{1e}$$

$$\sum_{r\in\mathcal{R}_d} e_{rij}\xi_{rd}^k \leq w_{ij}^k \quad k \in \mathcal{K}, (i,j) \in \mathcal{A}, d \in \mathcal{D}, \tag{1f}$$

$$\sum_{(i,j)\in\mathcal{A}} w_{ij}^k \leq \mathbf{F} \sum_{d\in\mathcal{D}} \sum_{(i,j)\in\mathcal{A}} \sum_{r\in\mathcal{R}_d} e_{rij}\xi_{rd}^k \quad k \in \mathcal{K}, \tag{1g}$$

$$v_{jd} \geq v_{id} + \sum_{k\in\mathcal{K}} \sum_{r\in\mathcal{R}_d} e_{rij}\xi_{rd}^k (s_{id} + t_{ij}) - (1 - \sum_{k\in\mathcal{K}} \sum_{r\in\mathcal{R}_d} e_{rij}\xi_{rd}^k)T \quad d \in \mathcal{D}, (i,j) \in \mathcal{A}, j \neq 0, \tag{1h}$$

$$v_{jd} \leq v_{id} + \sum_{k\in\mathcal{K}} \sum_{r\in\mathcal{R}_d} e_{rij}\xi_{rd}^k (s_{id} + t_{ij}) + (1 - \sum_{k\in\mathcal{K}} \sum_{r\in\mathcal{R}_d} e_{rij}\xi_{rd}^k)T \quad d \in \mathcal{D}, (i,j) \in \mathcal{A}, j \neq 0, \tag{1i}$$

$$v_{id} + s_{id} + t_{i0} \leq T \quad i \in \mathcal{N}_d, d \in \mathcal{D}, \tag{1j}$$

$$(v_{id} - v_{id'})g_{id}g_{id'} \leq \mathbf{L} \quad i \in \mathcal{N}, d, d' \in \mathcal{D} : d \neq d', \tag{1k}$$

$$\xi_{rd}^k \in \{0,1\} \quad r \in \mathcal{R}_d, k \in \mathcal{K}, d \in \mathcal{D}, \tag{1l}$$

$$w_{ij}^k \in \{0,1\} \quad k \in \mathcal{K}, (i,j) \in \mathcal{A}, \tag{1m}$$

$$u_i^k \in \{0,1\} \quad k \in \mathcal{K}, i \in \mathcal{N}, \tag{1n}$$

$$v_{id} \geq 0 \quad i \in \mathcal{N}_d, d \in \mathcal{D}. \tag{1o}$$

Objective (1a) aims at minimizing the total travel time. Constraints (1b) enforce that each customer is visited exactly once on each day on which service is required. Constraints (1c) impose that each driver can be assigned to at most one route on each day. Constraints (1d)–(1e) link the variables $\boldsymbol{\xi}$ and $\boldsymbol{u}$ to define the DC: the maximum number of different drivers to serve a customer is limited by $\mathbf{E}$. Constraints (1f)–(1g) link the variables $\boldsymbol{\xi}$ and $\boldsymbol{w}$ to define the RC: for each driver, at most $\mathbf{F}$ proportion of different arcs of routes can be traversed over the planning horizon. Constraints (1h) and (1i) determine the arrival times at the customers. Constraints (1j) impose that each driver must return to the depot before time T. Constraints (1k), together with (1h)–(1j), define the TC by ensuring that the maximum arrival time difference at a customer is limited by $\mathbf{L}$. Constraints (1l)–(1o) define the domain of the decision variables.

Note that constraints (1h) and (1i) used to define TC are with big-M coefficients in F1. For the continuous relaxation of F1, these constraints will probably be inessential to the problem optimality, making the TC constraints substantially relaxed and leading to a weak relaxation bound (see §6.2). To enhance the influence of TC on the relaxation tightness of F1, we propose a new reformulation to enhance the TC-related constraints. The reformulation requires the following additional parameters and variables. Let b_{rid} be the accumulated time of a vehicle traveling from the depot to customer i by route r on day d, which also includes the service times of the visited customers (we set $b_{rid}=0$ for each $d \in \mathcal{D}$ if customer i is not visited by route r). Let l_{rd} be the latest departure time of route r on day d from the depot to respect the duration time limit T, where $l_{rd}=T-t^R_{rd}$. Consider a new variable ϕ^k_{rd}, which is a nonnegative continuous variable that records the departure time of route $r \in \mathcal{R}_d$ operated by driver $k \in \mathcal{K}$ on day $d \in \mathcal{D}$. Based on the above notations, constraints (1h) and (1i) can be replaced by the following reformulated constraints (2a)–(2c) to address TC:

$$v_{id}=\sum_{k\in\mathcal{K}}\sum_{r\in\mathcal{R}_d} a_{ri}\phi^k_{rd}+\sum_{k\in\mathcal{K}}\sum_{r\in\mathcal{R}_d} b_{rid}\xi^k_{rd} \quad i\in\mathcal{N}_d, d\in\mathcal{D}, \tag{2a}$$

$$\phi^k_{rd}\le l_{rd}\xi^k_{rd} \quad r\in\mathcal{R}_d, k\in\mathcal{K}, d\in\mathcal{D}, \tag{2b}$$

$$\phi^k_{rd}\ge 0 \quad r\in\mathcal{R}_d, k\in\mathcal{K}, d\in\mathcal{D}. \tag{2c}$$

Similar ways may also be found that are used to model arrival times at customers in a set-partitioning-like formulation for VRPs (e.g., see Spliet and Gabor 2015). In the remainder of this paper, we refer to F1 with the route-based formulation using the enhanced TC constraints (2a)–(2c). We denote by LF1 the LP-relaxation of F1, and by z(LF1) the corresponding optimal cost.

2.3. Schedule-based Formulation F2

Let $\mathcal{H}$ be the index set of feasible schedules, where each schedule $h \in \mathcal{H}$ is a collection of D routes to be operated by a driver over the planning horizon. Let $\Psi_h=\{r^h_1, r^h_2, ..., r^h_D\}$ denote the vector of working routes in schedule h, where $r^h_d \in \mathcal{R}_d$ indicates the index of routes operated on day $d \in \mathcal{D}$ in schedule h, and $r^h_d=0$ indicates that no route is operated on day d. To model RC, a proportion of at most $\mathbf{F}$ of different arcs of routes in each schedule $h \in \mathcal{H}$ can be traversed, i.e.,

$\sum_{(i,j)\in\mathcal{A}} w^h_{ij} \leq \mathbf{F} \sum_{d\in\mathcal{D}} \sum_{(i,j)\in\mathcal{A}} e_{r^h_d ij}$, where for each arc $(i,j)\in\mathcal{A}$, $w^h_{ij}=0$ if $e_{r^h_d ij}=0$ for all $d\in\mathcal{D}$ and $w^h_{ij}=1$ otherwise. Each schedule $h\in\mathcal{H}$ has total travel duration $t^H_h=\sum_{d\in\mathcal{D}} t^R_{r^h_d,d}$. Let a_{hid} be a binary parameter that equals 1 if schedule h serves customer i on day d and equals 0 otherwise. In addition, a binary parameter o_{hi} is introduced that equals 1 if schedule h serves customer i on at least one day and equals 0 otherwise. Similar to F1, values b_{hid} and l_{hd} denote the accumulated time and latest departure time, respectively, which are used to define the enhanced constraints for TC. By defining the additional sets of variables: (i) ξ_h, a binary variable that equals 1 if schedule $h\in\mathcal{H}$ is operated and equals 0 otherwise and (ii) ϕ_{hd}, a nonnegative continuous variable that determines the departure time of the route operated on day $d\in\mathcal{D}$ of schedule $h\in\mathcal{H}$, the schedule-based formulation F2 is as follows:

$$z(\mathrm{F2}) = \min \sum_{h\in\mathcal{H}} t^H_h \xi_h \tag{3a}$$

$$\text{s.t.} \sum_{h\in\mathcal{H}} a_{hid}\xi_h = g_{id} \quad i\in\mathcal{N}, d\in\mathcal{D}, \tag{3b}$$

$$\sum_{h\in\mathcal{H}} \xi_h \leq K, \tag{3c}$$

$$\sum_{h\in\mathcal{H}} o_{hi}\xi_h \leq \mathbf{E} \quad i\in\mathcal{N}, \tag{3d}$$

$$v_{id} = \sum_{h\in\mathcal{H}} a_{hid}\phi_{hd} + \sum_{h\in\mathcal{H}} b_{hid}\xi_h \quad i\in\mathcal{N}_d, d\in\mathcal{D}, \tag{3e}$$

$$\phi_{hd} \leq l_{hd}\xi_h \quad h\in\mathcal{H}, d\in\mathcal{D}, \tag{3f}$$

$$v_{id} + s_{id} + t_{i0} \leq T \quad i\in\mathcal{N}_d, d\in\mathcal{D}, \tag{3g}$$

$$(v_{id} - v_{id'})g_{id}g_{id'} \leq \mathbf{L} \quad i\in\mathcal{N}, d, d'\in\mathcal{D} : d\neq d', \tag{3h}$$

$$\xi_h \in \{0,1\} \quad h\in\mathcal{H}, \tag{3i}$$

$$\phi_{hd} \geq 0 \quad h\in\mathcal{H}, d\in\mathcal{D}, \tag{3j}$$

$$v_{id} \geq 0 \quad i\in\mathcal{N}_d, d\in\mathcal{D}. \tag{3k}$$

Objective (3a) is to minimize the total travel time. Constraints (3b) guarantee that each customer must be served on each day that requires a service. Constraint (3c) limits the number of schedules selected in the solution to the number of available drivers. Constraints (3d) model the DC level $\mathbf{E}$ for each customer, whereas constraints (3e)–(3h) link variables $\boldsymbol{v}$, $\boldsymbol{\phi}$, and $\boldsymbol{\xi}$ to model the TC level $\mathbf{L}$ for each customer. Constraints (3i)–(3k) define the domain of the decision variables. In the formulation, RC is implicitly modeled by the definition of each schedule h.

We denote by LF2 and z(LF2) the LP-relaxations of F2 and its optimal solution cost, respectively.

3. Overview of the Exact Method

This section provides an overview of the exact method proposed to solve the VRPGCR-SDC that relies on route and schedule generation–based approaches (similar to Baldacci et al. 2008, Mingozzi et al. 2013). A set-partitioning model with exponentially many integer columns can be equivalently solved by a reduced one with a smaller column set that includes any columns used for an optimal solution. The method consists of combining different dual ascent procedures to find a near optimal dual solution of the set-partitioning model used to generate a reduced problem containing all optimal integer solutions. A general integer programming (IP) solver is then used to solve the reduced problem, using a known upper bound as a cutoff. The key observation is that a route can only be part of a solution that improves the best upper bound if its reduced cost is smaller than the gap between the upper bound and the cost of the dual solution.

The following proposition extends this observation to MIPs, whose continuous variables all have zero coefficients in the objective function, which is the case for formulations F1 and F2.

PROPOSITION 1. *Let $\boldsymbol{P}$ be an MIP defined as $z(\boldsymbol{P}) = \min\{\boldsymbol{cx} | s.t.\ \boldsymbol{Ax} + \boldsymbol{By} = \boldsymbol{b}, \boldsymbol{x} \in \{0,1\}^{n_1}, \boldsymbol{y} \in \mathbb{R}_+^{n_2}\}$. Let $\boldsymbol{x}$ be a primal feasible solution of cost ub, and let $\boldsymbol{\omega}$ be a dual feasible solution of the LP-relaxation of $\boldsymbol{P}$ of cost $z'(\boldsymbol{\omega})$. Any optimal solution $\boldsymbol{x}^*$ of cost $z(\boldsymbol{P})$ less than or equal to ub cannot contain a variable $x_i^* = 1$ having a reduced cost c_i' (computed with respect to the dual solution $\boldsymbol{\omega}$) greater than $ub - z'(\boldsymbol{\omega})$. (See §EC.2.1 for the proof)*

Let Π_1 be a dual feasible solution of LF1 of cost $z'(\Pi_1)$, and let ub represent an upper bound on the optimal solution cost of the VRPGCR-SDC. Let $\bar{\mathcal{R}}_d$ denote the subset of routes on day d, where the reduced cost of each variable ξ_{rd}^k with $k \in \mathcal{K}$, $d \in \mathcal{D}$, and $r \in \bar{\mathcal{R}}_d$ is less than or equal to $ub - z'(\Pi_1)$. Based on Proposition 1, the route set $\bar{\mathcal{R}} = \cup_{d\in\mathcal{D}}\bar{\mathcal{R}}_d$ is a candidate *optimal* route set for F1, which contains all routes of any optimal solution of cost less than or equal to ub. Given that the optimal route set $\bar{\mathcal{R}}$ represents a subset of $\mathcal{H}$ in F2, each schedule $h \in \hat{\mathcal{H}}$ is composed of routes r_d^h from $\bar{\mathcal{R}}_d \cup \{0\}$ on day $d \in \mathcal{D}$. Hence, we can define *a reduced schedule-based formulation* RF2, which is obtained from F2 by replacing the schedule set $\mathcal{H}$ with the *optimal* schedule set $\hat{\mathcal{H}}$. Let (LRF2) be the LP-relaxation of formulation RF2. Clearly, we have $z(\text{RF2}) = z(\text{F2})$ and $z(\text{LRF2}) \geq z(\text{LF2})$. In the exact method, the generation of schedules for RF2 is based on the optimal route set $\bar{\mathcal{R}}$. In addition, based on Proposition 1, we can further reduce the size of the resulting schedule set $\hat{\mathcal{H}}$. Indeed, given a dual feasible solution Π_2 of LRF2 of cost $z'(\Pi_2)$ and an upper bound ub, we define $\bar{\mathcal{H}} \subseteq \hat{\mathcal{H}}$ as the candidate optimal schedule set, where the reduced cost of each variable ξ_h for $h \in \bar{\mathcal{H}}$ with respect to Π_2 is no greater than $ub - z'(\Pi_2)$. Given the generated optimal route set $\bar{\mathcal{R}}$ and optimal schedule set $\bar{\mathcal{H}}$, the VRPGCR-SDC can be solved to optimality by the direct use of an

MIP solver applied to formulation F1 over the route set $\bar{\mathcal{R}}$ or to formulation RF2 over the schedule set $\bar{\mathcal{H}}$. The efficiency of this approach is related to the cardinality of the final sets $\bar{\mathcal{R}}$ and $\bar{\mathcal{H}}$, whose sizes depend on the gap between the ub and the costs of the dual solutions.

We next introduce two relaxations in §3.1 and the exact method is described in §3.2.

3.1. Relaxations R1 and R2

In this section, we describe two LP relaxations of formulations F1 and RF2 that can be used to compute tight lower bounds on the optimal solution cost of the VRPGCR-SDC.

Relaxation R1 of F1. The first relaxation R1 is obtained from formulation LF1 as follows.

i) Let $\mathcal{C}_d = \{C \subseteq \mathcal{N}_d : |C| = 3\}$ be the set of all triplets of the customers requiring service on day d, and let $\mathcal{C}(r)$ be the set of all triplets for which at least two customers are visited by route r (a subset of $\mathcal{C} = \cup_{d\in\mathcal{D}}\mathcal{C}_d$). We add to LF1 the following subset-row (SR3) inequalities:

$$\sum_{r\in\mathcal{R}_d: C\in\mathcal{C}(r)} \sum_{k\in\mathcal{K}} \xi_{rd}^k \le 1, C \in \mathcal{C}_d, d \in \mathcal{D}, \tag{4}$$

which correspond to a subset of subset-row inequalities proposed by Jepsen et al. (2008).

ii) We define continuous variables ζ_{rd} in $[0,1]$ and continuous variables $\eta_{rd} \ge 0$ that represent whether route r is selected in the solution on day d and the departure time of route r on day d, respectively. Due to constraints (1b) (the same route r can be operated by at most one driver on each day d), we have $\zeta_{rd} = \sum_{k\in\mathcal{K}} \xi_{rd}^k$ and $\eta_{rd} = \sum_{k\in\mathcal{K}} \phi_{rd}^k$. We also define variables $\tilde{w}_{ij}$ such that $\tilde{w}_{ij} = \sum_{k\in\mathcal{K}} w_{ij}^k$. We replace all of the constraints in LF1 with surrogate constraints, where "surrogate constraints" mean aggregated (or reformulated) constraints that remove the dimension of drivers $k \in \mathcal{K}$.

iii) We aggregate the column-dependent rows (2b) for all $r \in \mathcal{R}_d$.

Full details of these steps are given in the e-companion to this paper (see §EC.3). The resulting relaxation R1 is as follows:

$$z(\text{R1}) = \min \sum_{d\in\mathcal{D}} \sum_{r\in\mathcal{R}_d} t_{rd}^R \zeta_{rd} \tag{5a}$$

$$\text{s.t.} \sum_{r\in\mathcal{R}_d} a_{ri}\zeta_{rd} = g_{id} \quad i \in \mathcal{N}, d \in \mathcal{D}, \tag{5b}$$

$$\sum_{r\in\mathcal{R}_d} \zeta_{rd} \le K \quad d \in \mathcal{D}, \tag{5c}$$

$$\sum_{r\in\mathcal{R}_d} e_{rij}\zeta_{rd} \le \tilde{w}_{ij} \quad (i,j) \in \mathcal{A}, d \in \mathcal{D}, \tag{5d}$$

$$\sum_{(i,j)\in\mathcal{A}} \tilde{w}_{ij} \le \mathbf{F} \sum_{d\in\mathcal{D}} \sum_{(i,j)\in\mathcal{A}} \sum_{r\in\mathcal{R}_d} e_{rij}\zeta_{rd}, \tag{5e}$$

$$v_{id} = \sum_{r\in\mathcal{R}_d} a_{ri}\eta_{rd} + \sum_{r\in\mathcal{R}_d} b_{rid}\zeta_{rd} \quad i\in\mathcal{N}_d, d\in\mathcal{D}, \tag{5f}$$

$$\sum_{r\in\mathcal{R}_d} \eta_{rd} \leq \sum_{r\in\mathcal{R}_d} l_{rd}\zeta_{rd} \quad d\in\mathcal{D}, \tag{5g}$$

$$v_{id} + s_{id} + t_{i0} \leq T \quad i\in\mathcal{N}_d, d\in\mathcal{D}, \tag{5h}$$

$$(v_{id} - v_{id'})g_{id}g_{id'} \leq \mathbf{L} \quad i\in\mathcal{N}, d, d'\in\mathcal{D}: d\neq d', \tag{5i}$$

$$\sum_{r\in\mathcal{R}_d : C\in\mathcal{C}(r)} \zeta_{rd} \leq 1 \quad C\in\mathcal{C}_d, d\in\mathcal{D}, \tag{5j}$$

$$\zeta_{rd} \in [0,1] \quad r\in\mathcal{R}_d, d\in\mathcal{D}, \tag{5k}$$

$$\tilde{w}_{ij} \in [0,1] \quad (i,j)\in\mathcal{A}, \tag{5l}$$

$$\eta_{rd} \geq 0 \quad r\in\mathcal{R}_d, d\in\mathcal{D}, \tag{5m}$$

$$v_{id} \geq 0 \quad i\in\mathcal{N}_d, d\in\mathcal{D}. \tag{5n}$$

Note that it is sufficient to define variable $\tilde{w}_{ij}$ to take binary values due to constraints (5b), (5d) and (5e). Relaxation R1 avoids the use of the driver dimension $k\in\mathcal{K}$, a drawback of formulation LF1 being symmetric in the vehicles (the fleet is homogeneous). As a result, DC restriction is not handled in R1, and RC restriction is partially relaxed (constraints (5d) and (5e)). It is worth noting that the resulting formulation is similar to the model formulation proposed in Spliet and Gabor (2015) for the time windows assignment vehicle routing problem considering no exogenous time windows. In addition, column-dependent rows are not present in the formulation, thus simplifying the corresponding column generation procedure. Note that our aggregation technique of handling column-dependent rows will not lose optimality in our exact solution method as shown by the following theorem. The main intuition is that R1 is a relaxation of LF1 (with SR3 inequalities) in the aggregated fashion, and thus any feasible dual solution to R1 can be leveraged to reconstruct a feasible one to LF1.

THEOREM 1. *Given any dual feasible solution* Π_1 *of* R1*, one can always construct a corresponding dual feasible solution* Π_1' *for* LF1 *strengthened with SR3 inequalities such that their dual objective values are equal. (See §EC.2.2 for the proof)*

Based on this result and given an upper bound on the optimal solution cost, the dual solution Π_1' of LF1 with SR3 inequalities can be used to perform route generation to derive the reduced route set $\bar{\mathcal{R}}$.

Relaxation R2 of RF2. Relaxation R2 is obtained from formulation LRF2 (i.e., from formulation LF2 by replacing the schedule set $\mathcal{H}$ with the schedule set $\hat{\mathcal{H}}$ generated using the reduced route set $\bar{\mathcal{R}}$ from F1), replacing continuous variables ξ_h and ϕ_{hd} with ζ_h and η_{hd} for clarity, respectively, and aggregating constraints (3f) into constraints (6f) to avoid column-dependent rows. The resulting formulation, further strengthened with SR3 inequalities, is as follows:

$$z(\mathrm{R2}) = \min \sum_{h\in\hat{\mathcal{H}}} t_h^H \zeta_h \tag{6a}$$

$$\text{s.t.} \sum_{h\in\hat{\mathcal{H}}} a_{hid}\zeta_h = g_{id} \quad i\in\mathcal{N}, d\in\mathcal{D}, \tag{6b}$$

$$\sum_{h\in\hat{\mathcal{H}}} \zeta_h \le K, \tag{6c}$$

$$\sum_{h\in\hat{\mathcal{H}}} o_{hi}\zeta_h \le \mathbf{E} \quad i\in\mathcal{N}, \tag{6d}$$

$$v_{id} = \sum_{h\in\hat{\mathcal{H}}} a_{hid}\eta_{hd} + \sum_{h\in\hat{\mathcal{H}}} b_{hid}\zeta_h \quad i\in\mathcal{N}_d, d\in\mathcal{D}, \tag{6e}$$

$$\sum_{h\in\hat{\mathcal{H}}} \eta_{hd} \le \sum_{h\in\hat{\mathcal{H}}} l_{hd}\zeta_h \quad d\in\mathcal{D}, \tag{6f}$$

$$v_{id} + s_{id} + t_{i0} \le T \quad i\in\mathcal{N}_d, d\in\mathcal{D}, \tag{6g}$$

$$(v_{id} - v_{id'})g_{id}g_{id'} \le \mathbf{L} \quad i\in\mathcal{N}, d, d'\in\mathcal{D} : d\neq d', \tag{6h}$$

$$\sum_{h\in\hat{\mathcal{H}}:C\in\mathcal{C}(r_d^h)} \zeta_h \le 1 \quad C\in\mathcal{C}_d, d\in\mathcal{D}, \tag{6i}$$

$$\zeta_h \in [0,1] \quad h\in\hat{\mathcal{H}}, \tag{6j}$$

$$\eta_{hd} \ge 0 \quad h\in\hat{\mathcal{H}}, d\in\mathcal{D}, \tag{6k}$$

$$v_{id} \ge 0 \quad i\in\mathcal{N}_d, d\in\mathcal{D}. \tag{6l}$$

The following proposition shows the relation between relaxations R1 and R2, resulting from the fact that feasible schedule in R2 always can be decomposed to individual feasible routes on planning days in R1.

Proposition 2. *The following inequality holds:* $z(\mathrm{R2}) \ge z(\mathrm{R1})$*. (See §EC.2.3 for the proof)*

The computational results reported in §6 show that, in practice, the inequality can be strict, especially for problems with more constrained consistency requirements.

Given that R2 is a relaxation of LF2 (with SR3 inequalities), any feasible dual solution to R2 can be leveraged to reconstruct a feasible solution of LF2. This leads to the following theorem, which shows the correctness of using R2 for our exact algorithm.

Theorem 2. *Given any dual feasible solution* Π_2 *of* R2*, one can always construct a corresponding dual feasible solution* Π_2' *for* LRF2 *strengthened with SR3 inequalities such that their dual objective values are equal. (See §EC.2.4 for the proof)*

Based on this result and given an upper bound on the optimal solution cost, the dual solution Π_2' of LRF2 with SR3 inequalities can be used to perform a schedule generation to derive the reduced schedule set $\bar{\mathcal{H}}$, by using the optimal routes from $\bar{\mathcal{R}}$.

3.2. An Exact Algorithm

Based on the two relaxations introduced in the previous section, the exact method performs the following four major steps.

Step 1. Solve R1 via a CCG procedure to obtain its dual optimal solution Π_1^* of cost $z'(\Pi_1^*)$ (see §4). Set $lb_1 = z'(\Pi_1^*)$ and $ub = (1+\epsilon)\cdot lb_1$, where $\epsilon \in (0,1]$ is an assumed gap.

Step 2. Given Π_1^*, lb_1 and ub, compute the candidate optimal route sets $\bar{\mathcal{R}}_d$ for $d \in \mathcal{D}$ (see §4.2). If $|\cup_{d\in\mathcal{D}} \bar{\mathcal{R}}_d|$ is large (e.g., greater than 5,000), go to Step 3; otherwise, solve the reduced problem of F1 with the candidate optimal route set by an MIP solver. Let ub_1 be the optimal solution cost of the reduced problem of F1. We have the following cases: (i) If $ub_1 \leq ub$, then the algorithm terminates with $ub = ub_1$ as the optimal solution cost of the VRPGCR-SDC; (ii) If $ub_1 > ub$, set $ub = ub_1$, and repeat Step 2; (iii) If no feasible solution is found and $ub \leq 2\cdot lb_1$, then set $ub = 1.05\cdot ub$, and repeat Step 2; otherwise, the algorithm terminates without having found any feasible solution.

Step 3. Define set $\hat{\mathcal{H}}$ using the set of routes $\bar{\mathcal{R}} = \cup_{d\in\mathcal{D}}\bar{\mathcal{R}}_d$, and solve R2 via a CCG procedure to obtain its dual optimal solution Π_2^* of cost $z'(\Pi_2^*)$ (see §5). Set $lb_2 = z'(\Pi_2^*)$.

Step 4. Given Π_2^*, lb_2, and ub, compute the optimal schedule set $\bar{\mathcal{H}}$ (see §5.2). Solve the problem obtained from RF2 (or F2) by replacing set $\hat{\mathcal{H}}$ (or $\mathcal{H}$) with set $\bar{\mathcal{H}}$ using an MIP solver, and let ub_2 be the corresponding optimal solution cost. We have two cases: (i) If $ub_2 \leq ub$, then the algorithm terminates with $ub = ub_2$ as the optimal solution cost for the VRPGCR-SDC; (ii) If $ub_2 > ub$, set $ub = ub_2$, and go to Step 2; (iii) If no feasible solution is found and $ub \leq 2\cdot lb_1$, then set $ub = 1.05\cdot ub$, and go to Step 2; otherwise, the algorithm terminates without having found any feasible solution.

Note that this algorithm does not require the computation of an initial feasible solution and its corresponding upper bound value. It assumes that the gap between the lower bound and the optimal solution cost is less than ϵ; for example, $\epsilon = 10\%$ used in our experiments is a sufficient assumption. With this assumption, we only encountered case (i) in Step 2 or Step 4 in our experiments such that the optimal solution was obtained without repeating any steps. It is worth noting that if the algorithm terminates without finding any feasible solution in Step 4, one could still refer to an alternative method based on some of the current algorithmic components for the solution. For instance, the proposed CCG procedure can be embedded in a branch-and-bound framework, formally known as a branch-price-and-cut algorithm, that serves as another exact solution method for our problem.

4. Column-and-Cut Generation Procedure for R1 and Route Set $\bar{\mathcal{R}}$

In this section, we describe a CCG algorithm based on the simplex method to solve relaxation R1 and the procedure used to generate the optimal route set $\bar{\mathcal{R}}$.

The route set $\mathcal{R}'_d$, $d \in \mathcal{D}$, of the initial restricted master problem (denoted as RMP1) is initialized with a dummy route containing all customers on day d and with an extremely large route cost. The initial set of SR3 inequalities (5j) is set to equal the empty set, and the inequalities are separated by complete enumeration.

The dual vectors associated with constraints (5b)–(5j) of R1 are denoted by $\boldsymbol{\lambda} = \{\lambda_{id} \in \mathbb{R} | i \in \mathcal{N}, d \in \mathcal{D}\}$, $\boldsymbol{\mu} = \{\mu_d \leq 0 | d \in \mathcal{D}\}$, $\boldsymbol{\gamma} = \{\gamma_{ijd} \leq 0 | (i,j) \in \mathcal{A}, d \in \mathcal{D}\}$, $\varepsilon \leq 0$, $\boldsymbol{\pi} = \{\pi_{id} \in \mathbb{R} | i \in \mathcal{N}_d, d \in \mathcal{D}\}$, $\boldsymbol{\tau} = \{\tau_d \leq 0 | d \in \mathcal{D}\}$, $\boldsymbol{\iota} = \{\iota_{id} \leq 0 | i \in \mathcal{N}_d, d \in \mathcal{D}\}$, $\boldsymbol{\kappa} = \{\kappa_{idd'} \leq 0 | i \in \mathcal{N}, d, d' \in \mathcal{D} : d \neq d'\}$, and $\boldsymbol{\chi} = \{\chi_{Cd} \leq 0 | C \in \mathcal{C}_d, d \in \mathcal{D}\}$.

Let $c'(\zeta_{rd}) = t^R_{rd} - \sum_{i \in \mathcal{N}} a_{ri} \lambda_{id} - \mu_d - \sum_{(i,j) \in \mathcal{A}} e_{rij} \gamma_{ijd} + \mathbf{F} \varepsilon \sum_{(i,j) \in \mathcal{A}} e_{rij} - \sum_{i \in \mathcal{N}_d} b_{rid} \pi_{id} + l_{rd} \tau_d - \sum_{C \in \mathcal{C}(r)} \chi_{Cd}$ denote the reduced cost of variable ζ_{rd}, and let $c'(\eta_{rd}) = -\tau_d - \sum_{i \in \mathcal{N}_d} a_{ri} \pi_{id}$ denote the reduced cost of variable η_{rd}. At each iteration of CCG, the dual solution of the RMP1 is used to check whether $\min\{c'(\zeta_{rd}), c'(\eta_{rd})\} \geq 0$ is satisfied for each $d \in \mathcal{D}$ and $r \in \mathcal{R}_d$. The CCG algorithm is as follows:

Algorithm 1 Column-and-Cut Generation for R1

Step 0: Initialize the RMP1;

Step 1: Solve the RMP1 to obtain its optimal primal and dual solutions;

Step 2: Solve the pricing problem PP1, that is, $\min_{r \in \mathcal{R}_d \setminus \mathcal{R}'_d} \{c'(\zeta_{rd})\}$ for $d \in \mathcal{D}$, using a DP-based approach (see §4.1);

Step 3: If $\min_{r \in \mathcal{R}_d \setminus \mathcal{R}'_d} \{c'(\zeta_{rd})\} \geq 0$ for all $d \in \mathcal{D}$, go to Step 4; otherwise, add to the RMP1 the route having the most negative reduced cost $c'(\zeta_{rd})$ to $\mathcal{R}'_d$, $d \in \mathcal{D}$, and go to Step 1;

Step 4: If $\min_{r \in \mathcal{R}_d} \{-\sum_{i \in \mathcal{N}'_d} a_{ri} \pi_{id}\} - \tau_d \geq 0$ for $d \in \mathcal{D}$, that is, $c'(\eta_{rd}) \geq 0$, where $\mathcal{N}'_d = \{i \in \mathcal{N}_d | \pi_{id} > 0\}$, go to Step 5; otherwise, set $\tau_d = \min_{r \in \mathcal{R}_d} \{-\sum_{i \in \mathcal{N}'_d} a_{ri} \pi_{id}\}$ for $d \in \mathcal{D}$ (*dual adjustment*), and go to Step 2;

Step 5: If SR3 inequalities (5j) are all satisfied by the primal solution of the RMP1, the algorithm terminates; otherwise, add all violated SR3 inequalities to the RMP1, and go to Step 1.

To prove the optimality of the dual solution, our CCG algorithm for R1 (see Algorithm 1) solves the D pricing subproblems of variables ζ_{rd} (denoted as PP1), that is, $\min_{r \in \mathcal{R}_d} \{c'(\zeta_{rd})\}$ for $d \in \mathcal{D}$ followed by a *dual adjustment* operation to guarantee $\min_{r \in \mathcal{R}_d} \{c'(\eta_{rd})\} \geq 0$ for $d \in \mathcal{D}$, where the term $\min_{r \in \mathcal{R}_d} \{-\sum_{i \in \mathcal{N}'_d} a_{ri} \pi_{id}\}$ can be easily computed as a minimum-cost cycle for customer set $\mathcal{N}'_d$ with customer weight $-\pi_{id}$ for all $i \in \mathcal{N}'_d$ on day d using dynamic programming (DP).

If $\min\{c'(\zeta_{rd})\} \geq 0$ and $\min\{c'(\eta_{rd})\} < 0$, the dual adjustment operation changes τ_d within its feasible dual region to construct a new dual solution having the same dual bound, because the coefficient of variable τ_d is 0 in the dual objective function. This new dual solution will force $c'(\eta_{rd})$ to satisfy the dual optimality condition. However, it may induce that $c'(\zeta_{rd}) < 0$, and PP1 needs to be solved again.

In general, the DP aims to check the satisfaction of the dual constraints corresponding to variables ζ_{rd}, and the dual adjustment operation aims to enforce the satisfaction of the dual constraints corresponding to variables η_{rd}. The correctness of Algorithm 1 is shown by the following proposition.

PROPOSITION 3. *Algorithm 1 solves the relaxation* R1 *to optimality. (See §EC.2.5 for the proof)*

4.1. Solving the Pricing Problem PP1

Let $\mathcal{G}_d = (\mathcal{N}_d^+, \mathcal{A}_d)$ be a subgraph of graph $\mathcal{G}$ associated with day d. PP1 decomposes into D elementary shortest paths with resource constraints on weighted subgraphs $\mathcal{G}_d$, whose node and arc weights depend on the current dual solution of the RMP1. Our pricing subproblem generalizes the one proposed by Dell'Amico et al. (2006) because the node cost depends on the arrival time at the node. The mathematical formulation of PP1 is given in the e-companion (see §EC.4.1).

In this section, we extend the DP approach of Dell'Amico et al. (2006) by embedding sophisticated exploration strategies and dominance rules. We describe the algorithm for a given $d \in \mathcal{D}$, and thus we omit the use of the index d. Hereafter, we refer to customers and nodes interchangeably.

Define $\bar{\mathcal{G}} = \mathcal{G}_d$ with the node set $\bar{\mathcal{N}}^+ = \bar{\mathcal{N}} \cup \{0\} = \mathcal{N}_d^+$ and the arc set $\bar{\mathcal{A}} = \mathcal{A}_d$, and set $\bar{\mathcal{C}} = \mathcal{C}_d$. Each node $i \in \bar{\mathcal{N}}$ has delivery demand $\bar{p}_i = p_{id}$, pickup demand $\bar{q}_i = q_{id}$, and service time $\bar{s}_i = s_{id}$. Let $\bar{\pi}_i = -\pi_{id}$, $\bar{t}_{ij} = t_{ij} - \gamma_{ijd} + \mathbf{F}\varepsilon - \tau_d t_{ij}$, $\bar{\lambda}_i = -\lambda_{id} - \tau_d s_{id}$, and $\bar{\chi}_C = -\chi_{Cd}$, where $-\tau_d t_{ij}$ and $-\tau_d s_{id}$ aim to calculate $\tau_d l$ in the reduced cost function. For a path with time length L and with the latest departure time l, we have $\tau_d l = \tau_d(T - L) = \tau_d T - \tau_d L$. Because L includes arc travel times and node service times along the path, term $-\tau_d L$ can be split into terms $-\tau_d t_{ij}$ and $-\tau_d s_{id}$ for each traversed arc and node. Then, combining the constant term $\tau_d T$ and $-\mu_d$, we define the fixed cost of a path by $\bar{\mu} = \tau_d T - \mu_d$. Overall, the DP aims to find the minimum-cost cycle on graph $\bar{\mathcal{G}}$ passing through depot 0 subject to capacity and travel time upper limits Q and T. In the following, we describe forward and backward DP algorithms with dominance rules and acceleration strategies to solve PP1.

Bidirectional Labeling. A bidirectional labeling algorithm (i.e., a bidirectional DP algorithm) combining forward and backward labels is adopted to generate routes for PP1 (see Righini and Salani (2006)). We denote a label that represents a partial forward path departing from the depot and ending at customer i by $(i,\psi,\omega,\mathbb{S},b)$, where the vehicle holds ψ available capacity to collect packages from future customers and holds ω available capacity to distribute packages to future customers after serving customer i. Set $\mathbb{S}$ denotes the group of nodes visited by the current partial path, and parameter b indicates the time from the departure at the depot to the arrival at node i in the partial path. Let $\sigma(i,\psi,\omega,\mathbb{S},b)$ be the reduced cost associated with the label $(i,\psi,\omega,\mathbb{S},b)$. A label $(i,\psi,\omega,\mathbb{S},b)$ with $i=0$ and $\mathbb{S}\neq\varnothing$ represents a route in $\bar{\mathcal{G}}$.

The forward extension starts from the initial label $(i^0,\psi^0,\omega^0,\mathbb{S}^0,b^0)$, where $i^0=0$, $\psi^0=Q$, $\omega^0=Q$, $\mathbb{S}^0=\varnothing$, $b^0=0$, and its reduced cost $\sigma(i^0,\psi^0,\omega^0,\mathbb{S}^0,b^0)=0$. A partial path is extended by linking its end node i to another node j via an arc (i,j) to generate a new label $(j,\psi',\omega',\mathbb{S}',b')$. Each extension involves the following updates: (i) $\mathbb{S}'=\mathbb{S}\cup\{j\}$; (ii) $b'=b+\bar{s}_i+t_{ij}$; (iii) $\psi'=\psi-\bar{q}_j$; (iv) $\omega'=\min\{\omega-\bar{p}_j,\psi'\}$; and (v) $\sigma(j,\psi',\omega',\mathbb{S}',b')=\sigma(i,\psi,\omega,\mathbb{S},b)+\bar{t}_{ij}+\bar{\lambda}_j+\bar{\pi}_j b'+\sum_{C\in\bar{\mathcal{C}}:|\mathbb{S}\cap C|=1,j\in C}\bar{\chi}_C$. Each extension is admissible if $j\notin\mathbb{S}$, $b+\bar{s}_i+t_{ij}+\bar{s}_j+t_{j0}\leq T$ and $\min\{\omega-\bar{p}_j,\psi'\}\geq 0$, as shown by Proposition 4. The forward extension stops when no new labels can be generated.

Proposition 4. *Given a label $(i,\psi,\omega,\mathbb{S},b)$ and an arc (i,j) with $j\notin\mathbb{S}$ and $b+\bar{s}_i+t_{ij}+\bar{s}_j+t_{j0}\leq T$, extending the forward path to node j is feasible if and only if $\psi-\bar{q}_j\geq 0$ and $\min\{\omega-\bar{p}_j,\psi'\}\geq 0$. (See §EC.2.6 for the proof)*

Backward labels are generated similarly; the details are omitted for the sake of brevity.

A speed-up in solving PP1 can be obtained by removing dominated (forward/backward) paths. A dominated path is either a path that cannot lead to a feasible route or a path such that any route containing it cannot be part of any optimal solution.

The following dominance rules are defined by also incorporating the SR3 inequalities.

Proposition 5. *Given two labels $L_1=(i,\psi^1,\omega^1,\mathbb{S}^1,b^1)$ and $L_2=(i,\psi^2,\omega^2,\mathbb{S}^2,b^2)$, if (i) $\mathbb{S}^1\subseteq\mathbb{S}^2$, (ii) $\omega^1\geq\omega^2$, (iii) $b^1\leq b^2$, and (iv) $\sigma(i,\psi^1,\omega^1,\mathbb{S}^1,b^1)-\sigma(i,\psi^2,\omega^2,\mathbb{S}^2,b^2)\leq\sum_{j\in\bar{\mathcal{N}}\setminus\mathbb{S}^2}[\bar{\pi}_j]^-(b^2-b^1)+\sum_{C\in\bar{\mathcal{C}}:|\mathbb{S}^1\cap C|=1,|\mathbb{S}^2\setminus\mathbb{S}^1\cap C|=1,|\bar{\mathcal{N}}\setminus\mathbb{S}^2\cap C|=1}\bar{\chi}_C$, then label L_1 dominates label L_2. (See §EC.2.7 for the proof)*

A similar dominance holds for backward labels. State control techniques are also used to speed up the CCG procedure. The corresponding details can be found in the e-companion (see §EC.4.2).

4.2. Generating the Route Set $\bar{\mathcal{R}}_d$

At the end of CCG Algorithm 1, the optimal dual solution of R1 $\Pi_1^*=(\boldsymbol{\lambda}^*,\boldsymbol{\mu}^*,\boldsymbol{\gamma}^*,\varepsilon^*,\boldsymbol{\pi}^*,\boldsymbol{\tau}^*,\boldsymbol{\iota}^*,\boldsymbol{\kappa}^*,\boldsymbol{\chi}^*)$ is transformed into an equivalent dual solution Π_1' of LF1 strengthened with the same SR3 inequalities identified by CCG Algorithm 1 (see Theorem 1).

To generate the route set $\bar{\mathcal{R}}_d$ used by the exact algorithm given the dual solution Π_1' and the current upper bound ub, we use a pure forward DP procedure based on the forward strategy described in Section 4.1 in which the dominance rule described in Proposition 5 is not used.

The candidate optimal route set $\bar{\mathcal{R}} = \cup_{d\in\mathcal{D}}\bar{\mathcal{R}}_d$ contains any route r for a day d having a reduced cost $c'(\xi_{rd}^k)$ computed with respect to the dual solution Π_1' less than or equal to $ub - z'(\Pi_1^*)$.

5. Column-and-Cut Generation Procedure for R2 and Schedule Set $\bar{\mathcal{H}}$

In this section, we describe a CCG algorithm based on the simplex method to solve the relaxation R2 that uses the route set $\bar{\mathcal{R}}$ generated after solving R1. The section also describes the procedure used to generate the final schedule set $\bar{\mathcal{H}}$.

The schedule set $\hat{\mathcal{H}}'$ of the initial restricted master problem (denoted RMP2) is initialized with a dummy schedule covering all customers over all days and with an extremely large route cost. The initial set of SR3 inequalities (6i) is set equal to the empty set, and the inequalities are separated by complete enumeration.

The dual vectors associated with constraints (6b)–(6i) are denoted by $\hat{\boldsymbol{\lambda}} = \{\hat{\lambda}_{id} \in \mathbb{R} | i \in \mathcal{N}, d \in \mathcal{D}\}$, $\hat{\boldsymbol{\mu}} = \{\hat{\mu} \le 0\}$, $\hat{\boldsymbol{\beta}} = \{\hat{\beta}_i \le 0 | i \in \mathcal{N}\}$, $\hat{\boldsymbol{\pi}} = \{\hat{\pi}_{id} \in \mathbb{R} | i \in \mathcal{N}_d, d \in \mathcal{D}\}$, $\hat{\boldsymbol{\tau}} = \{\hat{\tau}_d \le 0 | d \in \mathcal{D}\}$, $\hat{\boldsymbol{\iota}} = \{\hat{\iota}_{id} \le 0 | i \in \mathcal{N}_d, d \in \mathcal{D}\}$, $\hat{\boldsymbol{\kappa}} = \{\hat{\kappa}_{idd'} \le 0 | i \in \mathcal{N}, d, d' \in \mathcal{D} : d \ne d'\}$, and $\hat{\boldsymbol{\chi}} = \{\hat{\chi}_{Cd} \le 0 | C \in \mathcal{C}_d, d \in \mathcal{D}\}$.

In the CCG algorithm (see Algorithm 2), the pricing problem (denoted as PP2) aims to find a schedule $h \in \hat{\mathcal{H}} \setminus \hat{\mathcal{H}}'$ having the most negative reduced cost $c'(\zeta_h) = t_h^H - \sum_{d\in\mathcal{D}}\sum_{i\in\mathcal{N}} a_{hid}\hat{\lambda}_{id} - \hat{\mu} - \sum_{i\in\mathcal{N}} o_{hi}\hat{\beta}_i - \sum_{d\in\mathcal{D}}\sum_{i\in\mathcal{N}_d} b_{hid}\hat{\pi}_{id} + \sum_{d\in\mathcal{D}} l_{hd}\hat{\tau}_d - \sum_{d\in\mathcal{D}}\sum_{C\in\mathcal{C}(r_d^h)}\hat{\chi}_{Cd}$ for variable ζ_h. The algorithm performs a dual adjustment operation similar to the operation performed for solving the relaxation R1 (see Step 4). The correctness of Algorithm 2 is shown by the following proposition.

Proposition 6. *Algorithm 2 solves the relaxation* R2 *to optimality. (See §EC.2.8 for the proof)*

5.1. Solving the Pricing Problem PP2

In this section, we formulate an integer program to solve the pricing problem PP2. Given $d \in \mathcal{D}$ and $r \in \bar{\mathcal{R}}_d$, let $\hat{\sigma}_{rd}$ be a parameter defined as $\hat{\sigma}_{rd} = t_{rd}^R - \sum_{i\in\mathcal{N}} a_{ri}\hat{\lambda}_{id} - \sum_{i\in\mathcal{N}_d} b_{rid}\hat{\pi}_{id} + l_{rd}\hat{\tau}_d - \sum_{C\in\mathcal{C}(r)}\hat{\chi}_{Cd}$. Let y_{rd} denote a binary variable that equals 1 if route r is operated on day d in a schedule and 0 otherwise. Let u_i denote a binary variable that equals 1 if customer i is visited at least once in a schedule and 0 otherwise. Let w_{ij} denote a binary variable that equals 1 if arc (i,j) is traversed at least once in a schedule and 0 otherwise. Given $\bar{\mathcal{R}} = \cup_{d\in\mathcal{D}}\bar{\mathcal{R}}_d$, the pricing problem PP2 can be formulated as the following IP problem:

$$z(\mathrm{PP2}) = \min \sum_{d\in\mathcal{D}}\sum_{r\in\bar{\mathcal{R}}_d} \hat{\sigma}_{rd} y_{rd} - \sum_{i\in\mathcal{N}} \hat{\beta}_i u_i - \hat{\mu} \tag{7a}$$

Algorithm 2 Column-and-Cut Generation for R2

Step 0: Initialize the RMP2;

Step 1: Solve the RMP2 to obtain its optimal primal and dual solutions;

Step 2: Solve the pricing problem PP2, that is, $\min_{h\in\hat{\mathcal{H}}\setminus\hat{\mathcal{H}}'}\{c'(\zeta_h)\}$ (see §5.1);

Step 3: If $\min_{h\in\hat{\mathcal{H}}\setminus\hat{\mathcal{H}}'}\{c'(\zeta_h)\}\geq 0$, go to Step 4; otherwise, add to RMP2 the schedule having the most negative reduced cost $c'(\zeta_h)$, and go to Step 1;

Step 4: If $\min_{r\in\bar{\mathcal{R}}_d}\{-a_{ri}\hat{\pi}_{id}\}-\hat{\tau}_d\geq 0$ for $d\in\mathcal{D}$, that is $c'(\eta_{hd})\geq 0$, go to Step 5; otherwise, set $\hat{\tau}_d=\min_{r\in\bar{\mathcal{R}}_d}\{-a_{ri}\hat{\pi}_{id}\}$ for $d\in\mathcal{D}$ (*dual adjustment*), and go to Step 2;

Step 5: If SR3 inequalities (6i) are all satisfied by the primal solution of the RMP2, the algorithm terminates; otherwise, add all violated inequalities to the RMP2, and go to Step 1.

$$\text{s.t.} \sum_{r\in\bar{\mathcal{R}}_d} y_{rd}\leq 1 \quad d\in\mathcal{D}, \tag{7b}$$

$$\sum_{r\in\bar{\mathcal{R}}_d} a_{ri}y_{rd}\leq u_i \quad i\in\mathcal{N}_d, d\in\mathcal{D}, \tag{7c}$$

$$\sum_{r\in\bar{\mathcal{R}}_d} e_{rij}y_{rd}\leq w_{ij} \quad (i,j)\in\mathcal{A}, d\in\mathcal{D}, \tag{7d}$$

$$\sum_{(i,j)\in\mathcal{A}} w_{ij}\leq \mathbf{F}\sum_{d\in\mathcal{D}}\sum_{(i,j)\in\mathcal{A}}\sum_{r\in\bar{\mathcal{R}}_d} e_{rij}y_{rd}, \tag{7e}$$

$$u_i\in\{0,1\} \quad i\in\mathcal{N}, \tag{7f}$$

$$y_{rd}\in\{0,1\} \quad r\in\bar{\mathcal{R}}_d, d\in\mathcal{D}, \tag{7g}$$

$$w_{ij}\in\{0,1\} \quad (i,j)\in\mathcal{A}. \tag{7h}$$

The objective function (7a) corresponds to the reduced cost of a schedule. Constraints (7b) allow at most one route on each day. Constraints (7c) link variables $\boldsymbol{y}$ and $\boldsymbol{u}$. Constraints (7d) and (7e) impose the RC requirement at level $\mathbf{F}$.

Solving PP2 with $\mathbf{F}=1$. PP2 has a special structure that is worth investigating when $\mathbf{F}=1$, namely, the problem is specialized by excluding RC. In this case, the pricing problem can be reformulated as follows:

$$z(\text{PP2})=\min\sum_{d\in\mathcal{D}}\sum_{r\in\bar{\mathcal{R}}_d}\hat{\sigma}_{rd}y_{rd}-\sum_{i\in\mathcal{N}}\hat{\beta}_i u_i-\hat{\mu} \tag{8a}$$

$$\text{s.t.} \sum_{r\in\bar{\mathcal{R}}_d} y_{rd}\leq 1 \quad d\in\mathcal{D}, \tag{8b}$$

$$a_{ri}y_{rd}\leq u_i \quad r\in\bar{\mathcal{R}}_d, d\in\mathcal{D}, i\in\mathcal{N}_d, \tag{8c}$$

$$u_i\in\{0,1\} \quad i\in\mathcal{N}, \tag{8d}$$

$$y_{rd}\in\{0,1\} \quad r\in\bar{\mathcal{R}}_d, d\in\mathcal{D}. \tag{8e}$$

The reformulated PP2 can now be decomposed into a master problem with decisions on $\{u_i, \forall i \in \mathcal{N}\}$ and a subproblem with decisions on $\{y_{rd}, \forall r \in \bar{\mathcal{R}}_d, d \in \mathcal{D}\}$. Note that constraints (8c) are reformulated from (7c), due to the fact that at most one route is to be selected for each day. With such reformulation, the subproblem is a binary IP model whose coefficient matrix of the constraints is *totally unimodular*, and a Benders decomposition-based approach can be used to solve PP2 to optimality. Additional details are given in the e-companion (see §EC.5).

5.2. Generating the Schedule Set $\bar{\mathcal{H}}$

At the end of CCG Algorithm 2, the optimal dual solution of R2 $\Pi_2^* = (\hat{\boldsymbol{\lambda}}^*, \hat{\boldsymbol{\mu}}^*, \hat{\boldsymbol{\beta}}^*, \hat{\boldsymbol{\pi}}^*, \hat{\boldsymbol{\tau}}^*, \hat{\boldsymbol{\iota}}^*, \hat{\boldsymbol{\kappa}}^*, \hat{\boldsymbol{\chi}}^*)$ of cost $z(\Pi_2^*)$ is transformed into an equivalent dual solution of LRF2 strengthened with the same SR3 inequalities identified by CCG Algorithm 2 (see Theorem 2).

Based on the following proposition, to generate the candidate optimal schedule set $\bar{\mathcal{H}}$, we also make use of the dual solution $\Pi_1^* = (\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*, \boldsymbol{\gamma}^*, \varepsilon^*, \boldsymbol{\pi}^*, \boldsymbol{\tau}^*, \boldsymbol{\iota}^*, \boldsymbol{\kappa}^*, \boldsymbol{\chi}^*)$ provided by the relaxation R1 to further reduce the size of the final schedule set $\bar{\mathcal{H}}$.

Proposition 7. *Let Υ_h^2 be the reduced cost of a schedule h computed with respect to solution Π_2^*, and let $\Upsilon_h^1 = \sum_{d \in \mathcal{D}} \sigma^*_{r_d^h, d}$, where r_d^h denotes the index of the route operated on day $d \in \mathcal{D}$ in schedule h, and σ^*_{rd} denotes the reduced cost of the corresponding route $r \in \bar{\mathcal{R}}_d$ computed with respect to solution Π_1^*. Then, set $\bar{\mathcal{H}}$ contains all schedules h such that $\Upsilon_h^2 \le ub - z(\Pi_2^*)$ and $\Upsilon_h^1 \le ub - z(\Pi_1^*)$. (See §EC.2.9 for the proof)*

The algorithm used to generate the schedule set $\bar{\mathcal{H}}$ relies on this proposition and is a straightforward extension of the DP algorithm mentioned in Section 5.1; additional details are omitted for the sake of brevity.

6. Numerical Experiments

This section provides the computational results of our exact method. The exact algorithm is coded in C# and runs on a PC equipped with an Intel Core i5 CPU 3.30 GHz and 16 GB of RAM. We used ILOG CPLEX 12.5 as an LP solver to tackle the RMPs of both CCGs and as an IP solver to tackle F0 and the reduced formulations F1 and F2. The computation time limit for the CCG was one hour to solve R1 and two hours to solve R2. The computation time limit for solving all formulations by the IP solver was three hours. All computation times are reported in seconds.

6.1. Benchmark Instances and Settings

Several datasets for ConVRPs are available in the literature to test heuristic algorithms, with instances including 10 to 199 customers; we mainly used small and medium datasets (≤ 50 customers) to examine our exact method. The experiments were based on the three datasets $\{S_A, S_B, S_C\}$, where S_A and S_B were taken from the literature and S_C was newly generated.

Dataset S_A was taken from Groër et al. (2009), which includes 10 instances with 10 or 12 customers. These instances are specified with $D = 3$ days, $K = 3$ drivers, route time duration $T = 30$, and vehicle capacity $Q = 15$. The average travel time on arcs ranges from 4.5 to 5.8. The probability that a customer requires service on a given day (the service frequency) is set to 70%. Pickup demands are not considered—that is, $q_{id} = 0$ for each customer i on day d. The service time for customer i on day d is set equal to 1, or $s_{id} = 1$. For the experiments based on this dataset, we set the consistency levels at $\mathbf{L} = 5$, $\mathbf{E} = 1$ and $\mathbf{F} = 1$.

Dataset S_B was taken from Goeke et al. (2019), which contains 144 instances with 20 or 30 customers and $D = 5$ days. The instances are classified into groups with service frequencies of $\{50\%, 70\%, 90\%\}$. Half of the instances in each group set different values of T (ranging from 160 to 230), and the other half have no route-duration restrictions (i.e., $T \to +\infty$). The vehicle capacity Q ranges from 140 to 200. The average travel time on arcs ranges from 25.0 to 30.1. The demands of customers range from 0 to 41. The service time s_{id} for customer i on day d is set to 10. In agreement with the results reported in Goeke et al. (2019), the experiments based on dataset S_B were carried out with $\mathbf{E} = 1$ and $\mathbf{F} = 1$; $\mathbf{L}$ was selected from $\{0.4 \cdot \mathbf{L}_{\max}, 0.6 \cdot \mathbf{L}_{\max}, 0.8 \cdot \mathbf{L}_{\max}, \mathbf{L}_{+\infty}\}$, where $\mathbf{L}_{+\infty} \to +\infty$ and $\mathbf{L}_{\max}$, ranging from 62 to 104, is the maximum arrival time difference derived when each instance is solved with $\mathbf{L}_{+\infty}$.

Dataset S_C consists of 50 new instances generated to examine our exact method in a general consistency setting. These new instances involve 20 to 50 customers whose information (i.e., locations, service times, and delivery demands) is based on the datasets of Groër et al. (2009) for the instances with 70% service frequency and the datasets of Kovacs et al. (2014b) for the instances with 50% and 90% service frequency. In each instance, we suppose that each customer also has a pickup demand, the volume of which equals the amount of the delivery demand multiplied by a random ratio in $[0.7, 1.3]$. These instances are specified with $D = 5$ days, route time duration $T = 180$, and vehicle capacity $Q = 140$. The average travel time on arcs ranges from 32.2 to 35.1. The driver number K in the instances ranges from 4 to 9. The service time for customer i on day d is set to a constant $s_{id} = 10$. Similar to the S_B dataset, each instance of S_C is specified with a service frequency of $\{50\%, 70\%, 90\%\}$. We first tested consistency levels with $\mathbf{L} = 55$, $\mathbf{E} = 3$ and $\mathbf{F} = 0.8$ to derive some computational results. Then, we tested every possible combination of the three consistency levels with some changed parameters to obtain managerial insights into opportunities for improving services for the VRPGCR-SDC.

Table 2 Computational results of dataset S_A

Instance ID	Solving F0		Our Method		t^*/t	Lower Bound Improvement				Extended case $\mathbf{E}=2$			
	ub^*	t^*	ub	t		gap_M	gap_N	gap_1	gap_2	ub_E	rdc	t_E	t_E/t
A1_3_0.7_10	142.03	1752	142.03	3	584.00	6.88%	4.91%	2.50%	1.44%	142.03	0.00%	3	1.00
A2_3_0.7_10	121.07	372	121.07	18	20.67	9.71%	4.66%	3.11%	1.75%	120.72	0.29%	14	0.78
A3_3_0.7_10	149.41	2562	149.41	22	116.45	7.59%	5.20%	2.80%	1.38%	149.13	0.19%	21	0.95
A4_3_0.7_10	150.89	1287	150.89	10	128.70	10.96%	8.11%	4.14%	1.84%	150.89	0.00%	8	0.80
A5_3_0.7_10	132.31	1056	132.31	36	29.33	10.10%	7.03%	3.75%	1.79%	127.05	4.14%	32	0.89
A6_3_0.7_12	171.02	15203	171.02	100	152.03	5.79%	4.01%	2.68%	1.14%	169.22	1.06%	75	0.75
A7_3_0.7_12	111.54	5085	111.54	280	18.16	9.39%	6.02%	2.76%	1.18%	105.28	5.95%	135	0.48
A8_3_0.7_12	145.69	8487	145.69	51	166.41	14.18%	9.83%	5.44%	2.45%	140.68	3.56%	32	0.63
A9_3_0.7_12	166.37	16621	166.37	178	93.38	10.00%	6.98%	4.94%	1.83%	162.33	2.49%	120	0.67
A10_3_0.7_12	140.42	3882	140.42	27	143.78	14.90%	9.61%	5.33%	2.48%	133.15	5.46%	23	0.85
Average					**145.29**	**9.95%**	**6.64%**	**3.74%**	**1.73%**		**2.31%**		**0.78**

All vehicles are required to start their routes at time zero in datasets S_A and S_B, and no waiting is allowed between customer visits for all three datasets.

We represent each instance by "**a_b_c_d**", where "**a**" is the instance ID, "**b**" is the number of available drivers, "**c**" is the service frequency, and "**d**" is the number of customers.

6.2. Results of Datasets S_A and S_B

We evaluate the performance of our algorithms on dataset S_A from Groër et al. (2009) and dataset S_B from Goeke et al. (2019) by setting $\mathbf{E}=1$ in our exact algorithm (also with relaxed RC $\mathbf{F}=1$). We also compare our algorithms with the results of the exact method of Goeke et al. (2019) that was run on an AMD FX-6300 processor at 3.5 GHz with 8 GB of RAM. Further, due to the fact that the existing exact algorithm is tailored for solving the case $\mathbf{E}=1$, we also conduct additional experiments on testing the extended case $\mathbf{E}=2$, to show our algorithmic advantages and the benefits on $\mathbf{E}=2$.

Results of Dataset S_A. We first compare our exact algorithm with the solution of the arc-based formulation F0 (see §EC.1) on solving the traditional ConVRP (under $\mathbf{E}=1$), investigate the lower bound improvement of reformulations, and explore benefits of the extended case $\mathbf{E}=2$.

Table 2 reports the results of dataset S_A, where each instance is associated with the following columns: the optimal solution cost computed by solving F0 using CPLEX (ub^*) and the corresponding computation time (t^*), the solution cost computed by our exact algorithm (ub) and the corresponding total computation time (t), the ratio between t^* and t (t^*/t), the percentage gap of the lower bound lb_M (gap_M) of R1 with the big-M-based constraints (1h)–(1i) and without SR3 inequalities, the percentage gap of the lower bound lb_N provided by R1 without SR3 inequalities

(gap_N), the percentage gap of the lower bound lb_1 (gap_1) provided by R1, and the percentage gap of the lower bound lb_2 provided by R2 (gap_2) (all gaps computed with respect to the optimal solution cost ub^* or our ub). In addition to the above results on solving the case $\mathbf{E} = 1$, the table also reports the optimal solution cost of $\mathbf{E} = 2$ (ub_E) on solving the same instance, the percentage of cost reduction compared with $\mathbf{E} = 1$ (rdc), the computation time spent solving $\mathbf{E} = 2$ (t_E), and the ratio of the computation time over that of $\mathbf{E} = 1$ (t_E/t).

From Table 2, it can be seen that our exact algorithm is much faster than directly solving F0. On average, solving F0 with CPLEX needs about 145 times more computation time than solving it with the exact algorithm. Table 2 shows the improved lower bound after the big-M-based constraints (1h)–(1i) have been replaced with the (2a)–(2c) TC constraints. As can be seen, lb_N is tighter than lb_M, and the average lower bound gap is improved from 9.95% to 6.64%. In the following section describing the results for dataset S_B in greater detail, we will show the impacts of different TC levels on the lower bound. Meanwhile, lower bounds lb_1 and lb_2 provided by R1 and R2 are even tighter (see gap_1 and gap_2) given the SR3 inequalities and the structure of the schedule-based formulation. All the instances of dataset S_A have also been solved to optimality by the method of Goeke et al. (2019) with an average computing time equal to 3.3 seconds.

Table 2 also reports the solutions of $\mathbf{E} = 2$. As discussed by Kovacs et al. (2015a), the most restricted DC level $\mathbf{E} = 1$ is impractical in some cases, because it is hard to request a single driver to be on duty on all days. By extending the DC level to $\mathbf{E} = 2$, significant cost savings and computational benefits are achieved. On average, the travel cost reduces by 2.31%, and at the same time the solution time reduces by 22%, since the reduced problem formulation based on optimal route/schedule sets can be solved faster at $\mathbf{E} = 2$.

Results of Dataset S_B. Goeke et al. (2019) generated dataset S_B to test the largest problem size that could be solved by their exact algorithm for solving the special case of the ConVRP.

Tables 3 and 4 provide an overview of the results of our solving the 144 instances of S_B from two perspectives. Table 3 groups by different TC levels (i.e., different values of the maximum arrival time difference), and Table 4 groups by different values of the service frequency. The TC level and the service frequency are given in the first column of Table 3 and Table 4, respectively. The columns of the two tables show the following information: the number of customers (N), the presence/absence of route-duration constraints (T = yes/no), and the number of instances solved to optimality by our algorithm (O) and by Goeke et al. (2019) (O_G) out of the 9 and 12 instances in each group in tables 3 and 4, respectively. Columns gap_G and t_G give the percentage gap of the lower bound and the computing time in seconds of the method of Goeke et al. The middle seven

Table 3 Results overview of the TC level of dataset S_B

TC level	N	T	O/O$_G$	gap_G	t_G	CCG for R1			CCG for R2			t	Extended case $\mathbf{E}=2$		
						gap_1	t_1	$\lvert\bar{\mathcal{R}}\rvert$	gap_2	t_2	$\lvert\bar{\mathcal{H}}\rvert$		O$_E$	rdc	t_E/t
$0.4\cdot\mathbf{L}_{\max}$	20	yes	9/9	0.0	164.0	4.69%	14.3	6476	1.86%	3785.5	279261	4677.7	9	2.32%	0.64
	20	no	7/5	2.2	3410.8	3.24%	41.6	6071	1.80%	3551.0	143872	4337.0	9	2.73%	0.30
	30	yes	6/8	0.4	1159.0	4.06%	110.3	11139	1.61%	6032.0	314945	8860.2	9	2.02%	0.63
	30	no	3/4	0.8	3482.1	3.94%	757.3	9035	2.03%	3832.0	166228	9313.7	8	2.25%	0.52
$0.6\cdot\mathbf{L}_{\max}$	20	yes	9/9	0.0	105.8	3.79%	13.3	6198	1.53%	3111.5	225161	3278.2	9	1.86%	0.64
	20	no	7/9	0.0	633.5	2.97%	44.6	5874	1.60%	2277.8	113548	2887.3	9	1.90%	0.40
	30	yes	6/9	0.0	290.7	3.42%	115.4	11047	1.25%	4412.3	275042	6516.2	9	1.59%	0.65
	30	no	3/7	0.0	541.0	3.86%	745.3	10295	1.72%	2799.0	154119	6285.7	8	2.23%	0.61
$0.8\cdot\mathbf{L}_{\max}$	20	yes	9/9	0.0	86.0	3.70%	15.0	6127	1.43%	2628.0	199719	2622.4	9	2.10%	0.58
	20	no	7/9	0.0	194.8	2.77%	40.9	5713	1.59%	1554.3	90684	1831.4	9	1.72%	0.40
	30	yes	6/9	0.0	266.0	3.36%	110.6	10641	1.22%	2904.3	229363	4601.0	9	1.71%	0.63
	30	no	3/7	0.0	466.0	3.68%	705.4	9044	1.63%	1610.0	110540	3578.0	8	2.12%	0.42
$\mathbf{L}_{+\infty}$	20	yes	9/9	0.0	59.4	3.66%	10.9	5835	1.33%	1671.5	189546	1224.6	9	2.56%	0.45
	20	no	7/9	0.0	81.9	2.67%	36.7	5465	1.44%	1102.0	77247	1071.4	9	1.49%	0.36
	30	yes	6/9	0.0	187.5	3.34%	65.8	10554	1.21%	2413.0	174557	2710.8	9	2.01%	0.25
	30	no	3/7	0.0	406.0	3.45%	499.5	8991	1.56%	1067.0	89627	2362.7	8	1.90%	0.26

columns show the average values of the major outputs of our algorithm for *the solved instances* in each group, including the percentage gap of the lower bound lb_1 (gap_1) provided by R1, the computation time of R1 (t_1), the size of the candidate optimal route set ($|\bar{\mathcal{R}}|$), the percentage gap of the lower bound lb_2 (gap_2) provided by R2, the computation time of R2 (t_2), the size of the candidate optimal schedule set ($|\bar{\mathcal{H}}|$), and the time used by our exact algorithm (t). The remaining columns show the results of resolving instances by setting $\mathbf{E}=2$: the number of instances solved to optimality (O$_E$), the cost reduction compared with $\mathbf{E}=1$ (rdc), and the ratio of the computation time over that of $\mathbf{E}=1$ (t_E/t). Detailed results on the solution of S_B by our algorithm are listed in the e-companion (see §EC.6), for a detailed comparison with Goeke et al. (2019).

Based on the two tables, our method solves 100 of the 144 instances of dataset S_B to optimality for the traditional ConVRP ($\mathbf{E}=1$), relative to the 128 instances solved by Goeke et al. (2019). The instances not solved by our method had intractable schedule sets given the large size of the candidate optimal route sets generated. This limitation will be explored in §6.3. With respect to the solution performance, our exact algorithm shares some features with Goeke et al. (2019). Solving instances with 30 customers (36 solved) is obviously more difficult than instances with 20 customers (64 solved). Meanwhile, instances with unlimited route-time duration restrictions are even more difficult to solve. In our algorithm, an appropriate restriction on T would limit the number of feasible routes, possibly reducing the size of the candidate optimal route set—thus making our algorithm more effective. Similarly, in Goeke et al. (2019), an appropriate restriction on T also reduces the number of possible customer clusters that can be served by each driver.

Table 4 Results overview of the service frequency of dataset S_B

Frequency	N	T	O/O$_G$	gap_G	t_G	CCG for R1			CCG for R2			t	Extended case $\mathbf{E}=2$		
						gap_1	t_1	$\|\bar{\mathcal{R}}\|$	gap_2	t_2	$\|\bar{\mathcal{H}}\|$		O$_E$	rdc	t_E/t
50%	20	yes	12/12	0.0	76.3	3.44%	4.2	1039		==>		25.3	12	1.56%	0.70
	20	no	12/11	0.6	970.5	3.01%	5.8	1143		==>		49.7	12	1.84%	0.62
	30	yes	12/12	0.0	260.0	3.64%	21.7	4321		==>		2317.4	12	2.46%	0.55
	30	no	12/7	0.1	1208.1	3.73%	43.1	9342	1.74%	2327.0	130129	5385.0	12	1.88%	0.45
70%	20	yes	12/12	0.0	108.1	4.56%	12.5	4103		==>		2396.5	12	3.23%	0.54
	20	no	12/11	0.5	793.4	3.34%	19.7	8780	1.78%	1902.3	97932	3913.8	12	2.46%	0.13
	30	yes	8/12	0.0	272.5	3.21%	62.6	16136	1.31%	3641.5	226959	7757.9	12	1.06%	0.50
	30	no	0/11	0.2	1221.0	—	—	—	—	—	—	—	12	2.86%	**
90%	20	yes	12/12	0.0	127.0	3.88%	23.5	13335	1.54%	2799.1	223422	6430.4	12	1.84%	0.50
	20	no	4/10	0.6	1476.9	1.33%	97.3	10696	1.10%	2778.0	131555	5832.0	12	1.58%	0.33
	30	yes	4/11	0.3	894.9	3.93%	217.3	19837	1.35%	4538.3	291511	11564.3	12	1.97%	0.61
	30	no	0/7	0.3	1243.6	—	—	—	—	—	—	—	8	1.39%	**

Notes: (i) "—": no feasible solution is found by CPLEX within the imposed time limit; (ii) "==>": instance solved to optimality using formulation F1. (iii) "**": no comparison with $\mathbf{E}=1$ due to the lack of results on solving $\mathbf{E}=1$.

However, in Tables 3 and 4, as well as in the detailed results, we can see solution performance differences between our method and theirs. In Table 3, when solving instances with more restricted TC levels and holding other parameters unchanged, the computation efficiency of their algorithm degenerates rapidly, especially for instances with $0.4 \cdot \mathbf{L}_{\max}$. Comparatively, our algorithm performs more stably, with only moderate increases in computation time. The reason may be that our algorithm holds the enhanced TC constraints in our major formulations R1 and R2, which maintain the TC of the route- and schedule-generation processes. However, Goeke et al. (2019) use an independent procedure, rather than their set-partitioning formulation, to adjust TC. In Table 4, we see another remarkable difference. Our algorithm performs better for instances with lower service frequency. Specifically, we solve all instances with 50% service frequency to optimality, in which six instances are not solved to optimality by Goeke et al. (2019), and 75% of those instances are solved by our method without even involving the schedule-generation process (i.e., directly solving F1 with $\bar{\mathcal{R}}$). The reason is that with a lower service frequency, the subgraph $\mathcal{G}_d$ of day d becomes smaller such that $\bar{\mathcal{R}}_d$ obtained for day d will be reduced in our algorithm, raising the computation efficiency. However, since the set-partitioning formulation of Goeke et al. (2019) relies on all of the customer clusters in the entire graph, the service frequency does not affect the number of clusters.

Apart from the above discussion showing our competitiveness on solving $\mathbf{E}=1$ instances with Goeke et al. (2019), the two tables also verify our advantages in solving the extended case $\mathbf{E}=2$. When $\mathbf{E}=2$, there is a robust operational benefit, in that the travel cost reduces by about 2%, and for some extreme cases, the cost reduction is up to 6.58%, shown in Table EC.5 in the e-companion. In addition, our algorithm is much more efficient at solving $\mathbf{E}=2$, since we can solve 140 out of 144 instances to optimality and the solution time also becomes much shorter.

Table 5 Computational results for instances of dataset S_C with 20 customers

Instance ID	CCG for R1				Solving F1		$t_1+\bar{t}_1$	Solving F0	
	lb_1	gap_1	t_1	$\lvert\bar{\mathcal{R}}\rvert$	ub_1	$\bar{t}_1$		ub_0	gap_0
C1_4_0.5_20	1266.05	0.85%	6	1171	1276.97	80	86	1512.60	18.45%
C2_4_0.5_20	1315.15	1.13%	8	822	1330.13	18	26	1692.28	27.23%
C3_4_0.5_20	1351.65	3.21%	6	1089	1396.52	97	103	—	—
C4_4_0.5_20	1249.14	1.28%	6	1109	1265.38	62	68	1503.84	18.85%
C5_4_0.5_20	1313.96	1.40%	7	987	1332.66	22	29	1537.53	15.37%
C6_4_0.5_20	1229.10	0.64%	7	973	1237.07	61	68	—	—
C7_4_0.5_20	1315.31	1.53%	6	785	1335.70	16	22	1644.97	23.15%
C8_4_0.5_20	1350.78	2.18%	8	1285	1380.91	44	52	—	—
C9_4_0.5_20	1245.24	1.11%	9	1313	1259.18	103	112	—	—
C10_4_0.5_20	1243.84	2.62%	6	850	1277.32	18	24	—	—
Average		**1.60%**	**7**	**1038**		**52**	**59**		**20.61%**
C11_5_0.7_20	2119.39	1.58%	14	3202	2153.40	1398	1412	—	—
C12_5_0.7_20	1877.57	1.55%	15	3265	1907.17	2302	2317	—	—
C13_5_0.7_20	2179.15	0.85%	15	1627	2197.91	599	614	—	—
C14_5_0.7_20	2228.45	2.08%	16	2865	2275.76	3585	3601	—	—
C15_5_0.7_20	2064.47	0.89%	14	3828	2082.97	869	883	—	—
C16_5_0.7_20	1905.06	3.55%	12	3746	1975.25	4134	4146	—	—
C17_5_0.7_20	1967.64	0.48%	19	4516	1977.19	2917	2936	—	—
C18_5_0.7_20	2095.65	0.38%	16	2782	2103.56	191	207	—	—
C19_5_0.7_20	1931.45	3.85%	20	4742	2008.87	5843	5863	—	—
C20_5_0.7_20	2089.80	1.59%	18	4029	2123.55	567	585	—	—
Average		**1.68%**	**16**	**3460**		**2241**	**2256**	—	—

Notes: "—": no feasible solution is found by CPLEX within the imposed time limit.

6.3. Results of Dataset S_C

We tested the performance of our exact method on the new instances of S_C, which were generated with a general consistency setting (i.e., $\mathbf{L}=55$, $\mathbf{E}=3$ and $\mathbf{F}=0.8$). The tests were conducted in two parts. First, we compared our exact method with the formulation F0 solved using CPLEX on a set of 20 instances, each including 20 customers. The size of the yielded candidate optimal route sets $\bar{\mathcal{R}}$ for F1 of these instances was less than 5000; thus, we directly solved the reduced problem of F1 to find an optimal solution. In the second part, we tested the solution performance based on 30 larger instances, with 25 to 50 customers. For all larger instances, the size of set $|\bar{\mathcal{R}}|$ was greater than 5000, and the exact algorithm solved the reduced problem of F2.

Table 5 shows the following columns: the lower bound value provided by R1 (lb_1) and its corresponding percentage gap (gap_1) computed with respect to the cost of the optimal solution (ub_1) found by solving F1, the computation time spent solving R1 (t_1), the size of the candidate optimal route set ($|\bar{\mathcal{R}}|$), the computation time spent solving the reduced problem of F1 ($\bar{t}_1$), the total computation time of the exact method ($t_1+\bar{t}_1$), the cost of the solution obtained by solving F0 (ub_0), and the percentage gap (gap_0) provided by CPLEX after reaching the time limit of three hours.

Table 6 **Computational results for instances of dataset S_C with 25 to 50 customers**

Instance ID	CCG for R1				CCG for R2				Solving F2		$t_1+t_2+\bar{t}_2$
	lb_1	gap_1	t_1	$\lvert\bar{\mathcal{R}}\rvert$	lb_2	gap_2	t_2	$\lvert\bar{\mathcal{H}}\rvert$	ub_2	$\bar{t}_2$	
C21_7_0.5_40	2357.37	3.34%	92	13504	2387.74	2.10%	2493	188284	2438.90	3822	6407
C22_7_0.5_40	2381.58	2.90%	184	11996	2418.10	1.41%	1587	142313	2452.80	1982	3753
C23_9_0.5_50	3151.70	3.91%	96	17442	3199.70	2.45%	2008	283563	3280.09	5279	7383
C24_9_0.5_50	2755.89	1.93%	246	26426	2778.90	1.11%	1967	374241	2810.00	6055	8268
C25_7_0.5_40	2295.12	1.86%	78	10283	2301.19	1.60%	1626	171450	2338.55	3131	4835
C26_7_0.5_40	2338.90	2.70%	66	12099	2347.85	2.33%	2255	330379	2403.77	5389	7710
C27_7_0.5_40	2391.80	2.81%	40	11196	2422.63	1.56%	2160	257913	2460.93	4884	7084
C28_9_0.5_50	3154.44	3.86%	90	17390	3205.53	2.31%	3346	217346	3281.20	5943	9379
C29_7_0.5_40	2355.22	2.19%	230	17289	2365.49	1.77%	2754	304555	2408.05	4845	7829
C30_7_0.5_40	2345.70	2.13%	57	13689	2379.45	0.72%	1483	176365	2396.77	4096	5636
Average		**2.76%**	**118**	**15131**		**1.73%**	**2168**	**244641**		**4543**	**6828**
C31_7_0.7_35	2990.18	1.96%	146	24082	3011.00	1.28%	2780	353433	3050.00	6301	9227
C32_6_0.7_30	2704.06	2.62%	45	13384	2732.21	1.61%	1676	195495	2776.81	3221	4942
C33_8_0.7_40	3561.74	2.20%	266	23837	3581.34	1.66%	2326	295326	3641.74	5650	8242
C34_7_0.7_35	3233.33	3.58%	154	26223	3256.00	2.91%	3408	372371	3353.42	5253	8815
C35_7_0.7_35	3245.02	2.14%	104	19667	3277.38	1.16%	2554	233082	3316.00	6198	8856
C36_6_0.7_30	2510.93	2.94%	84	17043	2534.10	2.05%	2487	285631	2587.02	3770	6341
C37_6_0.7_30	2737.92	3.15%	310	23092	2778.72	1.71%	2872	242637	2827.01	4736	7918
C38_6_0.7_30	2711.05	1.97%	84	13141	2741.44	0.87%	1554	131989	2765.48	2376	4014
C39_8_0.7_40	3466.72	2.17%	243	24697	3502.53	1.16%	2195	356195	3543.75	8659	11097
C40_8_0.7_40	3383.66	2.87%	325	27499	3432.44	1.47%	1877	294993	3483.66	7732	9934
Average		**2.56%**	**176**	**21267**		**1.59%**	**2373**	**276115**		**5390**	**7939**
C41_6_0.9_25	2563.36	2.34%	101	23171	2593.50	1.19%	2784	171506	2624.69	4298	7183
C42_6_0.9_25	2727.51	1.09%	44	14461	2737.36	0.73%	1715	129515	2757.52	2030	3789
C43_6_0.9_25	2825.60	2.69%	48	12736	2860.09	1.50%	1979	193541	2903.65	2419	4446
C44_6_0.9_25	2559.05	1.98%	112	25023	2578.90	1.22%	2591	264241	2610.81	5216	7919
C45_6_0.9_25	2755.44	2.37%	64	18967	2800.92	0.75%	3026	143486	2822.22	4199	7289
C46_6_0.9_25	2498.48	2.35%	67	21371	2515.82	1.67%	2358	189286	2558.48	3913	6338
C47_6_0.9_25	2722.80	2.50%	56	13351	2758.79	1.21%	2460	178338	2792.58	4266	6782
C48_6_0.9_25	2807.08	3.30%	44	15380	2845.06	1.99%	3360	255946	2902.89	5037	8441
C49_6_0.9_25	2568.18	1.91%	116	28340	2589.05	1.11%	3465	328150	2618.18	6951	10532
C50_6_0.9_25	2652.24	2.16%	57	13797	2656.02	2.02%	1985	197209	2710.75	2845	4887
Average		**2.27%**	**71**	**18660**		**1.34%**	**2572**	**205122**		**4117**	**6761**

The first 10 instances in Table 5 have their service frequency fixed at 50%. The candidate optimal route set for each instance contains 1038 routes on average. Given the limited size of $\bar{\mathcal{R}}$, the reduced problem of F1 could be solved very quickly (52 seconds on average) to optimality. The total time used by the exact method was near 1 minute on average, including the computation time for solving R1. When solving F0 for these instances, however, CPLEX could only find feasible solutions for 5 of the 10 instances within the time limit, and the gap from each obtained solution to the optimal value was as large as 20.61% on average. In comparison, CPLEX failed to find any feasible solution for the second ten instances (with a service frequency of 70%) because of the larger number of customers involved in the route planning on each day. This increase resulted in a larger $\bar{\mathcal{R}}$ (3460 routes on average) such that solving the reduced problem of F1 took more time (2241 seconds on average). Compared with the optimal solutions reported in ub_1, the lower bound gaps derived by the CCG based on R1 (see lb_1) were as tight as approximately 1.6–1.7% on average.

Table 6 displays the results of the 30 instances of S_C, in which the number of customers ranges from 25 to 50. For these instances, CPLEX failed to find any feasible solution. The first five columns of Table 6 report the same information as those in Table 5. The next four columns report information on R2, including the lower bound provided by R2 (lb_2) and its corresponding percentage gap (gap_2) computed with respect to the cost of the solution obtained by solving F2 (ub_2), the computation time (t_2) of lb_2, and the number of generated candidate optimal schedules ($|\bar{\mathcal{H}}|$). The last two columns ($\bar{t}_2$) and ($t_1 + t_2 + \bar{t}_2$) give the time spent solving F2 and the total computation time of the exact method, respectively.

Table 6 shows that the proposed exact algorithm can solve medium-size problems to optimality. The maximum problem size that can be solved is determined mainly by the size of the set $\bar{\mathcal{R}}$. Our exact algorithm has a bottleneck when $|\bar{\mathcal{R}}|$ reaches $30,000$, which will lead to an intractable size of $\bar{\mathcal{H}}$. The results in this table also verify Proposition 2 and show that lb_2 from R2 is tighter than lb_1 from R1. More specifically, the lower bound gap is reduced from 2.76% to 1.73%, from 2.56% to 1.59% and from 2.27% to 1.34% based on instances with service frequencies from 50% to 90%. When we solve S_B for the special case with $\mathbf{E} = 1$, the lower bound gap is reduced from 3.5% to 1.5%. These results indicate that for the VRPGCR-SDC, the lower bound of R2 is more stable than the lower bound of R1. Relaxation R1 cannot capture DC; thus, with a more restricted DC requirement, R1 becomes a looser lower bound, leading to a larger-size optimal route set.

7. Sensitivity Analysis and Managerial Findings

In this section, we conduct sensitivity analysis to see the impact of different consistency settings on the obtained solutions. With the obtained results, we derive some managerial insights by studying the trade-offs between the consistency requirements.

The experiments were based on the first 20 instances of S_C (10 instances with 50% service frequency and 10 instances with 70% service frequency), each of which involves 20 customers and a planning horizon of 5 days. By relaxing TC (i.e., $\mathbf{L} \to +\infty$), we first investigated the impact of different DC and RC levels on the obtained solutions. Given certain DC and RC levels, we then looked into the impact of different TC levels. Lastly, we tested the impacts of the considered consistency requirements on route familiarity.

7.1. Sensitivity Analysis on DC and RC Levels

Given a planning horizon of 5 days, we differentiated the DC and RC levels by the values taken from $\{1,2,3,4,5\}$ and $\{0.6,0.7,0.8,0.9,1.0\}$, respectively,[1] generating a total of 25 ($\mathbf{E}$, $\mathbf{F}$) combinations. For each combination, the number of available drivers K was initialized to guarantee the existence of feasible solutions. Tables 7 and 8 report the computational results under different ($\mathbf{E}$, $\mathbf{F}$) combinations, which are obtained by solving the two groups of instances with service frequencies of 50% and 70%, respectively. In the tables, the upper left part (Driver Number) shows the number of drivers initialized for the test of each combination. The upper right part (Total Cost) reports the average total travel cost over the 10 instances under different combinations. The lower left part (Average Cost) indicates the average travel cost per driver over the planning horizon (i.e., Total Cost divided by Driver Number). The lower right part (Cost Ratio) defines a benchmark setting (with $\mathbf{E}=5$ and $\mathbf{F}=1$) and reports the ratio of the results under different combinations with respect to that under the benchmark setting.

Table 7 Results based on different DC and RC levels with 50% service frequency

Driver Number	$\mathbf{F}=0.6$	$\mathbf{F}=0.7$	$\mathbf{F}=0.8$	$\mathbf{F}=0.9$	$\mathbf{F}=1.0$	Total Cost	$\mathbf{F}=0.6$	$\mathbf{F}=0.7$	$\mathbf{F}=0.8$	$\mathbf{F}=0.9$	$\mathbf{F}=1.0$
$\mathbf{E}=1$	4	4	4	3	3	$\mathbf{E}=1$	1523.09	1378.92	1359.91	1353.80	1344.54
$\mathbf{E}=2$	4	4	4	3	3	$\mathbf{E}=2$	1496.91	1346.50	1307.07	1305.01	1300.16
$\mathbf{E}=3$	4	4	4	3	3	$\mathbf{E}=3$	1496.91	1346.50	1306.36	1305.01	1300.16
$\mathbf{E}=4$	4	4	4	3	3	$\mathbf{E}=4$	1496.91	1346.50	1306.02	1305.01	1300.16
$\mathbf{E}=5$	4	4	4	3	3	$\mathbf{E}=5$	1496.91	1346.42	1306.02	1305.01	1300.16
Average Cost	$\mathbf{F}=0.6$	$\mathbf{F}=0.7$	$\mathbf{F}=0.8$	$\mathbf{F}=0.9$	$\mathbf{F}=1.0$	**Cost Ratio**	$\mathbf{F}=0.6$	$\mathbf{F}=0.7$	$\mathbf{F}=0.8$	$\mathbf{F}=0.9$	$\mathbf{F}=1.0$
$\mathbf{E}=1$	380.77	344.73	339.98	451.27	448.18	$\mathbf{E}=1$	117.15%	106.06%	104.60%	104.13%	103.41%
$\mathbf{E}=2$	374.23	336.62	326.77	435.00	433.39	$\mathbf{E}=2$	115.13%	103.56%	100.53%	100.37%	100.00%
$\mathbf{E}=3$	374.23	336.62	326.59	435.00	433.39	$\mathbf{E}=3$	115.13%	103.56%	100.48%	100.37%	100.00%
$\mathbf{E}=4$	374.23	336.62	326.50	435.00	433.39	$\mathbf{E}=4$	115.13%	103.56%	100.45%	100.37%	100.00%
$\mathbf{E}=5$	374.23	336.60	326.50	435.00	433.39	$\mathbf{E}=5$	115.13%	103.56%	100.45%	100.37%	100.00%

Notes: (i) The baseline for "Cost Ratio" is the result of the combination of $\mathbf{E}=5$ and $\mathbf{F}=1.0$. (ii) The "Average Cost" is equal to "Total Cost" divided by "Driver Number".

The benchmark setting implies a special case without considering any consistency requirements. The total travel costs for this special case remain the same for some ($\mathbf{E}$, $\mathbf{F}$) combinations with $\mathbf{F}=1$ (e.g., travel costs are equal to 1300.16 in Table 7 and are equal to 2186.51 in Table 8). In addition, maintaining the consistency requirements at reasonably high levels is not costly. For example, supposing that a 1% increase in total travel cost is acceptable, 12 and 15 of 25 combinations are acceptable for 50% and 70% service frequencies, respectively. However, when a higher DC level

[1] Note that we do not test the highest RC level 0.2 (1/5), because it will be infeasible for our instances. Since our instances have 50% and 70% service frequencies, some customers do not require services on all five days, and thus it is impossible to operate the same route on all days for each driver (corresponding to the highest RC level 0.2).

Table 8 Results based on different DC and RC levels with 70% service frequency

Driver Number	$\mathbf{F}=0.6$	$\mathbf{F}=0.7$	$\mathbf{F}=0.8$	$\mathbf{F}=0.9$	$\mathbf{F}=1.0$	Total Cost	$\mathbf{F}=0.6$	$\mathbf{F}=0.7$	$\mathbf{F}=0.8$	$\mathbf{F}=0.9$	$\mathbf{F}=1.0$
$\mathbf{E}=1$	4	4	4	4	4	$\mathbf{E}=1$	2304.16	2281.70	2258.37	2246.89	2235.48
$\mathbf{E}=2$	4	4	4	4	4	$\mathbf{E}=2$	2226.92	2210.02	2197.48	2193.94	2186.51
$\mathbf{E}=3$	4	4	4	4	4	$\mathbf{E}=3$	2226.43	2207.22	2197.48	2193.93	2186.51
$\mathbf{E}=4$	4	4	4	4	4	$\mathbf{E}=4$	2225.58	2205.22	2197.48	2193.93	2186.51
$\mathbf{E}=5$	4	4	4	4	4	$\mathbf{E}=5$	2225.58	2205.22	2197.48	2193.93	2186.51
Average Cost	$\mathbf{F}=0.6$	$\mathbf{F}=0.7$	$\mathbf{F}=0.8$	$\mathbf{F}=0.9$	$\mathbf{F}=1.0$	**Cost Ratio**	$\mathbf{F}=0.6$	$\mathbf{F}=0.7$	$\mathbf{F}=0.8$	$\mathbf{F}=0.9$	$\mathbf{F}=1.0$
$\mathbf{E}=1$	576.04	570.43	564.59	561.72	558.87	$\mathbf{E}=1$	105.38%	104.35%	103.29%	102.76%	102.24%
$\mathbf{E}=2$	556.73	552.51	549.37	548.48	546.63	$\mathbf{E}=2$	101.85%	101.08%	100.50%	100.34%	100.00%
$\mathbf{E}=3$	556.61	551.81	549.37	548.48	546.63	$\mathbf{E}=3$	101.83%	100.95%	100.50%	100.34%	100.00%
$\mathbf{E}=4$	556.40	551.31	549.37	548.48	546.63	$\mathbf{E}=4$	101.79%	100.86%	100.50%	100.34%	100.00%
$\mathbf{E}=5$	556.40	551.31	549.37	548.48	546.63	$\mathbf{E}=5$	101.79%	100.86%	100.50%	100.34%	100.00%

Notes: (i) The baseline for "Cost Ratio" is the result of the combination of $\mathbf{E}=5$ and $\mathbf{F}=1.0$. (ii) The "Average Cost" is equal to "Total Cost" divided by "Driver Number".

(i.e., $\mathbf{E}=1$) or a higher RC level (i.e., $\mathbf{F}=0.6$) is required, it is not surprising that a significantly higher cost is incurred. In essence, $\mathbf{E}=1$ implies that each driver must serve a disjoint subset of customers over the planning horizon, and $\mathbf{F}=0.6$ imposes traversing more common arcs of routes over the planning days. Both consistency requirements that have very high levels substantially limit the availability of cost-efficient routes, and thus are not suggested in practice. At the same time, imposing high RC levels sometimes requires more drivers to obtain feasible schedules, which decreases the utilization of the vehicle fleet (as seen from the average cost per driver in Figure 7).

By comparing the cost ratio results in Tables 7 and 8, we find that with different service frequency scenarios (i.e., 50% and 70%), the cost increase due to the DC and RC requirements under the 70% service frequency scenario is obviously lower than the 50% service frequency scenario. The reasoning behind this observation is that with a higher service frequency, customers are more likely to request a service every day, and thus drivers are more likely to traverse common arcs from day to day. This suggests that if a company serves a group of customers with a high service frequency, a high RC level can be targeted without incurring a high route cost.

7.2. Sensitivity Analysis on TC Levels

The sensitivity to TC levels is analyzed under five representative combinations of DC and RC: (i) $\mathbf{E}=5,\mathbf{F}=1$, the relaxation of both DC and RC (i.e., the sensitivity to TC without considering DC and RC); (ii) $\mathbf{E}=3,\mathbf{F}=0.8$, middle levels of DC and RC; (iii) $\mathbf{E}=1,\mathbf{F}=0.6$, high levels of DC and RC; (iv) $\mathbf{E}=1,\mathbf{F}=1$, a high level of DC and the relaxation of RC; and (v) $\mathbf{E}=5,\mathbf{F}=0.6$, the relaxation of DC and a high level of RC. For each combination, we increase the maximum arrival time difference $\mathbf{L}$ from 20 to 70 in increments of 5 and test the two groups of instances with service frequencies of 50% and 70% for the average total cost. Because the average total cost of each combination of DC and RC levels in Tables 7 and 8 is the cost without considering TC, we use that as the benchmark. We then calculate for each combination the comparative ratio of the average total cost for different $\mathbf{L}$ to investigate how the cost varies.

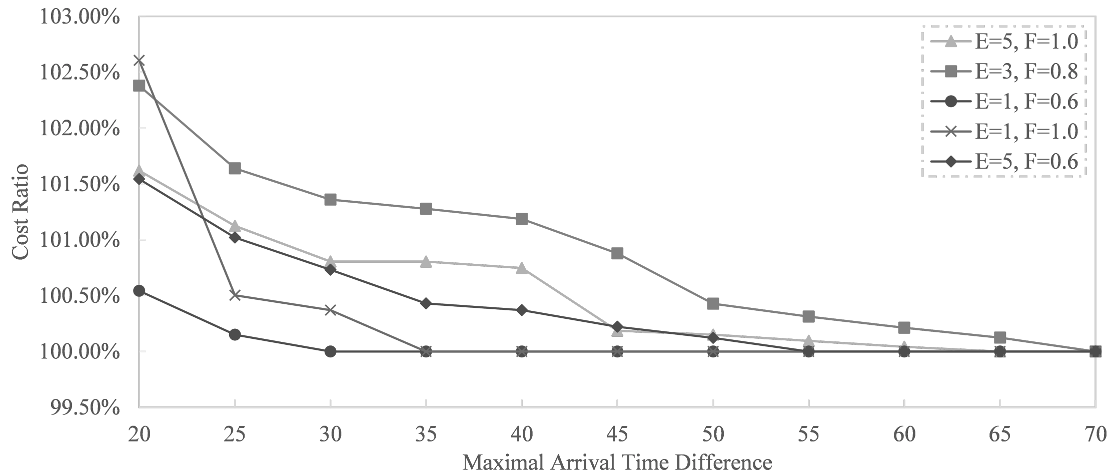

Figure 2 Sensitivity Analysis of TC for 50% Service Frequency.

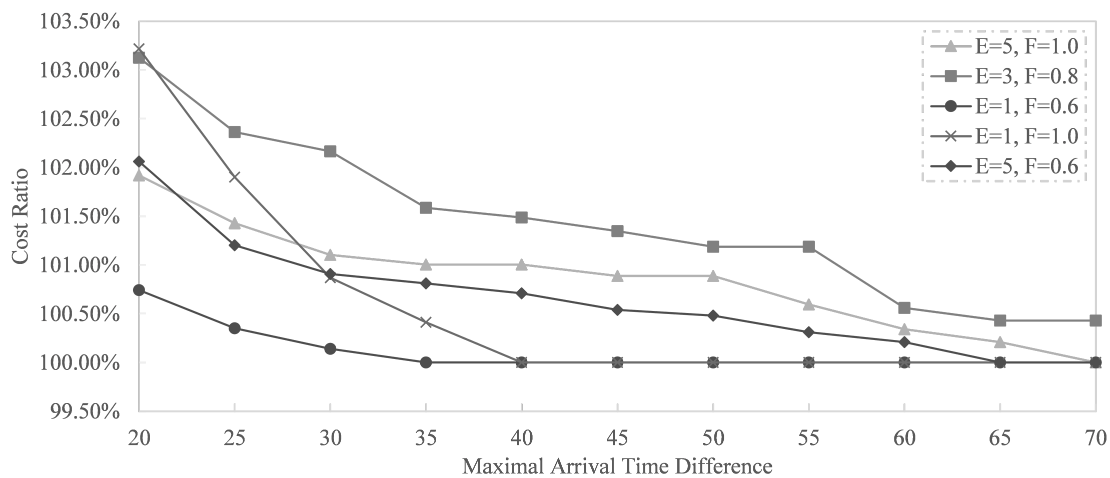

Figure 3 Sensitivity Analysis of TC for 70% Service Frequency.

Figures 2 and 3 depict the variations of the total travel cost in response to changes in the maximum arrival time difference for 50% service frequency and 70% service frequency, respectively. The sensitivity to TC generally presents a similar picture for different combinations of DC and RC levels. In both figures, the total travel cost has the highest sensitivity to TC at the middle levels of both DC and RC (i.e., $\mathbf{E}=3, \mathbf{F}=0.8$), and has the lowest sensitivity to TC at the high levels of both DC and RC (i.e., $\mathbf{E}=1, \mathbf{F}=0.6$).

Starting from $\mathbf{E} = 5, \mathbf{F} = 1$, as DC increases to the highest level (for the combination of $\mathbf{E} = 1, \mathbf{F} = 1$), certain high levels of TC will be awarded; the total travel cost will not be affected if $\mathbf{L} \geq 35$ (resp., $\mathbf{L} \geq 40$) for the 50% service frequency (resp., the 70% service frequency) for the case of $\mathbf{E} = 1, \mathbf{F} = 1$. This may reflect the effectiveness of using the principle of the ConRTR heuristic (developed by Groër et al. (2009)) to address TC; that is, keeping the highest level of DC will result in a small arrival time difference. However, it is important to note that although certain high levels of TC will be granted for the combination of $\mathbf{E} = 1, \mathbf{F} = 1$, if a very high level of TC is targeted, the total travel cost will further increase, as shown by $\mathbf{L} = 20$ in both figures. This phenomenon may be attributed to the fact that the highest level of DC indicates the special solution structure that each driver will serve a disjoint subset of customers, and the operated route of the driver on each day only traverses customers from the subset that requires service on that day. Thus, when considering TC for each customer, we only focus on the routes operated by the same driver who serves the customer. Because these operated routes are derived from the same subset of customers, certain high levels of TC can easily be granted. However, when pursuing a very high level of TC, we may seek another way to divide customers into K disjoint subsets of customers, which may perhaps raise the total travel cost.

Above $\mathbf{E} = 1, \mathbf{F} = 1$, if RC reaches to the level of 0.6 for the combination of $\mathbf{E} = 1, \mathbf{F} = 0.6$, the total travel cost will be further less affected by the TC requirement. This is due to the fact that a high RC level requires more common arcs to be traversed over several days, and thus similar customer visiting sequences are more likely to be kept over days, which benefits the TC requirement. The same phenomenon can be observed from $\mathbf{E} = 5, \mathbf{F} = 1$ to $\mathbf{E} = 5, \mathbf{F} = 0.6$ that imposes the RC level of 0.6 on the combination of both relaxed RC and DC.

7.3. Route Familiarity under Different Consistency Settings

Traditional TC and DC aim to improve customer satisfaction by enforcing more requirements on the delivery and pickup by drivers. However, the concerns of drivers are ignored. We propose the new RC to take the convenience of drivers into consideration. To verify the impacts of RC, we report in this section the average route difference ratio among K drivers (i.e., the average proportion of different arcs traversed over total arcs traversed among K drivers) under different consistency requirements settings. We conduct related tests for groups of both 50% service frequency and 70% service frequency, and record average values for comparison.

As can be seen in Figure 4, if we only impose the highest DC level $\mathbf{E} = 1$, it does not benefit route familiarity since it leads to a high average route difference ratio. The major reason for this may be that under $\mathbf{E} = 1$, customers are divided into disjoint subsets for independent routing, which makes the overall routing less friendly to drivers. If we start to impose a high TC level (e.g., $\mathbf{L} = 20$), the situation becomes slightly better, as can be seen from the bars $\mathbf{L} = 20$, and $\mathbf{E} = 1, \mathbf{L} = 20$ in Figure 4, where the latter imposes both high DC and TC levels. However, these results still show that both DC and TC in the traditional ConVRP cannot essentially benefit route familiarity. In Figure 4, if we enforce RC level $\mathbf{F} = 0.6$, we will have a low average route difference ratio of less than 0.6, suggesting high familiarity with routes, and the benefit still holds if we also set $\mathbf{E} = 1$ and $\mathbf{L} = 20$. This strongly motivates the use of RC to improve driver's route familiarity. Another point to notice in Figure 4 is that, 70% service frequency will lead to a lower average route difference ratio compared with 50% service frequency, which is consistent with the conclusion in Section 7.1 that a higher service frequency benefits the RC.

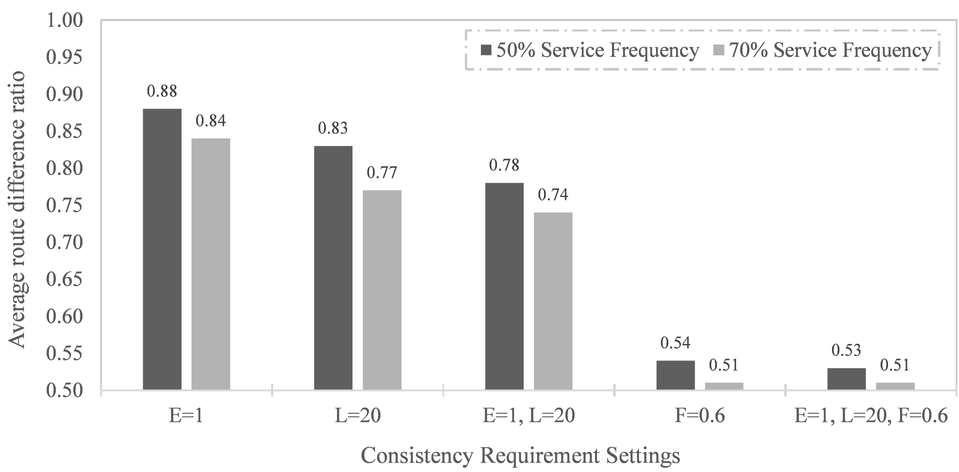

Figure 4 Average route difference ratio under different consistency requirement settings.

8. Conclusions

We studied a general multi-day vehicle routing problem that jointly considers time consistency (TC), driver consistency (DC), and route consistency (RC) requirements and involves simultaneous distribution and collection operations. For this problem, we developed two set-partitioning-based models (one for routes and one for schedules). We investigated lower bounds on the two models by solving their relaxations through column-and-cut generation techniques, and we designed an exact method for this problem that takes advantage of both models.

Numerical experiments were conducted by using both benchmark instances taken from the literature and newly generated instances with general settings of consistency requirements. The results demonstrate the effectiveness of the proposed method. Specifically, we solve instances to optimality up to the same size of the state-of-the-art method for the traditional consistent vehicle routing problem, and we solve instances with up to 50 customers with a general consistency setting. Moreover, a general setting of the driver consistency level leads to a significant cost saving. We also investigated the trade-offs between the three consistency requirements and explored some managerial insights.

A limitation of this work refers to the lack of flexibility of allowing vehicles to wait at customers, though its benefits to ConVRP solutions are known to be marginal (e.g., see Goeke et al. 2019). However, allowing vehicle waiting is indeed common and important for vehicle routing problems that consider endogenous hard time windows. To tackle vehicle waiting, our current solution framework can be extended by introducing an additional time dimension in defining the routing decisions. By doing this, the route generation pricing problem will need to consider the waiting time decision at the customer, whose solution can be much more complex (e.g., see Spliet and Gabor 2015). Furthermore, our solution framework follows a column (i.e., the routes and schedules) enumeration scheme, so working on the extended model will definitely generate many more routes and schedules with slightly different waiting time decisions at customers, thus significantly challenging the solution complexity of the problem. Hence, how to embed vehicle waiting into the problem, and its exact solution will be a future technical work to conduct.

In addition, there are some future research directions that are also worthy of further investigations. First, a tailored heuristic method could be studied to solve our problem at larger scales. Second, alternative ways could be explored to to model the route consistency and compare the resulting solutions. Third, there is also the motivation to incorporate more schedule-wise constraints (i.e., workload balance among drivers), by taking advantage of the solution techniques used in this work.

Acknowledgments

The authors thank two anonymous referees, the Associate Editor and the Editor for a thorough review and their invaluable comments on this paper. The first two authors contributed equally to this work and should be considered co-first authors. Jun Xia is the corresponding author.

References

Avci M, Topaloglu S (2016) A hybrid metaheuristic algorithm for heterogeneous vehicle routing problem with simultaneous pickup and delivery. *Expert Systems with Applications* 53:160–171.

Baldacci R, Christofides N, Mingozzi A (2008) An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts. *Mathematical Programming* 115(2):351–385.

Dell'Amico M, Righini G, Salani M (2006) A branch-and-price approach to the vehicle routing problem with simultaneous distribution and collection. *Transportation Science* 40(2):235–247.

Goeke D, Roberti R, Schneider M (2019) Exact and heuristic solution of the consistent vehicle-routing problem. *Transportation Science* 53(4):1023–1042.

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

Holland C, Levis J, Nuggehalli R, Santilli B, Winters J (2017) UPS optimizes delivery routes. *Interfaces* 47(1):8–23.

Jepsen M, Petersen B, Spoorendonk S, Pisinger D (2008) Subset-row inequalities applied to the vehicle-routing problem with time windows. *Operations Research* 56(2):497–511.

Kovacs AA, Golden BL, Hartl RF, Parragh SN (2014a) Vehicle routing problems in which consistency considerations are important: A survey. *Networks* 64(3):192–213.

Kovacs AA, Golden BL, Hartl RF, Parragh SN (2015a) The generalized consistent vehicle routing problem. *Transportation Science* 49(4):796–816.

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

Kovacs AA, Parragh SN, Hartl RF (2015b) The multi-objective generalized consistent vehicle routing problem. *European Journal of Operational Research* 247(2):441–458.

Lian K, Milburn AB, Rardin RL (2016) An improved multi-directional local search algorithm for the multi-objective consistent vehicle routing problem. *IIE Transactions* 48(10):975–992.

Liu R, Xie X, Augusto V, Rodriguez C (2013) Heuristic algorithms for a vehicle routing problem with simultaneous delivery and pickup and time windows in home health care. *European Journal of Operational Research* 230(3):475–486.

Mingozzi A, Roberti R, Toth P (2013) An exact algorithm for the multitrip vehicle routing problem. *INFORMS Journal on Computing* 25(2):193–207.

Righini G, Salani M (2006) Symmetry helps: Bounded bi-directional dynamic programming for the elementary shortest path problem with resource constraints. *Discrete Optimization* 3(3):255–273.

Smilowitz K, Nowak M, Jiang T (2013) Workforce management in periodic delivery operations. *Transportation Science* 47(2):214–230.

Spliet R, Desaulniers G (2015) The discrete time window assignment vehicle routing problem. *European Journal of Operational Research* 244(2):379–391.

Spliet R, Gabor AF (2015) The time window assignment vehicle routing problem. *Transportation Science* 49(4):721–731.

Subramanian A, Drummond LMA, Bentes C, Ochi LS, Farias R (2010) A parallel heuristic for the vehicle routing problem with simultaneous pickup and delivery. *Computers & Operations Research* 37(11):1899–1911.

Subramanyam A, Gounaris CE (2016) A branch-and-cut framework for the consistent traveling salesman problem. *European Journal of Operational Research* 248(2):384–395.

Subramanyam A, Gounaris CE (2018) A decomposition algorithm for the consistent traveling salesman problem with vehicle idling. *Transportation Science* 52(2):386–401.

Subramanyam A, Wang A, Gounaris CE (2018) A scenario decomposition algorithm for strategic time window assignment vehicle routing problems. *Transportation Research Part B: Methodological* 117:296–317.

Tarantilis CD, Stavropoulou F, Repoussis PP (2012) A template-based tabu search algorithm for the consistent vehicle routing problem. *Expert Systems with Applications* 39(4):4233–4239.

UPS (2019) Returns management & reverse logistics: UPS returns exchange Accessed September 11, 2019. https://www.ups.com/us/en/services/returns/exchange.page.