A Heuristic Algorithm for solving a large-scale real-world territory design problem

Highlights

- We consider a territory design problem in the dairy industry.
- We design an efficient, iterated hybrid genetic algorithm.
- We present extensive computational results involving more than 1000 customers.
- We explore managerial findings by studying trade-offs between service requirements.

A Heuristic Algorithm for Solving a Large-scale Real-world Territory Design Problem

Lin Zhou[a], Lu Zhen[b*], Roberto Baldacci[c], Marco Boschetti[d], Ying Dai[a], Andrew Lim[e]

[a] *College of Management, Chongqing University of Technology, Chongqing, China*

[b] *School of Management, Shanghai University, Shanghai, China*

[c] *Department of Electrical, Electronic, and Information Engineering "Guglielmo Marconi", University of Bologna, Cesena, Italy*

[d] *Department of Mathematics, University of Bologna, Rimini, Italy*

[e] *Department of Industrial Systems Engineering and Management, National University of Singapore, Singapore*

* *Corresponding author. Email: lzhen@shu.edu.cn. Tel: +86-21-66137925. Fax: +86-21-66134284*

Highlights

A Heuristic Algorithm for Solving a Large-scale Real-world Territory Design Problem

Lin Zhou, Lu Zhen, Roberto Baldacci, Marco Boschetti, Ying Dai, Andrew Lim

- We consider a territory design problem in the dairy industry.
- We design an efficient, iterated hybrid genetic algorithm.
- We present extensive computational results involving more than 1000 customers.
- We explore managerial findings by studying trade-offs between service requirements.

A Heuristic Algorithm for Solving a Large-scale Real-world Territory Design Problem

Lin Zhou[a], Lu Zhen[b,*], Roberto Baldacci[c], Marco Boschetti[d], Ying Dai[a], Andrew Lim[e]

[a]*College of Management, Chongqing University of Technology, Chongqing, China*
[b]*School of Management, Shanghai University, Shanghai, China*
[c]*Department of Electrical, Electronic, and Information Engineering "Guglielmo Marconi", University of Bologna, Cesena, Italy*
[d]*Department of Mathematics, University of Bologna, Rimini, Italy*
[e]*Department of Industrial Systems Engineering and Management, National University of Singapore, Singapore*

Abstract

In this work, we present and evaluate heuristic techniques for a real-world territory design problem of a major dairy company which produces and distributes perishable products. The problem calls for grouping customers into geographic districts, with the objective of minimising the total operational cost, computed as a function of the fixed costs of the districts and the routing costs. Two inter-connected decision levels have to be tackled: partitioning customers into districts and routing vehicles according to complex operational constraints. To solve the problem, a hybrid multi-population genetic algorithm is designed, enhanced with several evolution and search techniques. The proposed design is extensively tested on instances derived from the literature and on real-world large-scale instances, involving more than 1000 customers. The results show the effectiveness of the different components of the algorithm and the feedback from the company's planners confirms that it produces high-quality, operational solutions. Additionally, we explore some managerial findings with respect to the adoption of alternative objectives and service requirements.

*Corresponding author

Email addresses: zhoulin@cqut.edu.cn (Lin Zhou), lzhen@shu.edu.cn (Lu Zhen), r.baldacci@unibo.it (Roberto Baldacci), marco.boschetti@unibo.it (Marco Boschetti), daiying@cqut.edu.cn (Ying Dai), isealim@nus.edu.sg (Andrew Lim)

Keywords: territory design, periodic vehicle routing, hybrid genetic algorithm, dairy industry, real-world instances.

1. Introduction

In many real-world applications of retails goods distribution, customers are not served on a daily basis; instead, the demand is characterised by some other periodicity [1, 2]. The delivery patterns must be chosen over a suitable planning horizon in order to achieve the desired service level and cost benefit. This specific vehicle routing problem (VRP), known in the literature as the periodic vehicle routing problem (PVRP), may lead to complex vehicle dispatching and routing problems, both for the operators and the drivers—due to the constraints of distinct visit frequencies and available visit patterns in the planning horizon, especially for large-scale distribution management.

In order to simplify the daily problem and increase the drivers' familiarity with particular districts or areas, companies assign each driver to a fixed set of customers in the medium-long term (e.g., a semester or a season) [3, 4]. With increased familiarity, drivers become acquainted with their territories, which helps them serve their customers more effectively while enhancing service consistency [5]. At the same time, the operational and administrative work for operators is simplified. Increasing service consistency is becoming more important for transportation providers because, as competition increases, customer satisfaction becomes crucial.

To achieve these benefits, two inter-connected decision levels have to be tackled: partitioning customers into districts (also called areas or territories) [6] and routing vehicles according to complex operational constraints [7]. The process of partitioning the service area into sectors is referred to as *districting*, and problems studying the design of efficient territories are classified as districting/territory design problems [8] or territory planning problems [3]. In addition to logistics and transportation, some other important applications of territory design are political districts, sales territories, school locations, emergency sites, solid waste collection, and police patrol zones [8, 9].

In this paper, we address a new territory design problem based on the distribution problem of a major dairy company: the firm needs to rationalise its distribution network to greatly improve its competitive advantage. This rationalisation process aims to achieve remarkable savings while preserving responsiveness and service quality, requiring the joint optimisation of the distribution districts and the periodic vehicle routes.

1.1. Our contributions

Given the inherent computational complexity of the problem, both in terms of the associated optimisation problem and the size of the instances considered, we decided to solve the problem using a hybrid multi-population genetic algorithm. This choice is motivated by the fact that the literature clearly shows that existing exact algorithms for vehicle routing problems related to our problem can only solve simplified versions of our problem to optimality, and even these versions involve very small instances with only a few dozen customers. In contrast, our computational study considers instances with more than 1000 customers.

In the literature, territory design problems are generally addressed with solution approaches that favor the contiguity and compactness of the designed districts; as a result, the districts have almost round shapes [10, 9]. These approaches are not capable of considering the full complexity of the routing constraints, since they do not explicitly consider the routing component of the problem. This limitation may strongly affect the quality of the real-world solutions produced. In fact, compactness may be counterproductive in routing applications, especially in the presence of a periodic structure—and having partially overlapping territories is not necessarily more costly. Therefore we designed an integrated routing approach that benefits from of two intertwined decision levels: partitioning customers into areas and routing vehicles.

To prove the effectiveness of the proposed algorithm and its components, we perform extensive computational experiments on instances derived from the literature and the real world. Furthermore, we explore managerial findings by studying the trade-offs between the service requirements.

The paper is organised as follows. The next section presents the background of our study, states the problem, and reviews the relevant literature. The hybrid genetic algorithm is detailed in Section 3. A computational study is reported and analysed in Section 4. Finally, we give insights about the algorithm usage and its benefits, and indicate future work directions in Section 5.

2. Motivation, problem definition, and related works

The tactical problem addressed in this paper is a real application from a major dairy company whose core business is the production and distribution of perishable products (fresh milk, cheese, and butter, to name a few).

Recently, the competitive dynamics of the dairy industry have changed drastically: to be competitive, it is no longer enough to provide a high-quality product. The most important product feature has suddenly become price. As a result, the company has been forced to quickly reduce prices, to be competitive with the other companies. Continuous price reductions, however, are never sustainable if they are not supported by significant cost cutting. Of all the possible cost curbing measures, the only feasible one seems to be a reduction of the distribution costs. In fact, at the moment, the other major costs—namely, raw materials, machineries and plants, and labor—cannot be significantly lowered.

The production and distribution processes of the company are highly complex, relying on a manufacturing structure and a distribution network made up of several facilities scattered evenly across a large territory, which make up a smoothly functioning supply network.

The distribution network is organised hierarchically: *production facilities* replenish a set of *distribution centres*, which are strategically located to efficiently cover the distribution area. The products are successively distributed from these centres to a well-structured network of *transit points* which are in charge of the distribution to the final customers. Distribution is performed by a two-part system: (i) *primary shipping*, carried out with large trucks, from the production facilities to the distribution centres and from the distribution centres to the transit points; and (ii) *secondary shipping*, with smaller trucks, from the transit points to the final customers.

Our study addresses the secondary shipping from the transit points to the final customers. Not all customers require service every day, and each customer is associated with only one transit point. Hence, an instance of our districting problem is defined by a transit point, which can be in either an urban or suburban area. Almost all the customers require at least one visit per working week (Monday to Saturday), and the company's planners design template routes with a planning horizon of six days. The template routes are then implemented daily at the operational level, where last-minute requirements in the daily operations are taken into account. The use of template routes simplifies the administrative operations and also improves the driver's familiarity with the working regions, the customers, and their product demands. This familiarity allows a driver to run a route efficiently, making daily adjustments as necessary; it can also lead to a more accurate delivery process overall. The template routes, generally implemented for a variable number of weeks, are revised according to the seasonality of product

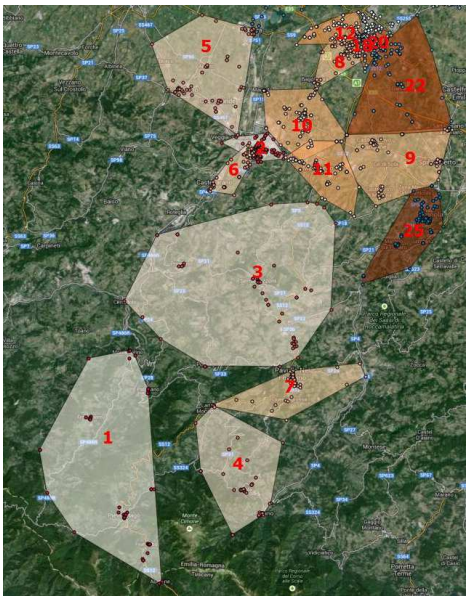


Figure 1: Distribution areas for a transit point.

demand and the acquisition of new customers. In practice, a set of template routes can be *locally* revised (routes are selectively updated based on new requirements) or *globally* revised (new template routes are generated, usually every 6-12 months). A crucial objective of the company is to minimise the total distribution cost while ensuring a high level of service.

Figure 1 shows a sample of distribution areas for a transit point where each area is identified by a different polygon. Each area is assigned to a driver, who is in charge of distributing the products to the customers included in the area. An additional feature of the problem considered in this paper is that the underlying distribution network is based on a real road network, hence both distances and travel times are computed using digital maps.

Below, we formally describe the problem addressed in this paper, followed by a literature review of related works.

2.1. Problem definition

The problem is defined with a planning horizon of n_d days represented by the index set $D = \{1, 2, \ldots, n_d\}$, and a distribution network represented by a complete directed graph $G = (N', A)$, where the set N' contains $n_c + 1$ vertices and A is an arc set. The set of arcs A represents the set of all possible links among the vertices associated with the locations of the customers and

of the depot. We have $N' = \{0\} \cup N$, where node 0 represents a depot (or a transit point) and the set $N = \{1, \ldots, n_c\}$ represents n_c customers.

A fleet of n_v identical vehicles, located at the depot, is available on each day of the planning horizon. Each vehicle has a capacity Q, a maximum working time L, and a fixed cost F.

For each customer $i \in N$, the following data are defined.

- A *service frequency* f_i representing the number of times customer i must be visited during the planning horizon;
- A set $C_i \subseteq 2^D$ of allowable day-combinations of f_i visit days. For example, a customer may require two visits (i.e., $f_i = 2$) during the planning horizon of six days (i.e., $n_d = 6$), and these visits can take place on one of the following day-combinations: Monday-Thursday, Monday-Friday, or Tuesday-Friday (i.e., $C_i = \{\{1, 4\}, \{1, 5\}, \{2, 5\}\}$. In the following, set C_i is also used to denote the index set of the allowable day-combinations;
- A quantity q_i^d that customer i must receive if visited on day d;
- A *service time* s_i^d to complete all operations at customer i if visited on day d;
- A time window $[e_i^d, l_i^d]$, where e_i^d and l_i^d represent the earliest and latest times for serving the customer i on day d.

In our application, value q_i^d, a total of all the different products to be delivered to the customer, is estimated based on historical data. In practice, the quantities may vary, because during the weekends a customer may require more products, and distribution is not performed on Sunday. However, daily quantities $\{q_i^d\}$ are generally variable only for customers having high frequencies (5 and 6). The service time s_i^d depends on both the time required for the delivery (as a function of the quantity q_i^d) and the time spent by the driver on administrative tasks related to the delivery, which can vary depending on the customer. Moreover, a customer's time window can vary during the week, due to different daily opening hours or customer preference. All vehicles are saturated in terms of weight capacity, hence values q_i^d and Q are expressed accordingly.

The non-negative cost (or distance) matrix $[c_{ij}]$ and time matrix $[t_{ij}]$ are associated with graph G, where c_{ij} and t_{ij} represent the transportation cost

and the time for traversing arc $(i, j) \in A$, respectively. The cost F is an estimate of the fixed weekly cost for using the vehicle.

A *feasible* route for a vehicle on day $d \in D$ is a simple circuit in G passing through the depot and a subset of the customers that can be visited on that day d, and such that:

(i) The sum of the customer demands is less than or equal to Q;
(ii) Each customer on the route is visited within its time window, so if the vehicle arrives at i on day d before e_i^d, the service is delayed to time e_i^d;
(iii) The total working time of the route, computed as the sum of the service, travel, and waiting times, is less than or equal to L.

The cost of a route (routing cost) is given by the sum of the costs c_{ij} of the arcs used by the route.

The problem consists of assigning every customer $i \in N$ to exactly one vehicle (or, equivalently, one area) to ensure driver consistency and designing at most n_v feasible routes for each day $d \in D$. Each vehicle performs at most one route—or, equivalently, each area is operated at most once for each day of the planning horizon, so that each customer $i \in N$ is visited f_i times, according to a feasible day-combination.

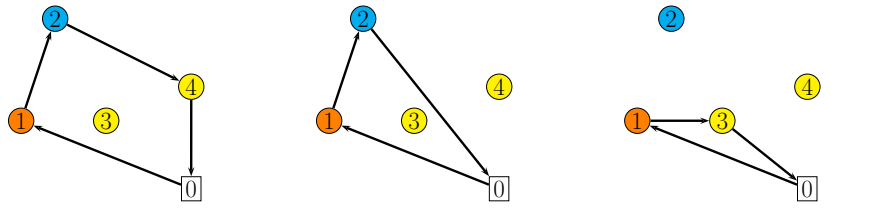
The objective is to minimise the sum of the fixed vehicle costs and the route costs. It follows that if cost F is set to a large enough constant value, then the objective is to minimise the number of vehicles (or areas) used and then minimise the sum of the route costs; whereas if F is set to zero, the objective reduces to minimising the total routing cost.

Figure 2 gives an example of a solution for the instance data reported in Table 1. The example involves a three-day planning horizon, eight customers, and two areas (vehicles) ($n_d = 3$, $n_c = 8$, and $n_v = 2$). For each area, the figure shows the three vehicle routes corresponding to the different days of the planning horizon. Customers having the same frequency values are represented with the same colours.

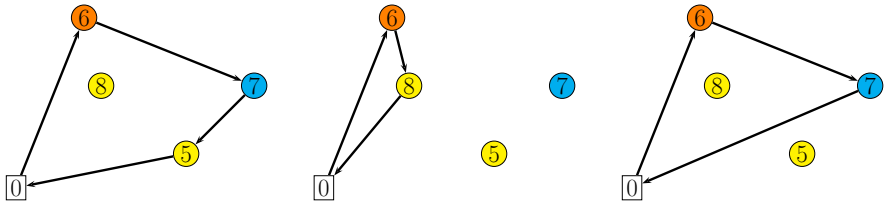
Hereafter, the following notation and terms are used. For a customer $i \in N$, the visit days of a day-combination $s \in C_i$ are represented by binary coefficients a_{si}^d, where a_{si}^d is equal to 1 if day d belongs to day-combination s, 0 otherwise. We have that $\sum_{d \in D} a_{si}^d = f_i$, $\forall s \in C_i$, and quantity q_i^d is equal to 0 if $a_{si}^d = 0$, $\forall d \in D, s \in C_i$ (i.e., customer i is not visited on day d). The terms *day-combination* and *pattern* are used interchangeably.

i	f_i	C_i	i	f_i	C_i
1	3	$\{1,2,3\}$	5	1	$\{1\}$, $\{3\}$
2	2	$\{1,2\}$, $\{1,3\}$	6	3	$\{1,2,3\}$
3	1	$\{1\}$, $\{3\}$	7	2	$\{1,2\}$, $\{1,3\}$
4	1	$\{1\}$, $\{2\}$	8	1	$\{2\}$, $\{3\}$

Table 1: Example: customers, frequencies and day-combinations.



(a) Area 1 - Day 1 (b) Area 1 - Day 2 (c) Area 1 - Day 3



(d) Area 2 - Day 1 (e) Area 2 - Day 2 (f) Area 2 - Day 3

Figure 2: Example of a solution for the data of Table 1.

2.2. Related work

To the best of our knowledge, the problem considered in this paper has never been addressed in the literature. Related problems are location problems [10] and VRPs [11, 12]. In particular, we deal with the joint optimisation of districting and routing problems, usually referred to in the literature as territory design problems (TDPs) [8].

Once the customers are assigned to vehicles, our problem is related to the PVRP, for which many variants were developed in the decades since its introduction by Beltrami and Bodin [13]. As noted, the PVRP is a generalisation of the classic capacitated VRP, consisting of designing a set of routes

for a homogeneous fleet of vehicles located at a central depot for each day of a given p-day period—subject to customers' day-combinations. The reader is referred to Francis et al. [14], Irnich et al. [15], Campbell and Wilson [16] for reviews of relevant models, variants, and solution techniques whereas for an efficient heuristic algorithm see Vidal et al. [17].

Models proposed in the literature for TDPs can first be classified by whether or not a routing component is considered, and then by whether they use continuous or discrete models. Examples of continuous models not considering the routing component are the ring radial network model [18] and the disk model [19]. Discrete models that do not consider a routing component are generally based on p-centers or p-median problems [9, 20]. Discrete routing-based models can be further classified by the time dimension of the planning horizon (single-period or multi-period problems). Sandoval et al. [21] propose an exact method based on an integer programming model which minimizes a p-center dispersion measure. Their approach solved test instances with up to 300 nodes. Moreno et al. [22] provide a hybrid approach for solving a territory design problem of a pork and poultry distributor based in the region of Valparaíso in Chile. Their approach generates the clusters by a K-means algorithm and then it combines the clusters using an integer programming model that minimizes the number of territories required to cover all of the clients. The resulting method allows significant savings when compared to the original solution used by the company. Solana et al. [23] consider the same problem considered in Salazar-Aguilar et al. [24], which requires the division of a geographical area into compact, contiguous and balanced territories, with respect to one or several measures of activity. They propose a matheuristic based on a Greedy Randomized Adaptive Search Procedure (GRASP).

Based on the above classification, our model belongs to the class of discrete, routing-based, multi-period problems; below we review related works. (For detailed reviews of models not addressed here, the reader is referred to Kalcsics [9] and the recent book by Ríos-Mercado [25]). Approaches for TDPs belonging to this class can be further classified, depending on the way the routing component is integrated: namely, whether a cluster-based routing or integrated routing approach is used. In cluster-based approaches [26, 5], two-stage algorithms are used for the routing, improving the assignment phase in which customers are assigned to districts or areas. Rodrigues and Ferreira [27] describe a two-stage algorithm with applications in solid waste collection where, in the first phase, a sectorization based on electromagnetism and

Coulomb's Law ("those belonging to the same sector should demonstrate some kind of *attraction*"; those "in different sectors should present some kind of *repulsion*") is followed by a second phase addressing the routing problems in each sector, modeled as a mixed capacitated arc routing problem. Integrated routing approaches [4] completely integrate the districting and routing components.

The problem studied in the work by Sungur et al. [26] is a stochastic multi-period VRP with soft time windows. The stochasticity is the result of uncertain service times and probabilistic customers. A master plan approach is used, and robust optimisation handles the uncertainty in the service time. Schneider et al. [5] developed a two-phase approach to investigate the design requirements of a TDP subject to time window constraints and then studied the influence of the constraints on its performance. To improve drivers' efficiency in logistics operations, Smilowitz et al. [4] enhanced workforce management by evaluating drivers' familiarity with both customers and regions, using the number of repeat visits by each driver to each customer and region. The authors designed three different heuristic approaches to solve a variation of the PVRP that includes consistency measures. Another related problem is the consistent vehicle routing problem (ConVRP) [28, 29, 30, 31], which aims to design synchronised routes on multiple days to serve a group of customers—while minimising the total travel cost. A survey of different modelling concepts and measurements related to ConVRPs can be found in the work by Kovacs et al. [32].

Problems closely related to ours have also been studied by Lei et al. [33], Lei et al. [34], Bender et al. [35] and Rodríguez-Martín et al. [36].

Lei et al. [33] introduced the multiple travelling salesmen and districting problem with multi-periods and multi-depots. In their problem, the customers must be partitioned into districts, and a depot must be assigned to each district; in each period of the planning horizon, each customer must be visited exactly once by a vehicle. The objective function adopted uses a weighted sum of four measures: the number of districts, the compactness of subdistricts, district similarity in subsequent periods, and salesmen's profit. The authors designed an adaptive large neighborhood search heuristic that was tested on instances with up to 400 customers and a maximum of three periods. Lei et al. [34] extended this work by considering deterministic or stochastic customers. Districts must be determined for each period of the planning horizon before the stochastic customers are revealed. This problem is formulated as a multi-objective optimisation problem and solved

with a multi-objective evolutionary algorithm. In both these problems, each customer must be served exactly once per period; hence customer visit day-combinations are not considered.

Bender et al. [35] investigated a multi-period service territory design problem which calls for partitioning the set of customers into service territories and determining the visiting weeks and days for each customer such that all districts, week clusters, and day clusters are balanced and compact. The main differences compared to our problem are that they prioritised geographical compactness and modelled visit combinations differently. More precisely, in their problem the visits of each customer must be periodically recurring according to a customer-specific weekly pattern; that is, each customer must be visited every pre-defined number of weeks. Further, there are restrictions on the number of visits per week and on which weekdays customers may be visited. The authors proposed a location-allocation heuristic that was tested on real-world instances involving up to 115 customers and 16 weeks. Based on the work of Bender et al. [35], Bender and Kalcsics [37] model a multi-period service territory design problem using a mixed-integer linear programming formulation. Since the proposed model has a high level of symmetry between solutions, the authors try to characterize these symmetries and propose ideas to eliminate them in the formulation and reduce the search space. They solve the scheduling component of the problem by a location-allocation based heuristic which determines visiting schedules for the service providers for fixed districts. They propose a branch-and-price algorithm to solve large instances to proven optimality.

The problem most closely related to the one addressed in this paper has been recently studied by Rodríguez-Martín et al. [36]. The authors introduced a new variant of the PVRP in which each customer is served by the same vehicle/driver on all visits. The resulting problem has been called the PVRP with driver consistency. A constraint on the maximum number of customers visited by a vehicle in a period is imposed. The objective is to minimise the total routing cost over the planning horizon. The authors proposed an integer linear programming formulation for the problem, together with families of valid inequalities and a branch-and-cut algorithm to solve the problem to optimality. The resulting algorithm was tested on randomly generated instances involving up to 71 customers, five periods and four vehicles. Our work considers a more general objective function and, more importantly, an additional set of operational constraints, such as the vehicle capacity constraints and the time window constraints, which are very

important in practice.

3. Hybrid genetic search

In this section, we describe a heuristic algorithm based on the hybrid genetic search with adaptive diversity control metaheuristic proposed by Vidal et al. [17]. This metaheuristic is one of the most efficient for solving VRPs, outperforming the current state-of-the-art metaheuristics, such as tabu search, scatter search, variable neighbourhood search, record-to-record ILP, and adaptive large neighbourhood search [17]. It combines the ability of local searches (LS) to find local optima by exploring a region of search space [38] with that of global searches to identify promising regions by using genetic algorithms (GA); in this paper their metaheuristic is further enhanced. Algorithms based on their metaheuristic are particularly effective in solving complex real-world problems [39].

There exists a large body of literature on metaheuristics. The book by Gendreau and Potvin [40] provides a broad coverage of the concepts, implementations, and applications in this important field of optimisation. In particular, concerning GAs and evolutionary algorithms, the book by Eiben and Smith [41] offers a thorough introduction to evolutionary computing, descriptions of popular evolutionary algorithm variants, discussions of methodological issues and particular evolutionary computing techniques.

In the following, we denote with $\mathcal{P}$ the population, which is a set of individuals further partitioned into n_{sp} subpopulations, denoted by $\mathcal{P}_s$, $s = 1, \ldots, n_{sp}$. The proposed heuristic is summarised in Algorithm 1, denoted as hybrid multi-population GA (`HMPG`).

The method *evolves* both feasible and infeasible solutions partitioned into subpopulations. The algorithm first generates an initial population $\mathcal{P}$ of feasible solutions (line 2) that is further partitioned into n_{sp} subpopulations $\mathcal{P}_s$, $s = 1, \ldots, n_{sp}$. At each iteration, the algorithm applies a number of operators to each subpopulation in order to (i) select two individuals (*parents*) and combine them to yield two new individuals (*offspring*) (lines 6-7), (ii) apply a mutation operator (line 8), and (iii) improve offspring by using a local improvement procedure (*education*) (line 9). The new offspring are then inserted into subpopulations according to a *similarity* function (line 10) aimed at grouping individuals with similar solution characteristics together. The algorithm then proceeds by applying crossover (*area-based crossover*) and a local improvement procedure (lines 12-16). These steps generate new

Algorithm 1 HMPG

Input: A problem instance
Output: Best feasible solution

1: $it \leftarrow 0$, $it_{nimp} \leftarrow 0$, $it_{sf} \leftarrow 0$, $\Delta \leftarrow \Delta_0$ ▷ Initialisation
2: Generate the initial population $\mathcal{P}$ (§3.7.1)
3: Partition population $\mathcal{P}$ into n_{sp} subpopulations $\{\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_{n_{sp}}\}$
4: **while** ($it_{nimp} < IT_{nimp}$ and CPU time $< CPU_{max}$) **do** ▷ Main loop
5: **for all** $\overline{\mathcal{P}} \in \{\mathcal{P}_1, \ldots, \mathcal{P}_{n_{sp}}\}$ **do**
6: Select parent solutions $P1$ and $P2$ from $\overline{\mathcal{P}}$ (*binary tournament*)
7: Generate offspring $C1$ and $C2$ from $P1$ and $P2$ (*routing-crossover*, §3.4)
8: Randomly select $C1$ or $C2$ and create new solution $C3$ (*mutation*, §3.5)
9: Educate offspring $C1$, $C2$, and $C3$ with probability p_{ls} (*local improvement procedure*, §3.6)
10: Insert $C1$, $C2$, and $C3$ into subpopulations based on their *similarities* (§3.7.2)
11: **end for**
12: Randomly select two subpopulations $\mathcal{P}_{i_1}$ and $\mathcal{P}_{i_2}$ from $\mathcal{P}$
13: Select parent solutions $P1 \in \mathcal{P}_{i_1}$ and $P2 \in \mathcal{P}_{i_2}$
14: Generate offspring $C1$ and $C2$ from $P1$ and $P2$ (*area-crossover*, §3.4)
15: Educate offspring $C1$ and $C2$ with probability p_{ls} (*local improvement procedure*)
16: Insert $C1$ and $C2$ into subpopulations based on their *similarities*
17: **for all** $\overline{\mathcal{P}} \in \{\mathcal{P}_1, \ldots, \mathcal{P}_{n_{sp}}\}$ **do**
18: **if** ($|\overline{\mathcal{P}}| \geq \Delta$) **then**
19: Select survivors (§3.7.2)
20: **end if**
21: **end for**
22: **if** ($it_{sf} = IT_{sf}$) **then**
23: $\Delta \leftarrow \Delta_0 f_{shf}(it)$ (*shrinking* function, §3.7.2) , $it_{sf} \leftarrow 0$
24: **end if**
25: **if** (the incumbent solution has been improved) **then** $it_{nimp} \leftarrow 0$
26: **else** $it_{nimp} \leftarrow it_{nimp} + 1$
27: **end if**
28: $it \leftarrow it + 1$, $it_{sf} \leftarrow it_{sf} + 1$
29: **end while**
30: **return** best feasible solution

individuals based on the combination of the area structures of individuals belonging to different subpopulations, thus diversifying the search. In lines 17-21 of the algorithm, a survivor selection mechanism is executed whenever the number of individuals in a subpopulation reaches Δ (line 18). The value of Δ is dynamically changed during the iterations (line 23) and updated every IT_{sf} iterations based on a *shrinking* function $f_{shf}(.)$ which reduces the value of Δ.

In the algorithm, it represents the incumbent number of iterations, whereas it_{nimp} and it_{sf} are the number of non-improving iterations and the number of iterations since the value of Δ was updated, respectively.

Our algorithm mainly differs from the scheme proposed by Vidal et al. [17] as follows.

- It generates the initial population using a greedy heuristic instead of randomly generating it.
- It uses a diversification mechanism to combine individuals from different subpopulations (lines 12-16).
- It applies a local improvement procedure based on a simulated annealing (SA) solution framework.
- It dynamically reduces, by means of the *shrinking* function $f_{shf}(.)$, the maximum sizes of the subpopulations used to activate the survivor selection mechanism.

It is worth noting that for solving large-scale instances of our problem, it is of great important to ensure that Algorithm 1 runs in polynomial time. Therefore, the different components of the algorithm have been implemented to guarantee this property. For the sake of the exposition, we omit the corresponding technical details, and we refer the reader to Vidal et al. [42] for a general discussion about this issue. In the next subsections, we describe the different components of Algorithm 1 in detail.

3.1. Solution representation

The first step in defining an evolutionary algorithm is to represent solutions of the underlying problem as *individuals*. In Algorithm `HMPG`, individuals are represented by means of two *chromosomes*.

Chromosome 1

Area	1	1	1	1	2	2	2	2
Customer	1	2	3	4	5	6	7	8
Day-combination	1	1	2	1	1	1	2	1

Chromosome 2

Day	1			2		3		1			2		3	
Routing	1	2	4	1	2	1	3	6	7	5	6	8	6	7
Area	1							2						

Figure 3: Chromosome encoding for the solution of Figure 2.

(1) *Chromosome 1* represents the partitioning of the customers into areas and the corresponding pattern of visits. The first tier of the chromosome contains the area indices and the second and the third tiers give the customer indices and the associated day-combinations, respectively.

(2) *Chromosome 2* represents the vehicle routes and also includes three tiers. The first tier gives the day of the planning period. The second and the third tiers contain the vehicle routes and areas, respectively. The vehicle routes are represented by the sequence of the customers visited. The routes do not include the depot that is both the start and end of the route.

Figure 3 provides an example of chromosomes representing the solution reported in Figure 2 for the instance data of Table 1. Note that the customers' indices reported in the second tier of Chromosome 1 are not necessary; they have been introduced to clarify the description.

Algorithm `HMPG` explores both feasible and infeasible solutions. More precisely, an individual always represents a partition of the set of customers into different areas, and each customer has been assigned exactly one day-combination from among all its day-combinations (in Chromosome 1). The individual is infeasible if a routing constraint is violated by the vehicle routes represented by Chromosome 2.

3.2. Greedy algorithm for generating initial population

In this section, we describe a two-phase greedy algorithm (`TFG`) used to build feasible solutions and generate the initial population for Algorithm `HMPG` (see Section 3.7.1). The greedy algorithm performs the three main tasks defining our problem: (i) partitioning the customers into areas, (ii) assigning day-combinations to the customers, and (iii) routing the vehicles according to constraints.

The proposed algorithm comprises two phases. In the first, the *Partition and pattern assignment*, customers are partitioned into areas and day-combinations are assigned to the customers. In the second phase, the *Routing*, vehicle routes are built to serve the customers. In the following the two phases are described in detail.

Phase 1: Partition and pattern assignment

We define a *seed* to be a representative customer of an area. For a customer $i \in N$ we define $\overline{q}_i = \sum_{s \in C_i} \sum_{d \in D} a_{si}^d q_i^d / |C_i|$ to be an estimate of the total demand of the customer over the planning horizon, and we also define $\overline{Q} = n_d Q$ to be the total aggregate capacity of a vehicle along the planning horizon.

The algorithm starts partitioning customers into areas using customer demands $\overline{q}_i$ and vehicle capacity $\overline{Q}$. The procedure, based on the heuristic algorithm proposed by Mulvey and Beck [43] for the capacitated p-median problem, works as follows. Given a set of seeds $\{s_1, s_2, \ldots, s_k\}$ with $k \leq n_v$, the partition minimises the total customer assignment *regret*, where the regret of a customer assignment is defined as the absolute value of the difference in routing cost between the customer's first- and second-nearest seeds. The procedure assigns customers to the seeds by decreasing the regret value. When (and if) all customers are assigned—that is, customers are all clustered around the respective seeds, an intracluster phase reassigns each cluster or area to the seed that minimises the sum of the routing costs between it and all other cluster members. If a new set of seeds is identified, the assignment/reassignment process is repeated. When the seeds remain stable across iterations, pairwise interchanges of customers between clusters are performed to locally optimise the solution. We denote with $\{s_1, s_2, \ldots, s_k\}$ the final set of seeds computed by the above procedure and with $\overline{A}_h$, $h = 1, \ldots, k$, the resulting clusters or areas. We have $\sum_{i \in \overline{A}_h} \overline{q}_i \leq \overline{Q}$, $h = 1, \ldots, k$.

A step-by-step description of the partition and pattern assignment phase is given in Algorithm 2. In the algorithm, the set $\overline{N}$ represents the set

of remaining customers, whereas sets A_h represents the set of customers comprising the incumbent area h. Further, set A_h^d is the set of customers assigned to day d according to the day-combinations selected for the customers. Value $\overline{Q}_d$ is the total demand associated with day d, based on the set of customers in A_h and the corresponding day-combinations. Value $\overline{L}_d$, computed as the minimum-time arborescence problem rooted at the depot using customers in A_h^d (see [44]), represents a lower estimate of the total working time of the vehicle route on day d, disregarding the time window constraints. The day-combinations are defined for each customer by means of a function $f_{dca}(.)$ which computes a correlation between the days of the planning horizon and the set of available day-combinations for that customer. More precisely, given $i \in N$, we first compute coefficients $p_i^d = \sum_{s \in C_i} a_{si}^d / |C_i|$, $\forall d \in D$; the day-combination l^* assigned to customer i is computed as $l^* = f_{dca}(i, \mathbf{p}_i) = \text{argmax}_{s \in C_i}\{\sum_{d \in D} a_{si}^d p_i^d\}$. Any ties are broken by selecting the combination resulting in the largest total residual vehicle capacity, computed over the days of the combination and based on values $\overline{Q}_d$.

The algorithm (line 1) takes as input k seeds and, using the procedure described above, generates an initial partition $\overline{A}_h$, $h = 1, \ldots, k$, of the customers. We assume that the number of seeds k selected ensures the existence of a partition.

The set of remaining customers $\overline{N}$ to be partitioned is initialised in line 2. For each area h, the candidate area $\overline{A}$ is initialised as $\overline{A}_h$ (line 4). The seed i^* of the area (line 5) is assigned with a day-combination (line 6), and the emerging area A_h and sets A_h^d are computed. Customers in $\overline{A}$ are then selected based on the routing cost from the seed i^*, in order to try to insert them into the emerging area A_h (line 8). Given a customer $i \in \overline{A}$, we first compute its day-combination l and then check if the corresponding insertions on the days associated with l are feasible with respect to values $\overline{Q}_d$ and $\overline{L}_d$ (lines 8-10). A parameter $0 < \omega < 1$ is used to reduce the value of the maximum working time L and compare it to the estimate $\overline{L}_d$. If the insertion of i is feasible, area A_h is updated and i is removed from set $\overline{N}$ (line 11). Notice that the day-combinations in steps 6 and 8 are defined using function $f_{dca}(.)$. If set $\overline{N}$ is not empty at the end of Algorithm 2, sets A_h, $h = 1, \ldots, k$, are not a partition of the set of customers. In this case, a new initial set of seeds must be considered in order to try to define a partition of the customers.

Algorithm 2 Partition and pattern assignment phase.

Input: Set of k seeds or customers $\{s_1, s_2, \ldots, s_k\}$

Output: Set of areas $\{\overline{A}_1, \overline{A}_2, \ldots, \overline{A}_k\}$ representing a partition of the customer set and day-combinations $\{l_i\}$

Build initial areas $\{\overline{A}_1, \overline{A}_2, \ldots, \overline{A}_k\}$
$\overline{N} \leftarrow N$
for $h = 1, \ldots, k$ **do**
    Select the candidate area $\overline{A} \leftarrow \overline{A}_h$
    Select the seed customer $i^* \leftarrow s_h$, $\overline{N} \leftarrow \overline{N} \setminus \{i^*\}$, $\overline{A} \leftarrow \overline{A} \setminus \{i^*\}$
    $A_h \leftarrow \{i^*\}$, $l_{i^*} \leftarrow f_{dca}(i^*, \mathbf{p}_{i^*})$ and update A_h^d, $d \in D$
    while $(\overline{A} \neq \emptyset)$ **do**
        Select $i \in \overline{A}$ as the closest customer to i^*, $l_i \leftarrow f_{dca}(i, \mathbf{p}_i)$, $\overline{A} \leftarrow \overline{A} \setminus \{i\}$
        Compute $\overline{Q}_d$, and $\overline{L}_d$, $d \in D$ based on A_h, i and l_i
        if $(\overline{Q}_d \leq Q$ and $\overline{L}_d \leq \omega L$, $d \in D)$ **then**
            $A_h \leftarrow A_h \cup \{i\}$, update A_h^d, $d \in D$, $\overline{N} \leftarrow \overline{N} \setminus \{i\}$
        end if
    end while
end for

return Customers areas A_h, $h = 1, \ldots, k$, and day-combinations $\{l_i\}$

Phase 2: routing

By the end of the first phase, the customer set N has been partitioned into k areas (A_h, $h = 1, \ldots, k$) each with its corresponding day-combinations $\{l_i\}$. For each area, the day-combinations also define the customers to be served on each day of the planning horizon (A_h^d, $d \in D$, $h = 1, \ldots, k$). Therefore, the overall problem decomposes into a problem for each area, which is further decomposed into n_d subproblems. Note that after Phase 1, the vehicle capacity constraints are satisfied because the vehicle capacity Q is explicitly considered for each day of the planning period when the day-combinations are defined.

Each subproblem corresponds to a generalisation of the well-known traveling salesman problem with time window constraints (TSPTW), which finds in a weighted digraph a least-cost tour starting from a selected vertex (the depot), visiting each vertex of the graph exactly once according to a given time window, and returning to the starting vertex [45].

Savelsbergh [46] showed that even the problem of verifying if a feasible TSPTW solutions exists is an $\mathcal{NP}$-complete problem. Our subproblem adds the maximum working time constraint to the TSPTW. Note that this constraint cannot be modelled by associating a generic time window $[e_0, l_0]$ such that $L = l_0 - e_0$ with the depot, because in our case the departure time of the vehicle from the depot is flexible, and the total working time is a function of the departure time.

To solve each subproblem, we used the generalised insertion heuristic for the TSPTW proposed by Gendreau et al. [47], modified to consider the maximum working time constraint. The algorithm gradually builds a route by inserting, at each step, a vertex in its neighbourhood on the current route. This is done by dynamically changing the departure time of the vehicle—while checking the feasibility of the remaining part of the route and the maximum time duration constraint. The emerging route is also locally reoptimised. A post-optimisation phase based on the successive removal and reinsertion of all vertices is also executed after a feasible route has been determined. For the details of the algorithm, the reader is referred to Gendreau et al. [47].

3.3. Evaluation and similarity functions

Algorithm `HMPG` considers both feasible and infeasible solutions, using the following *fitness function* $f_P(S)$ to compute the cost associated with a given

individual S:

$$f_P(S) = f(S) + \alpha_1 \cdot P_{TW}(S) + \alpha_2 \cdot P_{WT}(S) + \alpha_3 \cdot P_{VC}(S), \qquad (1)$$

where $f(S)$ is the objective value of solution S, as defined in Section 2. The penalty functions $P_{TW}(S)$, $P_{WT}(S)$, and $P_{VC}(S)$ measure the violations of the time window, maximum working time, and vehicle capacity constraints, respectively. Their computation is based on similar functions used in the literature to penalize infeasible solutions [17, 39]. The corresponding penalty coefficients α_1, α_2, and α_3 are user-defined parameters.

Let S_1 and S_2 be two (not necessarily feasible) solutions. For the pair of solutions (S_1, S_2) we introduce a *similarity* function $f_{sf}(S_1, S_2)$ that measures how similar solution S_1 is to solution S_2. Function $f_{sf}(S_1, S_2)$ is defined as follows.

Let n_1 and n_2 be the number of areas forming solutions S_1 and S_2, respectively; without loss of generality we assume $n_1 \leq n_2$. We denote the corresponding index sets of the areas with $I_1 = \{1, 2, \ldots, n_1\}$ and $I_2 = \{1, 2, \ldots, n_2\}$. The following two additional functions are used to compute function $f_{sf}(S_1, S_2)$:

- Function $f_a(h_1, h_2)$, $h_1 \in I_1$, $h_2 \in I_2$ provides the number of customers in common between areas h_1 and h_2 of solutions S_1 and S_2.
- Function $f_b(S_1, S_2, i) \in \{0, 1\}$ is equal to 1 if customer i has been assigned the same day-combination in solutions S_1 and S_2, and 0 otherwise.

Function $f_{sf}(S_1, S_2)$ is defined as the following weighted function:

$$f_{sf}(S_1, S_2) = \frac{\beta_1}{n_c} \sum_{h_1 \in I_1} \max_{h_2 \in I_2} \{f_a(h_1, h_2)\} + \frac{\beta_2}{n_c} \sum_{i \in N} f_b(S_1, S_2, i), \qquad (2)$$

where $\beta_1 \geq 0$, and $\beta_2 \geq 0$ are user-defined parameters such that $\beta_1 + \beta_2 = 1$. In (2) the first summation computes, for each area $h_1 \in I_1$, the maximum number of customers in common with the areas in I_2, whereas the second summation counts the number of customers having the same day-combinations in solutions S_1 and S_2. We have that $f_{sf}(S_1, S_2) = f_{sf}(S_2, S_1) = 1$ if S_1 and S_2 are identical in terms of the areas defined and the day-combinations associated with the customers.

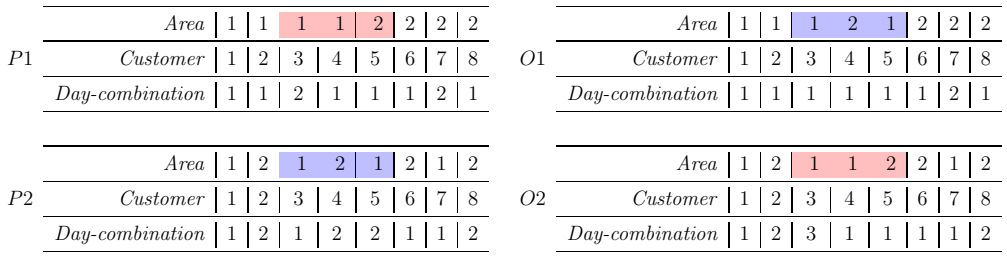


Figure 4: The 2PX crossover.

3.4. Parent selection and crossover

Parent selection is performed through a binary tournament, which randomly (with uniform probability) picks two individuals (parents) from the selected population and keeps the one with the best fitness function.

Crossover operators diversify the search by combining the features from the parents in the offspring. In our algorithm, we propose two different crossover operators, applied to Chromosomes 1 and 2: *area-crossover* and *routing-crossover*.

The *area-crossover* operator is a 2-point crossover operator (hereafter denoted as 2PX) which works on the first chromosome with the aim to diversify the search based on the area-customer assignments. Given two parents $P1$ and $P2$, two random numbers a and b are chosen in the range $[1, n_c]$. The two parents are split at points a and b to create two offspring, $O1$ and $O2$, by exchanging the segments $[a, b]$ of the parents. Figure 4 shows an example based on the instance of Table 1, where the selected segments of parents $P1$ and $P2$ are represented with different colors. In the new offspring, customers not involved in the exchange will maintain the same area indices and day-combinations of the corresponding parents, while the customers involved in the exchange (Customers 3, 4, and 5 in the example) will be assigned to new areas. These customers are first removed from the routes of Chromosome 2 and then (per increasing customer index) re-inserted into the offspring, using the least extra-cost day-combination among the day-combinations of the customer. Day-combinations of Chromosome 1 are modified accordingly.

The *routing-crossover* operator is a 1-point crossover operator (hereafter denoted as 1PX) which works on the second chromosome to diversify the search, based on the day-combinations and visit sequences (i.e., the vehicle routes) of the customers. Given two parents $P1$ and $P2$, two random

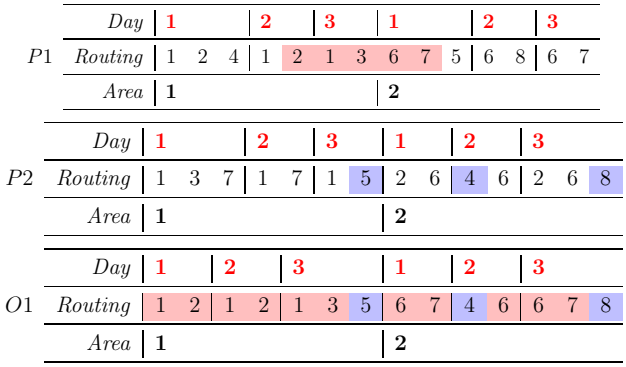


Figure 5: The 1PX crossover.

numbers a and b are chosen in the range $[1, \sum_{i \in N} f_i]$, where the upper limit represents the total number of visits of any individual. In this case, only one offspring is generated. The new offspring $O1$ is created as follows. The selected customers in the segment $[a, b]$ of parent $P1$ are first associated with the same day-combinations and area in $O1$. Figure 5 shows an example of an 1PX operator based on the example in Table 1, where customers 2, 1, 3, 6, and 7 are involved in the crossover operation and appear in $O1$ as they did in parent $P1$. Offspring $O1$ is then completed with the remaining customers (4, 5, and 8) not involved with the selected segment of $P1$. More precisely, customers in $P2$ but not $P1$ are considered in turn, based on the routing sequence defined by $P2$ (i.e., 5, 4, and 8), and inserted in $O1$ in the same area with the same day-combination as $P2$. A least-cost insertion criteria is chosen to define the positions of the customers in the new routes of $O1$.

3.5. Mutation operation

Mutation is performed by varying the genes of a chromosome to prevent the algorithm from being trapped in local optima, thus improving the diversity of the solutions. In this paper, we use the following mutation operators.

1. *Customer swaps.* The operator randomly selects NP_{MUT} pair of customers from different areas; for each pair (i, j) of customers the area of customer i is assigned to customer j, and vice versa (in Chromosome 1). Chromosome 2 is updated by first removing customers i and j and then selecting (for i followed by j) the least-cost insertions based on

the day-combinations and vehicle routes. Chromosome 1 is updated accordingly.

2. *Customer moves.* The operator randomly selects NC_{MUT} customers; each one is removed from its original area and re-inserted into a different area. More precisely, the operator inserts each removed customer into the least-cost area remaining. The insertion of customers into an area is performed as for operator *Customer swaps*.
3. *Area opening.* The operator randomly selects an area and NC_{MUT} customers belonging to that area. Then, a new area is created and the selected customers are moved into it. The insertion of the customers into the new area is performed as for operator *Customer moves*.
4. *Area closing.* The operator randomly selects an area and closes it. All the customers belonging to the closed area are inserted into the remaining areas by the same insertion heuristic used by *Customer moves*.

In the above moves, the different day-combinations associated with a customer are evaluated using a least-cost insertion heuristic operating on the selected set of routes. Furthermore, for a given individual, each of the above operators is chosen with equal probability. All random operations are performed by a uniform distribution over the respective range.

3.6. Local improvement procedure

A crucial feature of evolutionary algorithms is their ability to improve solutions (based on local improvement procedures or LS) through a process capable of escaping from local optima and performing an efficient search of the solution space.

In this section, we describe an SA based algorithm [48] which is an extension of LS optimisation algorithms. As for general LS algorithms, this algorithm creates a systematic way to explore the solution space using *neighbourhood structures*. For a complete overview of SA and related applications, the reader is referred to Delahaye et al. [49].

Following the SA scheme, the procedure starts with a given solution. At each iteration of the algorithm, new solutions are generated using neighbourhoods of the current solution. If the new solution improves the current one, then it replaces the current one. Otherwise, the new solution becomes the current one with a given probability. This latter move is one of the main features of SA: the ability to accept solutions that degrade the objective function. The algorithm uses a parameter, *temperature* T, initialised at a high

value. As the temperature decreases, only moves improving the objective, or having little objective deterioration, are accepted. Allowing moves with objective degradation ensures that the solution space is thoroughly explored. Finally, when the temperature tends to zero, no further deterioration of the objective is accepted. In our implementation, the temperature T is set to an initial value T_0 at the beginning of Algorithm `HMPG` and decreases according to a cooling parameter $\alpha_T \in (0, 1)$, which has a minimum value equal to T_f. The temperature T is updated at each main iteration of Algorithm 1.

We designed four neighbourhood structures, which are described in detail below. The four structures are examined in random order (using a uniform distribution); the sequence is terminated at the first improving move. Within each structure, the moves are executed in sequence.

3.6.1. Routing improvement

To try to improve the single-vehicle routes on the different days of the planning horizon, we adopt two well-known neighbourhoods originally designed for the TSP by Lin and Kernighan [50], the 2-opt and 3-opt neighbourhoods. The neighbourhood of the solution of a 2-opt (3-opt) operator is the set of solutions that can be reached from the solution by deleting two (three) arcs in the solution, and adding two (three) other arcs in order to reconnect the route (e.g., see [42] for a detailed description of these moves).

3.6.2. Day-combination improvement

The definition of the day-combinations associated with the customers plays a crucial role in the quality of the solutions. Hence, we designed two specific neighbourhoods based on the day-combinations.

(i) *Day-combination reassignment.* For a customer $i \in N$ with associated day-combination $s^* \in C_i$, we consider in turn each alternative day-combination in $C_i \setminus \{s^*\}$. Customer i is removed from its current set of routes and the day-combinations in $C_i \setminus \{s^*\}$ are evaluated to see if they improve the solution. The operation is repeated by randomly selecting NC_{LP} customers.

(ii) *Route elimination.* This neighbourhood reduces the number of routes in order to decrease the solution cost. Indeed, given a solution with vehicle routes operating on each day of the planning horizon, an improved solution may be obtained by eliminating a route on one day and reassigning its customers to the other days. For a given area, we select

the vehicle route having the minimum percentage of vehicle utilisation computed with respect to the vehicle capacity. We also try to reassign the corresponding set of customers to days other than the current day, using the same procedure used for the day-combination reassignment.

3.6.3. Area improvement

This neighbourhood operator evaluates new customer-area assignments by repositioning customers among areas.

For a customer $i \in N$ we define the set of customers $N(i)$ that contains the NCC_{LP} customers nearest to i (in terms of the cost matrix $[c_{ij}]$) belonging to areas other than the area of customer i. We denote with $A(i)$ the set of areas associated with the customers nearest to customer i. Based on the definitions of sets $\{N(i)\}$, the following neighbourhoods are considered.

(i) *Customer move.* This move randomly selects NC_{LP} customers and tries to move each selected customer i into the least-cost area belonging to set $A(i)$. The insertion into a new area is performed by choosing the least extra-cost day-combination for the vehicle routes of the selected area.

(ii) *Customer swap.* This move randomly selects NC_{LP} customers. Then, for each selected customer i and for each $j \in N(i)$, i and j are removed from their respective areas; i is inserted into the area of j, and vice versa. The insertion is again performed by selecting the least extra-cost day-combinations.

(iii) *Area elimination.* This move is based on the *area closing* move described in Section 3.5. The move selects an area having a number of customers less than or equal to $\alpha_{LP}\, n_c$ (if any), and tries to move each customer in that area into the least-cost area by means of the same procedure used for the *Customer move*.

All random operations are performed by a uniform distribution over the respective range.

3.6.4. Feasibility improvement

The problem addressed in this paper is characterised by complex routing constraints. Allowing the search to move to infeasible solutions enlarges the search space and can lead to high-quality solutions. Indeed, as mentioned previously, our algorithm considers both feasible and infeasible solutions; infeasible solutions are associated with a penalty term in the objective function

(1). However, this raises the issue of finding correct weights for the penalty terms [41] and reducing the infeasibility as soon as the penalty terms become dominant in the evaluation function. To resolve this issue, whenever the penalty term associated with the evaluation of an area exceeds a given threshold, the area is eliminated and two new areas are created using the greedy algorithm described in Section 3.2. The threshold is computed as a function of the number of customers and the number of non-improving iterations.

3.7. Population management

In this section, we describe the Algorithm `HMPG`'s generation of the initial population $\mathcal{P}$ and the population management mechanisms which identify and propagate the characteristics of good solutions and enhance population diversity.

3.7.1. Population initialisation

To initialise the population $\mathcal{P}$ of Algorithm `HMPG` we use Algorithm `TFG` (described in Section 3.2).

Let $Q_{tot} = \sum_{i \in N} \sum_{s \in C_i} \sum_{d \in D} a_{si}^d q_i^d / |C_i|$ be an estimate of the total customer demand over the planning horizon, and let $\overline{n}_v = \left\lceil \frac{Q_{tot}}{n_d Q} \right\rceil$ be an estimate of the number of vehicles (or areas) required to serve the whole set of customers. The population $\mathcal{P}$ is generated by executing the following steps for a pre-defined number of iterations.

1) Randomly sample the number of seeds k in the interval $[\overline{n}_v + ST_{PM}, n_v]$.
2) Generate a set of seeds $U = \{s_1, \ldots, s_k\}$ as follows. Initialise $U = \{i\}$ with i randomly sampled from the set N. Then, for each $h = 2, \ldots, k$, set U is expanded with a customer i^* in $N \setminus U$ such that $i^* = \text{argmax}_{i \in N \setminus U} \{\sum_{j \in U} c_{ij}\}$; i.e., the customer that maximises the cost with respect to the current set of seeds is selected to expand set U. Let $U = \{s_1, s_2, \ldots, s_k\}$ be the final set of seeds generated.
3) Use the algorithm `TFG` to generate a solution S using $U = \{s_1, s_2, \ldots, s_k\}$ as the initial set of seeds (see line 1 of Algorithm `TFG`).

At Step 2 of the above procedure, a final set U is checked for repetition, and at each iteration the generated solution S is added to the population $\mathcal{P}$. Set $\mathcal{P}$ contains NS_{PM} solutions, where NS_{PM} is a user-defined parameter.

The whole set of individuals (solutions) $\mathcal{P}$ generated is further partitioned into n_{sp} subpopulations by first initialising each subpopulation $\mathcal{P}_s$,

$s = 1, \ldots, n_{sp}$, with a single solution as follows. We randomly select a solution $S \in \mathcal{P}$ that is used to initialise the first subpopulation and is removed from $\mathcal{P}$; i.e., $\mathcal{P}_1 = \{S_1\}$ where $S_1 = S$. Then, the next subpopulation $\mathcal{P}_s$ is initialised by selecting the solution S' from $\mathcal{P}$ such that $\sum_{h=1}^{s-1} f_{sf}(S', S_h)$ is minimised (to create diversified initial subpopulations). Set $\mathcal{P}_s$ is initialised as $\mathcal{P}_s = \{S_s\}$, where $S_s = S'$ and solution S' is removed from $\mathcal{P}$. The procedure is repeated up to $s = n_{sp}$.

Once the initial single-individual subpopulations $\{\mathcal{P}_1, \ldots, \mathcal{P}_{n_{sp}}\}$ have been identified, each individual S in the remaining set $\mathcal{P}$ is inserted into the appropriate subpopulation as follows. For a solution S we compute

$$\Psi(S, \mathcal{P}_s) = \frac{1}{|\mathcal{P}_s|} \sum_{i=1}^{|\mathcal{P}_s|} f_{sf}(S, S_i), \quad \forall s = 1, \ldots, n_{sp}, \tag{3}$$

and S is inserted into the subpopulation $\mathcal{P}_s$ maximising function $\Psi(S, \mathcal{P}_s)$ (i.e., individuals sharing similar characteristics are inserted into the same subpopulation).

3.7.2. Multi-population evolution strategy

The multi-population evolution strategy is composed of n_{sp} subpopulations, which are independently managed; their sizes or cardinalities dynamically change during the iterations.

The survivor selection mechanism identifies individuals to make up the next generation so that population diversity is preserved and elite individuals (in terms of cost) are protected. More precisely, once the cardinality of a subpopulation $\mathcal{P}_s$ is greater than the maximum size Δ (see line 18 of Algorithm `HMPG`), a survivor selection mechanism is used to reduce the number of individuals in the subpopulation by removing $|\mathcal{P}_s| - \Delta$ individuals, as follows.

For each pair of solutions $S_1, S_2 \in \mathcal{P}_s$ we compute the function $f_{sf}(S_1, S_2)$. Let sf_{min} and sf_{max} be the minimum and maximum values obtained in computing values $f_{sf}(S_1, S_2)$, respectively. Let Θ be the index set of pairs (S_1, S_2), ordered for decreasing values of $f_{sf}(S_1, S_2)$ in the range $[sf_{max} - (sf_{max} - sf_{min})/3, sf_{max}]$ (pairs outside of that range are excluded), such that $|\Theta| \leq |\mathcal{P}_s| - \Delta$. For a pair $h \in \Theta$, we denote the corresponding pair with (S_1^h, S_2^h) and the value $f_{sf}(S_1, S_2)$ with sf_h.

We first remove from $\mathcal{P}_s$ at most $|\mathcal{P}_s| - \Delta$ individuals by selecting individual S_1^h from each pair (S_1^h, S_2^h), $h \in \Theta$; that is, we remove individuals

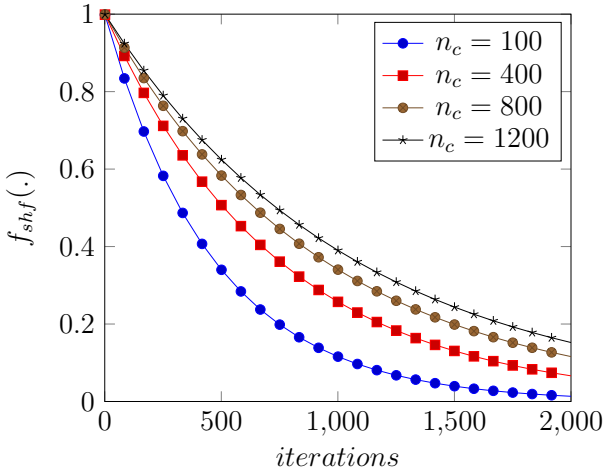


Figure 6: Shrinking function $f_{shf}(i)$ with $\eta = 5$ and $H = 1000$

having similar characteristics, in order to preserve population diversity. Let $\mathcal{P}_s$ be the resulting set of individuals. If $|\mathcal{P}_s| > \Delta$, then the individuals in $\mathcal{P}_s$ are ordered for increasing values of the fitness function and the first Δ are selected to form the new subpopulation; hence elite individuals, in terms of cost, are selected to form the next generation.

To speed up the computation—while at the same time propagating the characteristics of good solutions and providing the means for a thorough and efficient search—the maximum subpopulation size is dynamically reduced during the different algorithm iterations. More precisely, at a generic iteration of the algorithm the maximum subpopulation size Δ_0 is reduced by means of a *shrinking function*. The shrinking function, denoted as $f_{shf}(.)$, is defined as:

$$f_{shf}(i) = \eta e^{\frac{-i}{H \sqrt[3]{n_c}}}, \tag{4}$$

where i represents the iteration number and η and H are user-defined parameters. Then, the value Δ is computed as $\Delta = \Delta_0 f_{shf}(i)$. Figure 6 shows an example of function $f_{shf}(i)$ for different values for the number of customers n_c.

To properly update the subpopulation size, the maximum Δ is updated during the main iteration of the algorithm every ISF_{PM} iterations.

3.8. Termination criteria

To comprehensively address the behaviour of the Algorithm HMPG, the termination condition combines the maximum number of non-improving iterations IT_{nimp} and a maximum CPU time CPU_{max}.

4. Computational study

We present extensive experimental analysis with three main aims. Firstly, we evaluate the performance of Algorithm HMPG for test instances from the literature, and measure the effectiveness of the algorithm by the quality of the solutions produced. We consider the PVRP with driver consistency, as considered by Rodríguez-Martín et al. [36], and we compare the solutions produced by the algorithm with the solutions obtained by the exact method of Rodríguez-Martín et al. [36] (Section 4.1). Secondly, we extensively analyse the performance of our algorithm using real-world instances (Section 4.2) and conduct experiments to measure the contribution of each main component of HMPG. Finally, we conclude this section with a sensitivity analysis to evaluate the impact of different service requirements on the obtained solutions. With these results, we derive some managerial insights by studying the trade-offs between the service requirements.

The metaheuristic was coded in C using Visual Studio 2013 and run on a single thread of a 1.99 GHz Intel i7-8550 CPU under Windows 7 operating system. In order to identify good parameter values for our algorithm, we performed a parameter calibration (see Appendix A for the details).

4.1. Results on the instances from the literature

In this section, we evaluate the performance of Algorithm HMPG on the set of PVRP with driver consistency instances generated by Rodríguez-Martín et al. [36].

Test instances

Rodríguez-Martín et al. [36] generated benchmark instances in which driver consistency is not implicit in the solution of the corresponding PVRP; hence the corresponding instances cannot be solved as pure PVRP instances. The authors generated instances with a number of customer nodes in $\{10, 20, 30, 40, 50, 60, 70\}$. All the node coordinates were randomly generated in the interval $[0, 100] \times [0, 100]$. The depot is placed at node 0

and the customers at the other nodes. The routing costs c_{ij} are computed as the Euclidean distance between i and j. The number of periods ranges between 2 and 5; the number of vehicles available at the depot n_v varies between 2 and 4; the vehicle capacity Q varies between 3 and 27. The authors generated a set of 240 instances, which is available at `https://doi.org/10.17632/p4n2xw84bv.1`.

Results obtained

On each instance of this set, the algorithm was run ten times using the set of parameters defined by Table A.6. The branch-and-cut algorithm of Rodríguez-Martín et al. [36] was run on a personal computer with an Intel Core i7 CPU at 3.4 GHz and 16 gigabytes of RAM, with an imposed time limit on their method of 7200 seconds. Since our machine was about as fast as theirs, we applied the same time limit to Algorithm `HMPG` setting the maximum CPU time CPU_{max} to 7200 seconds. Further, since only capacity constraints are present, parameters α_1 and α_2 were set to 0, whereas the value of α_3 was set to 50.

Table 2 summarises a comparison of the results reported by Rodríguez-Martín et al. [36] and the results obtained by Algorithm `HMPG`. In Table 2, the instances are grouped by number of customers. For each group of instances, the table gives the number of instances in the group ("$\#ist$") and compares our algorithm to the exact method of Rodríguez-Martín et al. separating the results obtained by the instances solved to optimality and the remaining set of instances (for which optimality was not proved). For each group of instances and each method, the table shows the number of instances solved to optimality ("$\#opt$") and the corresponding average computing time in seconds ("t"). For Algorithm `HMPG`, the table also shows the average percent deviation ("$\%dev$") of the solutions not solved to optimality. The percent deviation of a value x with respect to a reference value y is computed as $100 \times \frac{x}{y}$.

For the instances not solved to optimality by the branch-and-cut, the table reports the average percent deviation ("$\%dev$") from the best known solution cost (computed either by the branch-and-cut or Algorithm `HMPG`) and the corresponding average computing time in seconds ("t") for each method. The comparison considers only the instances for which the branch-and-cut was capable of computing feasible solutions (228 out of 240 instances).

Table 2 shows that Algorithm `HMPG` computed optimal solutions for almost all the instances solved to optimality by the branch-and-cut method

Table 2: Comparison with the exact method of Rodríguez-Martín et al. [36].

		Solved to optimality					**Optimality not proved**			
		Branch-and-cut		HMPG			Branch-and-cut		HMPG	
n_c	$\#ist$	$\#opt$	t	$\#opt$	$\%dev$	t	$\%dev$	t	$\%dev$	t
10	24	24	2.0	24		0.4				
20	34	34	75.5	34		20.1				
30	36	36	261.9	36		80.5				
40	33	27	862.0	26	101.2	154.3	100.0	7200.0	100.8	1409.2
50	36	22	770.7	21	102.0	371.5	100.2	7200.0	100.2	1148.7
60	34	19	1762.8	17	101.0	715.4	100.7	7200.0	100.8	1692.3
70	31	17	2036.5	14	101.2	874.6	100.8	7200.0	100.3	1679.4
	228	179		172						

(172 compared to 179 instances). Moreover, Algorithm HMPG shows an average percent deviation on the remaining seven instances equal to 101.4%, which means that high-quality solutions can also be computed for these instances. Of the instances not solved to optimality by the branch-and-cut method, Algorithm HMPG is capable of computing solutions close to those computed by the branch-and-cut. It is worth noting that on the group of larger instances with 70 customers, Algorithm HMPG computed (on average) better solutions than the branch-and-cut. Finally, the average computing times of our algorithm are highly competitive with respect to the average computing times of the branch-and-cut.

Appendix B gives the complete details about the results summarised in Table 2. The detailed results also show that Algorithm HMPG was capable of computing feasible solutions for six instances for which the branch-and-cut failed to compute a feasible solution. Over the ten runs, the differences among the solution costs computed are limited. Further, Algorithm HMPG takes full advantage of the total computing time spent as shown by the average computing times necessary to compute the best solutions found.

4.2. Results on the real-world instances

In this section, we describe the results obtained by Algorithm HMPG on real-world instances, we investigate the effectiveness of the different algorithm components, and we present sensitivity analyses.

Table 3: Type of day combinations.

Freq./Day	Mon.	Tue.	Wed.	Thu.	Fri.	Sat.
2	0	1	0	0	1	0
	0	1	0	0	0	1
	0	0	1	0	0	1
	0	1	0	1	0	0
	1	0	0	1	0	0
	0	0	1	0	1	0
3	1	0	1	0	1	0
	0	1	0	1	0	1
4	1	0	1	0	1	1
	0	1	0	1	1	1
	0	1	1	1	0	1
	1	1	0	1	1	0
	1	1	1	0	1	0
	1	0	1	1	1	0
5	1	1	1	0	1	1
	0	1	1	1	1	1
	1	1	1	1	0	1
	1	0	1	1	1	1
	1	1	0	1	1	1

Test instances

The data for the instances were prepared and provided by the dairy company that motivated our study. Four different regions (A, B, C, and D) corresponding to the distribution areas of four transit points or depots were selected by the company for the computational study. A specific set of customers is associated with each region and must be served by the region's transit point. In particular, all the depots and corresponding customer locations have been georeferenced using a digital map; the map was used to compute distances and travel times by defining the travelling speeds on the different types of roads, depending on the type of vehicle used for the distribution. For the sake of our preliminary experiments, small-size instances were also generated by the company. The data have the following characteristics.

- Both urban and rural areas are represented by the locations.
- The number of customers ranges from 25 to 1200.
- The planning period is from Monday to Saturday, i.e., $n_d = 6$.

- All the data associated with the customers were defined based on the company's data. The frequencies range from 1 to 6. Table 3 gives the day-combinations to be considered, based on the frequency values. Further, customers with a visit frequency equal to 1 can be served any day of the planning period. A customer with a frequency less than or equal to 4 receives the same demand quantity on any of its visits, whereas for customers having frequencies equal to 5 or 6, the last day (i.e., Saturday) is generally associated with a higher demand than the weekdays. Regarding the time windows, the customers have early opening times, so waiting times at the customers are generally not an issue.

- Only one type of vehicle is used, with a capacity equal to 1450 kilograms ($Q = 1450$) and a maximum working time of 420 minutes ($L = 420$). The weekly cost F associated with each area is set to 1000.

- The cost matrix c_{ij} was defined based on the real routing costs of the vehicles.

A total of 36 representative instances were prepared by the company. The instances are grouped by size ("S" for small, "M" for medium, or "L" for large) and by the corresponding depot (A, B, C, or D).

Results obtained

Algorithm `HMPG` was run ten times on each instance using the parameter values given in Table A.6. For the real-world instances, the maximum CPU time (seconds) CPU_{max} of Algorithm `HMPG` depended on the number of customers and ranges from 1200 to 10800 seconds. Tables 4 and 5 summarise the results obtained.

Table 4 gives detailed information about the structure of the best solutions computed by the algorithm for medium- and large-size instances. For each instance, the table reports the total solution cost ("z"), the routing cost ("rc"), the area cost ("ac"), and the number of areas ("$\#ar$"). Further, the table shows information about the average number of customers and routes per area ("$\#avg$ cust." and "$\#avg$ routes"). Information about the maximum utilisation of the vehicles in terms of capacity and time are also reported in the table ("$UT(Q)$" and "$UT(T)$"). The capacity (time) utilisation of a

Table 4: Details about the solutions of real-world instances.

Name	n_c	z	rc	ac	$\#ar$	$\#avg$ cust.	$\#avg$ routes	$UT(Q)$	$UT(L)$
MA-01	200	10955.7	2955.7	8000.0	8	25.0	6.0	0.74	0.83
MA-02	400	19917.8	4917.8	15000.0	15	26.7	6.0	0.61	0.82
MA-03	600	28308.9	6308.9	22000.0	22	27.3	6.0	0.58	0.93
LA-01	800	32545.2	7545.2	25000.0	25	32.0	6.0	0.62	0.96
LA-02	1200	50710.1	10710.1	40000.0	40	30.0	6.0	0.57	0.84
MB-01	200	13922.7	3922.7	10000.0	10	20.0	6.0	0.69	0.67
MB-02	400	23099.2	6099.2	17000.0	17	23.5	5.8	0.47	0.64
MB-03	600	30326.1	7326.1	23000.0	23	26.1	5.8	0.52	0.69
LB-01	800	35762.1	8762.1	27000.0	27	29.6	6.0	0.54	0.70
LB-02	1200	76904.3	22904.3	54000.0	54	22.2	5.4	0.65	0.73
MC-01	200	13779.8	4779.8	9000.0	9	22.2	6.0	0.96	0.88
MC-02	400	19367.1	6367.1	13000.0	13	30.8	5.8	0.91	0.94
MC-03	600	23441.8	7441.8	16000.0	16	37.5	6.0	0.68	0.97
LC-01	800	30374.2	10374.2	20000.0	20	40.0	5.9	0.68	0.91
LC-02	1200	69793.2	21793.2	48000.0	48	25.0	5.9	0.62	0.83
MD-01	200	6004.8	1004.8	5000.0	5	40.0	6.0	0.72	0.70
MD-02	400	15387.6	3387.6	12000.0	12	33.3	6.0	0.59	0.81
MD-03	600	22424.8	5424.8	17000.0	17	35.3	6.0	0.75	0.80
LD-01	800	28904.4	6904.4	22000.0	22	36.4	6.0	0.68	0.91
LD-02	1200	52629.5	14629.5	38000.0	38	31.6	5.7	0.87	0.89

route is computed as the ratio between the total vehicle demand (total working time) of the route and the vehicle capacity Q (maximum working time L).

Table 5 under section `HMPG` reports the results obtained for Algorithm `HMPG`. For each instance, the table reports the instance name and the corresponding number of customers, the cost of the best solution found, the percent deviation ("$\%dev$") of the best solution found over the ten runs, and the average percent deviation of the solutions corresponding to the ten runs ("$\%dev_a$"). Column t reports the average computing time of the ten runs in seconds. The remaining sections of the table are about the effectiveness of the different components of `HMPG`, as described in the following.

The results of Table 5 show that Algorithm `HMPG` reached the imposed time limit for several instances. These instances represent challenging territory design instances. Over the ten runs, `HMPG` performs on average quite well, as shown by column $\%dev_a$. It is worth considering different runs of the algorithm with the aim of computing improved solutions. Because the problem with these instances is tactical, the corresponding running times are compatible with the planning operations of the company.

Table 5: Results of Algorithm HMPG on real-world instances and the effectiveness of its components.

			No-FI			**No-MP**			**No-SA**			**No-DynPop**			**No-DM**			HMPG-**Lit**			HMPG		
Name	n_c	z^*	$\%dev$	$\%dev_a$	t	$\%dev$	$\%dev_a$	t	$\%dev$	$\%dev_a$	t	$\%dev$	$\%dev_a$	t	$\%dev$	$\%dev_a$	t	$\%dev$	$\%dev_a$	t	$\%dev$	$\%dev_a$	t
SA-01	25	1525.7	**100.0**	100.3	89.6	**100.0**	100.5	59.5	**100.0**	100.6	154.6	**100.0**	100.6	378.2	**100.0**	100.7	118.3	100.3	101.2	29.5	**100.0**	100.1	60.5
SA-02	50	2782.0	101.6	102.2	227.9	102.5	102.5	235.7	100.8	101.6	430.9	101.1	101.5	291.3	100.3	101.2	413.4	101.5	102.2	52.2	**100.0**	101.2	373.5
SA-03	75	2901.3	101.9	102.6	800.6	102.0	103.0	453.6	100.9	101.5	930.7	100.4	102.3	728.4	101.0	101.5	535.4	100.5	102.3	474.9	**100.0**	101.4	623.0
SA-04	100	4162.2	131.0	134.1	1200.0	102.6	103.3	763.8	**100.0**	102.4	701.7	101.6	102.1	610.9	102.3	103.0	829.9	103.3	104.7	782.7	**100.0**	102.8	691.4
SB-01	25	1336.2	**100.0**	100.1	66.2	**100.0**	100.3	40.5	**100.0**	100.3	89.4	**100.0**	101.1	31.6	**100.0**	100.2	144.5	100.0	100.2	44.4	**100.0**	100.0	29.9
SB-02	50	4676.4	100.4	101.1	447.1	101.9	113.2	195.5	101.0	114.6	685.1	100.6	101.0	826.3	**100.0**	116.6	344.8	115.0	116.2	55.6	**100.0**	101.0	547.1
SB-03	75	6158.1	105.1	113.9	648.9	101.6	102.8	524.3	101.9	108.3	671.7	102.3	114.3	1039.5	102.3	117.0	1159.8	118.0	118.9	370.1	**100.0**	101.3	1101.4
SB-04	100	7854.8	110.7	111.4	1200.0	**100.0**	109.7	891.3	110.6	114.3	1200.0	110.1	110.6	872.7	109.9	111.2	1200.0	110.6	111.8	1195.4	**100.0**	110.1	1157.2
SC-01	25	1588.3	100.0	100.2	139.4	100.0	100.6	89.3	100.0	100.1	157.6	100.9	141.1	129.5	100.0	101.0	55.5	100.1	147.4	90.8	**100.0**	100.1	79.4
SC-02	50	4553.3	100.4	101.5	551.5	101.2	101.9	475.1	101.2	102.2	444.5	100.1	100.8	485.1	100.5	101.0	633.8	101.2	101.5	230.6	**100.0**	100.5	502.7
SC-03	75	6063.7	102.2	102.8	1200.1	106.7	107.4	1019.1	100.4	102.7	1085.4	100.0	103.0	1219.7	100.0	104.1	1168.3	102.4	116.1	920.3	**100.0**	101.0	1150.9
SC-04	100	7670.4	101.2	106.5	1200.0	106.7	103.6	1035.8	102.8	103.8	1200.5	100.1	102.2	1135.1	101.2	107.8	1113.3	117.0	117.9	666.8	**100.0**	101.6	1161.4
SD-01	25	1280.0	**100.0**	100.1	39.8	**100.0**	100.2	25.1	**100.0**	100.2	96.9	**100.0**	100.3	217.6	**100.0**	100.3	38.0	100.1	100.2	30.2	**100.0**	100.1	58.7
SD-02	50	2515.4	100.3	100.2	341.2	100.3	100.4	152.6	100.2	100.2	1018.4	**100.0**	100.1	244.7	**100.0**	100.1	302.4	100.3	100.5	63.5	**100.0**	100.1	181.0
SD-03	75	2611.0	100.8	101.4	963.0	101.9	102.5	397.1	101.0	101.6	1140.5	100.4	101.3	671.5	100.9	101.4	782.8	109.7	110.7	233.6	**100.0**	100.3	467.8
SD-04	100	3702.6	100.2	100.6	1097.9	100.5	101.0	768.8	100.3	100.6	1134.4	100.2	100.8	1106.4	**100.0**	100.4	943.8	100.1	100.7	332.7	**100.0**	100.4	887.6
MA-01	200	10955.7	112.8	115.9	3600.0	**100.0**	107.3	2394.8	106.3	108.3	3600.0	100.6	111.5	3600.0	111.4	115.7	3600.0	113.9	119.4	3600.0	**100.0**	106.5	3600.0
MA-02	400	19917.8	122.8	124.8	3600.0	112.4	114.6	3600.0	121.5	124.3	3600.0	111.9	118.5	3600.0	119.7	123.9	3600.0	109.5	115.4	3165.7	**100.0**	109.9	3600.0
MA-03	600	28308.9	109.9	116.6	4800.0	101.8	122.4	4800.0	102.5	113.9	4800.0	108.6	117.3	4800.0	105.9	110.9	4800.0	**100.0**	110.0	4800.0	**100.0**	103.4	4800.0
MB-01	200	13922.7	101.9	102.0	3600.0	**100.0**	101.9	3162.5	101.4	106.8	3600.0	101.4	107.1	3600.0	102.4	105.2	3600.0	101.6	108.1	3518.8	**100.0**	101.5	3600.0
MB-02	400	23099.2	110.3	118.0	3600.0	106.3	110.7	3600.0	101.3	108.2	3600.0	105.2	110.1	3600.0	109.2	115.0	3600.0	**100.0**	110.5	3600.0	**100.0**	108.2	3600.0
MB-03	600	30326.1	122.7	124.3	4800.0	110.1	113.0	4800.0	111.4	114.8	4800.0	112.7	117.5	4800.0	118.3	122.6	4800.0	105.7	109.9	4800.0	**100.0**	106.7	4800.0
MC-01	200	13779.8	106.1	110.7	3600.0	100.2	107.1	3600.0	108.5	109.6	3600.0	102.5	108.1	3600.0	107.8	109.7	3600.0	117.2	124.8	3261.5	**100.0**	101.2	3600.0
MC-02	400	19367.1	124.3	140.4	3600.0	103.5	106.2	3600.0	107.7	112.9	3600.0	108.9	120.2	3600.0	115.6	122.2	3600.0	106.4	108.1	3600.0	**100.0**	105.6	3600.0
MC-03	600	23441.8	-	-	4800.0	121.1	136.8	4800.0	127.8	161.1	4800.0	110.9	154.1	4800.0	-	-	4800.0	121.6	135.0	4800.0	**100.0**	110.9	4800.0
MD-01	200	6004.8	100.4	100.6	3224.7	100.2	100.4	1559.3	100.5	100.9	2899.2	100.6	101.1	2405.4	**100.0**	100.4	3600.0	100.2	100.8	1187.2	**100.0**	100.3	2339.2
MD-02	400	15387.6	111.3	112.4	3600.0	102.2	102.8	3600.0	104.0	114.0	3600.0	103.0	111.3	3600.0	**100.0**	113.3	4800.0	109.6	112.8	4014.7	**100.0**	104.4	4800.0
MD-03	600	22424.8	104.3	111.4	4800.0	105.9	112.2	4800.0	107.0	118.8	4800.0	101.9	108.6	4800.0	109.8	120.5	4800.0	104.0	109.5	4800.0	**100.0**	102.1	4800.0
LA-01	800	32545.2	119.4	123.0	7200.0	113.1	126.7	7200.0	109.0	127.7	7200.0	122.0	129.1	7200.0	122.3	137.5	7200.0	106.0	112.8	6150.9	**100.0**	108.4	7200.0
LA-02	1200	50710.1	118.6	126.7	10800.0	112.9	118.7	10800.0	113.6	119.6	10800.0	112.9	118.7	10800.0	120.0	124.2	10800.0	114.6	120.0	10800.0	**100.0**	105.8	10800.0
LB-01	800	35762.1	132.7	145.6	7200.0	134.8	142.4	7200.0	127.6	143.6	7200.0	131.1	133.0	7200.0	132.7	141.2	7200.0	113.3	117.8	7200.0	**100.0**	109.6	7200.0
LB-02	1200	76904.3	123.3	155.1	10800.0	107.6	151.8	10800.0	112.0	129.8	10800.0	119.1	149.5	10800.0	129.9	144.3	10800.0	125.0	148.6	10800.0	**100.0**	122.6	10800.0
LC-01	800	30374.2	-	-	7200.0	114.5	140.1	7200.0	119.1	147.7	7200.0	114.4	148.3	7200.0	108.4	161.9	7200.0	122.1	141.0	6122.7	**100.0**	133.3	7200.0
LC-02	1200	69793.0	117.3	133.7	10800.0	102.6	103.8	10800.0	115.3	124.0	10800.0	102.6	113.7	10800.0	105.1	123.6	10800.0	116.3	126.0	10800.0	**100.0**	102.9	10800.0
LD-01	800	28904.4	111.9	124.6	7200.0	117.4	126.7	7200.0	131.7	136.9	7200.0	112.9	126.1	7200.0	111.3	120.3	7200.0	109.5	114.8	7107.6	**100.0**	111.5	7200.0
LD-02	1200	52629.5	129.2	153.5	10800.0	108.8	135.7	10800.0	116.0	150.3	10800.0	118.1	136.1	10800.0	119.5	146.9	10800.0	113.7	114.8	10800.0	**100.0**	118.1	10800.0
			109.9			105.6			107.4			106.1			107.6			108.1			100.0	105.4	

Table 4 shows that the number of areas in the solutions for large-scale instances ranges from 20 to 54. The distribution regions of the different depots are indeed geographically heterogeneous, with both urban and rural areas, as indicated by the wide range of routing costs as well. In the context of this study, the main purpose was to serve all the customers with a minimum total distribution cost. However, serving geographically dispersed customers can be particularly costly for the company since dedicated routes serving a few customers on specific days must be required. In terms of capacity and time utilisation of the vehicles, the data show that the most critical resource is time, as shown by the maximum utilizations reported by columns "$UT(Q)$" and "$UT(T)$". Indeed, service and travel times play a crucial role in the distribution. It is worth noting that all the drivers are employed by the company. If the driver completes the route before the end of his working hours, he must complete the working day at the depot from which the journey started.

Impact of the main HMPG components

To analyse the impact on performance of the various algorithmic components of the proposed metaheuristic, we perform experiments on HMPG by removing each of them in turn. In particular, as mentioned in Section 3, Algorithm HMPG also introduces new features with respect to the algorithms proposed by Vidal et al. [17] and Zhou et al. [39]. Hence in this section we also evaluate the effectiveness of the new features.

We evaluate the following different versions of HMPG.

1. "No-FI". The feasibility improvement procedure described in Section 3.6.4 is not used.
2. "No-MP". Only one subpopulation is considered; i.e., a no multi-population strategy is adopted.
3. "No-SA". A classical local search is used instead of the SA algorithm described in Section 3.6. The local search terminates whenever no improvements can be found.
4. "No-DynPop". The maximum size of the subpopulation used to activate the survivor selection mechanism is not dynamically changed (see Section 3.7.2).
5. "No-DM". The diversification mechanism from different subpopulations (lines 12-16 of Algorithm 1) is disabled.

6. "No-MP", "No-SA", "No-DynPop", "No-DM" (denoted as HMPG-Lit). Furthermore, in this version, we consider only one feasible subpopulation (containing feasible solutions only) and, in addition, we also consider one *infeasible* subpopulation (containing infeasible solutions only).

The last of the six versions listed above has been introduced to simulate the solution framework proposed by Vidal et al. [17], and thus to compare the effectiveness of the new components introduced to solve our problem with respect to the literature.

Table 5 shows the results obtained by the different variants of HMPG using the same notation described above. In the table, the percentage value in bold reported under column "%*dev*" indicates that the corresponding variant computed a solution having the best solution cost known for the associated instance. An entry in the table marked with a symbol "-" means that no feasible solutions were computed by the corresponding method.

The results show that each algorithm component plays an important role in the good overall performance of the HMPG algorithm. As shown by the average percentage values reported in the last line of the table, the new features of our algorithm—namely the feasibility improvement procedure, the SA, and the diversification mechanism—greatly contribute to its effectiveness. Further, in the set of small-sized instances "S" for the variant "No-DynPop", the table shows an average computing time of 624.3 seconds and an average deviation of 101.2%, whereas HMPG shows an average computing time of 567.1 seconds and an average deviation of 100.0%. Hence, the mechanism used to dynamically change the maximum sizes of the subpopulations is also quite effective. Finally, version HMPG-Lit was not able to obtain the same best solutions of HMPG. Algorithm HMPG-Lit computed best solutions only for two instances.

4.3. Sensitivity analyses

In this section, we perform different sensitivity analyses on important components of the problem in order to generate useful managerial insights for effective tactical decision making.

It is worth noting that, based on the details reported in Table 4, the vehicle capacity is not a critical component. Moreover, the maximum working time L is defined based on existing labor laws and regulations, hence investigating alternative scenarios based on the value of L would involve decisions

not of interest to the company. For this reason, we do not perform analyses on the values of the vehicle capacity and of the maximum working time L.

We start analysing the impact of the fixed area cost F and the time window constraints. Then, we focus our attention on the level of the service quality to the customers. We conclude this section with an analysis of the impact of the service requirements, based on the frequency of service and the day-combinations.

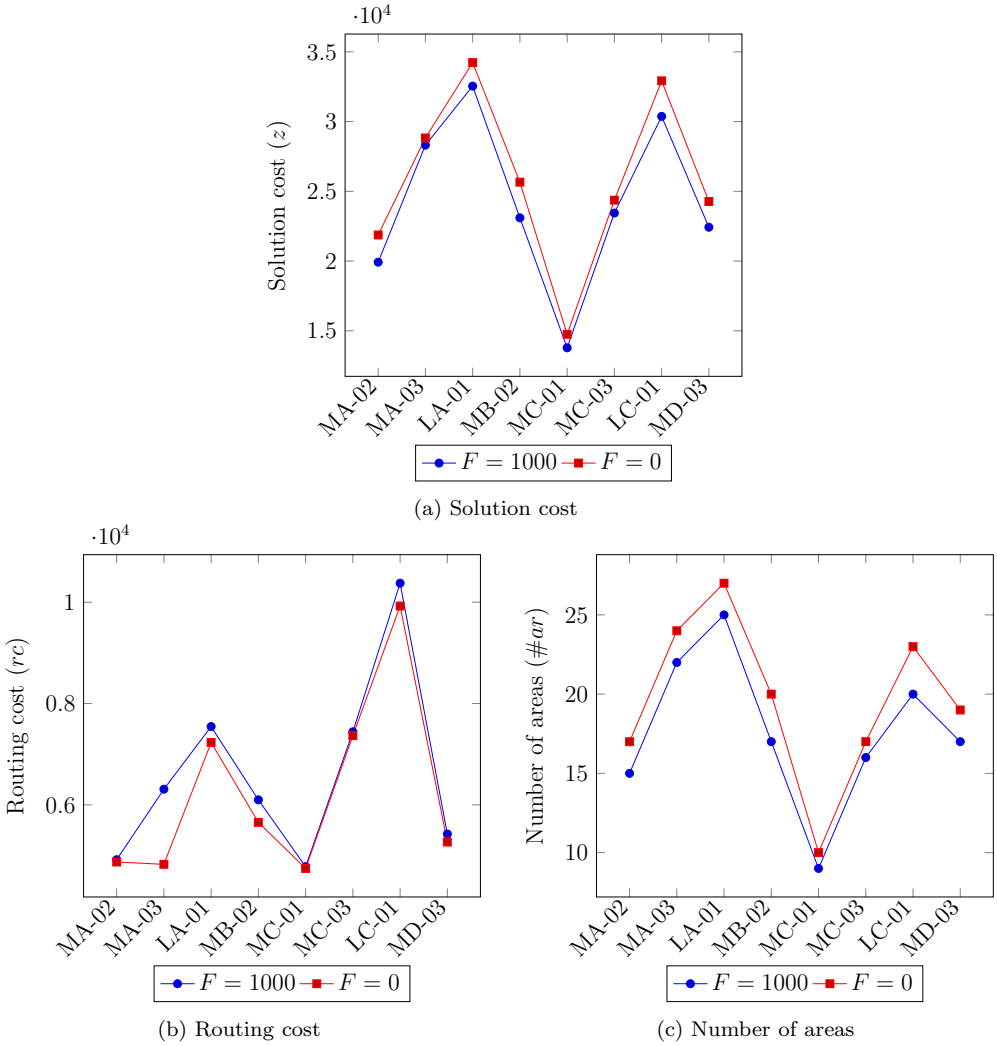
The results in this section involve a subset of the whole set of real-world instances identified with the company's operators. Moreover, for each instance evaluated in the experiments, Algorithm `HMPG` is run ten times and the best solution found over the ten runs is selected as the corresponding solution. We use the same set of parameters for `HMPG` defined by Table A.6, that were used in the previous experiments.

Sensitivity analysis - Fixed cost and time window widths

The first set of experiments evaluates the impact of the fixed cost F used in the objective function of the problem. As mentioned previously, the value of F in our experiments (e.g., 1000) was defined by the company's operators as the marginal weekly cost for an area or a vehicle. In these experiments, eight medium- and large-size instances were selected from the whole set. The solutions when F is set to 1000 are compared with the solutions when F is set to 0, i.e., when the objective function minimises the total routing cost disregarding the fixed costs. The solutions when $F = 0$ provide lower estimates of the routing costs for all eight instances considered, being the objective function the pure minimization of the total routing cost.

Figure 7 depicts the results obtained; the corresponding detailed results can be found in Appendix C. The three figures (a), (b), and (c) show the *real* solution costs (z), the routing costs (rc), and the number of areas ($\#ar$), respectively, for the two cases. The solution cost z corresponds to the real cost of the solution, therefore it also includes the fixed cost of 1000 in both scenarios where $F = 1000$ and $F = 0$.

Figure 7-a shows that the setting $F = 1000$ always produces better solutions than $F = 0$ (i.e., the real solution cost z is smaller), and provides the right balance of the number of areas (see Figure 7-c). As expected, the routing costs when $F = 0$ are always lower, but a trade-off can be found—as shown by the results of instance MA-03: the gap between the two solution costs is only about 1.8%. In this solution, two more areas are used when $F = 0$, but at a considerably lower routing cost (less than 23% of the solu-



(a) Solution cost

(b) Routing cost

(c) Number of areas

Figure 7: Sensitivity analysis on the value of F.

tion when $F = 1000$). This suggests that there are alternatives for servicing the specific territory, at (more or less) the same cost.

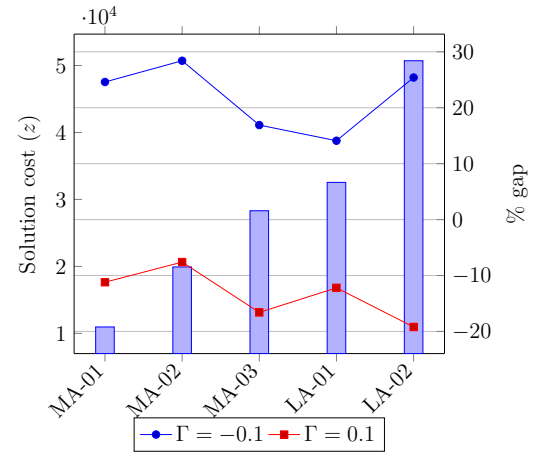
To perform a sensitive analysis on the impact of the time windows, ten medium- and large-size instances are selected from the groups of instances A and D. We consider different scenarios by varying the time window widths: for each customer $i \in N$ and day $d \in D$, we first compute $\Delta = l_i^d - e_i^d$. Then we define the new time window for customer i on day d to be $[\max\{0, e_i^d - \Gamma\Delta\}, \min\{l_i^d + \Gamma\Delta, 1440\}]$ (in our instances, the lower limit 0 and the upper limit 1440 are never reached). Two different values of Γ were considered: (i) $\Gamma = -0.1$, i.e., the time window is narrowed and (ii) $\Gamma = 0.1$, i.e., the time window is widened.

Figures 8 and 9 show the results obtained over the ten instances. The different figures report the solution costs, the routing costs, the number of areas in each instance, and the percent gaps of the two scenarios considered (i.e., $\Gamma = -0.1$ and $\Gamma = 0.1$). The corresponding detailed results can be found in Appendix C. The percent gap of a value x with respect to a reference value y is computed as $100 \times \frac{(x-y)}{y}$.

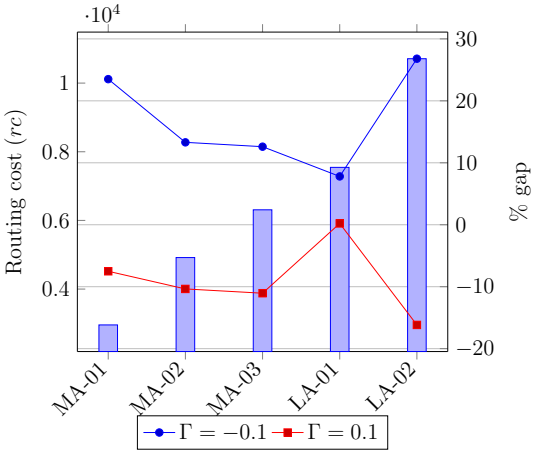
The results show that significant savings can be obtained by enlarging the time window widths; comparably, reducing the time windows greatly increases the total cost (e.g., up to about 40% for instance LD-02). This difference makes it clear that the time window constraints represent a critical component of the level of service provided to the customers. The results indicate territories which are particularly sensitive to variations in the time windows (e.g., see instances MA-02 and MD-02), as evidenced by the changes in routing costs and number of areas. Given that customers are assigned to specific depots or transit points, the results of this analysis can also reveal critical territories (and the corresponding groups of customers) that should be revised to improve the assignments of customers to depots.

Notice that in our application the time windows are defined by the customers, based on their activities and daily operations. In most cases, the margins for negotiating their definitions are very limited.

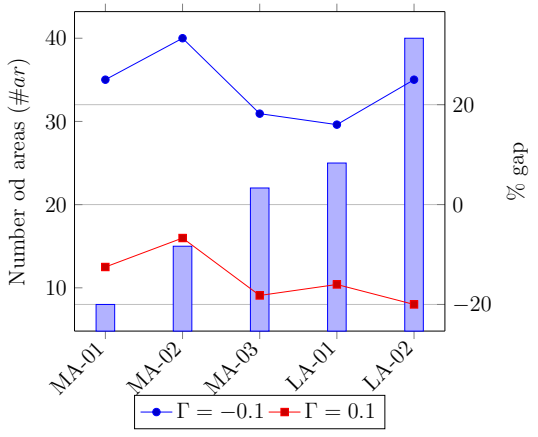
To further analyse the results obtained, Figure 10 gives an overview of the customers distributions of the two larger instances LA-02 and LD-02. The figure show the distributions of the customers in term of the distances (in kilometers) from the depot locations. For instance LA-02 about 99% of the customers are within a distance of 40 kilometers from the depot whereas for instance LD-02 the percentage is about 94, and that the LD-02 instance also contains customers which are about 100 kilometers far from the main depot.



(a) Solution cost

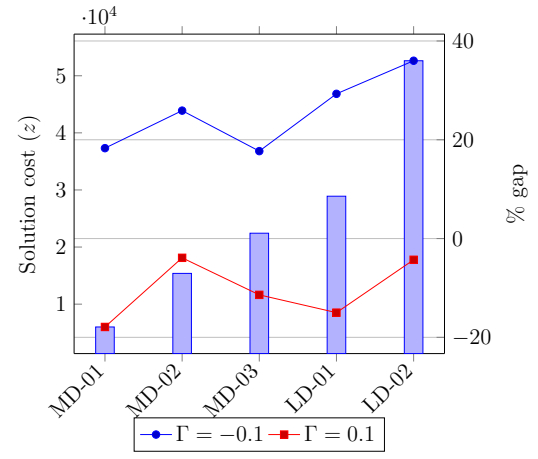


(b) Routing cost

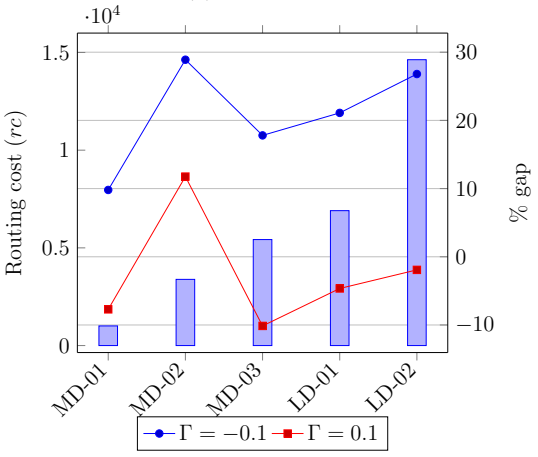


(c) Number of areas

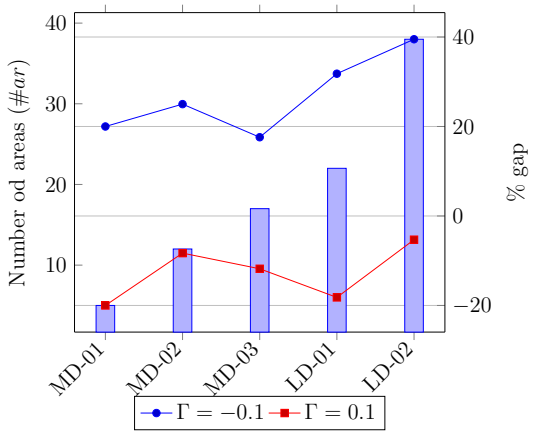
Figure 8: Sensitivity analysis on the time window width (group A).



(a) Solution cost



(b) Routing cost



(c) Number of areas

Figure 9: Sensitivity analysis on the time windows width (group D).

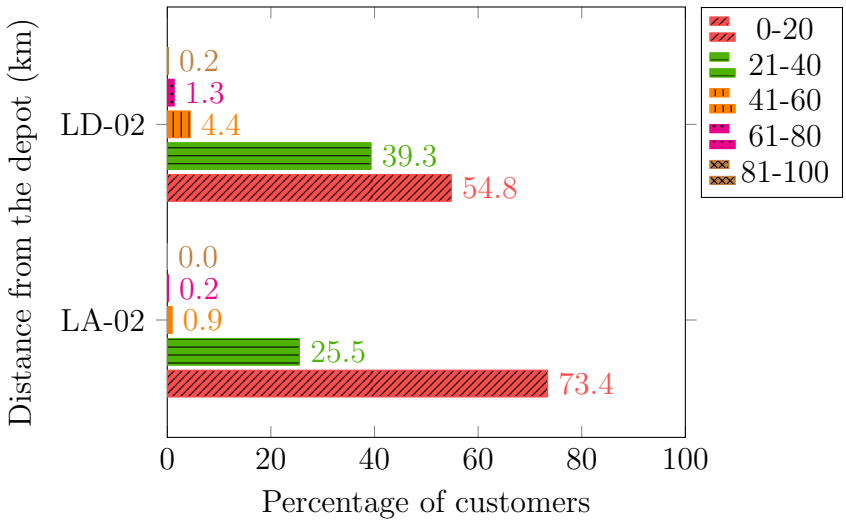


Figure 10: Distribution of the customer distances from the main depots (instances LD-02 and LA-02)

In practice, instance LD-02 involves also rural customers far from the depot location. This distributions could explain the fact that for the case $\Gamma = -0.1$ (i.e., the time windows are narrowed), the increases of the solution costs are about 36% and 25% for instances LD-02 and LA-02, respectively. In other words, for a set of customers which are spread over a larger geographical area, improving the quality of service by ensuring tight delivery windows results in a more sensitive increase of the distribution cost.

Sensitivity analysis - Customer frequencies

As aforementioned, the definition of the time window depends on the customer's preferences, hence it is very difficult to reduce the distribution cost by revising their values. Another parameter that greatly impacts the efficiency of a distribution plan is the frequency of visits; however, in this case the company has larger decision margins in their definition. Thus cost savings can potentially be achieved by changing the customers' visit frequencies.

In this section, we describe a new metric, the *Marginal Routing Cost* (MRC), associated with the set of customers and used to measure the marginal impact on the total routing cost of each customer. We propose a method based on the MRC which selectively revises the visit frequencies with the aim of reducing the total distribution cost. Below, we describe the MRC and the method used to revise the frequencies, followed by an analysis

of its effectiveness.

Given an instance of the problem, let $\mathcal{R} = \{\ell_1, \ldots, \ell_k\}$ be the index set of k routes forming a feasible solution with cost z involving v areas. For a route $\ell \in \mathcal{R}$, b_ℓ is its routing cost and $\pi(\ell)$ is the day of the planning period when the route is operated. For each $i \in N$, the MRC u_i is defined as follows:

$$u_i = \frac{1}{f_i} \sum_{\ell \in \mathcal{R}_i} q_i^{\pi(\ell)} \frac{b_\ell}{Q_\ell}, \tag{5}$$

where $\mathcal{R}_i \subseteq \mathcal{R}$ is the index set of the routes in solution $\mathcal{R}$ serving customer i, and Q_ℓ is the total demand of route ℓ. We have $|\mathcal{R}_i| = f_i$ and $z = vF + \sum_{\ell \in \mathcal{R}} b_\ell = vF + \sum_{i \in N} f_i u_i$, hence u_i represents a marginal routing cost per visit for servicing customer i.

Customers associated with high u_i values represent critical customers in terms of the distribution cost; savings can potentially be achieved by revising their frequency values. Thus we adopt a simple procedure to revise the frequencies based on the MRC values. The procedure works as follows.

(i) The MRC u_i are sorted for decreasing values; $(u_{i_1}, u_{i_2}, \ldots, u_{i_{n_c}})$ is such that $u_{i_1} \geq u_{i_2} \geq \cdots \geq u_{i_{n_c}}$;

(ii) We consider the first $h = \Upsilon\, n_c$ customers having the highest u_i values. Let $W = \{i_1, i_2, \ldots, i_h\}$ be the index set of these customers;

(iii) For each customer $i \in W$ with $f_i > 1$ we execute the following steps.

  (a) We recompute the customer frequency as $\overline{f}_i = f_i - 1$.

  (b) We associate the set of day-combinations as defined by Table 3 to the new frequency value $\overline{f}_i$.

  (c) We compute

$$\overline{q}_i = f_i \frac{\min_{d \in D: q_i^d > 0}\{q_i^d\} + \max_{d \in D}\{q_i^d\}}{2}, \tag{6}$$

  an estimate of the demand that must be delivered to the customer at each visit. For our set of instances, values $\overline{q}_i$ result in upper estimates on the total demand delivered by the routes in $\mathcal{R}$ with the original frequency f_i, so the customer frequency is reduced but the original level of service (in terms of total demand) is preserved.

  (d) We compute $\overline{q}_i^d = \lceil \overline{q}_i / \overline{f}_i \rceil$, $d \in D$.

(iv) We associate, with each customer $i \in N$, the new frequency value $\overline{f}_i$ and demands $\overline{q}_i^d$. Based on the new frequency value $\overline{f}_i$ and associated

day-combinations, we assume $\overline{q}_i^d = 0$ if the customer cannot be served on day d. The time windows of the customers are left unchanged, hence the same level of service is preserved regarding time windows as well.

To fully evaluate the impact of the above procedure, a solution method which provides optimal or close-to-optimal solutions is required. We select four small-size instances from group D (instances involving up to 100 customers), for which Algorithm `HMPG` has provided close-to-optimal solutions in the computational study reported in Section 4.1.

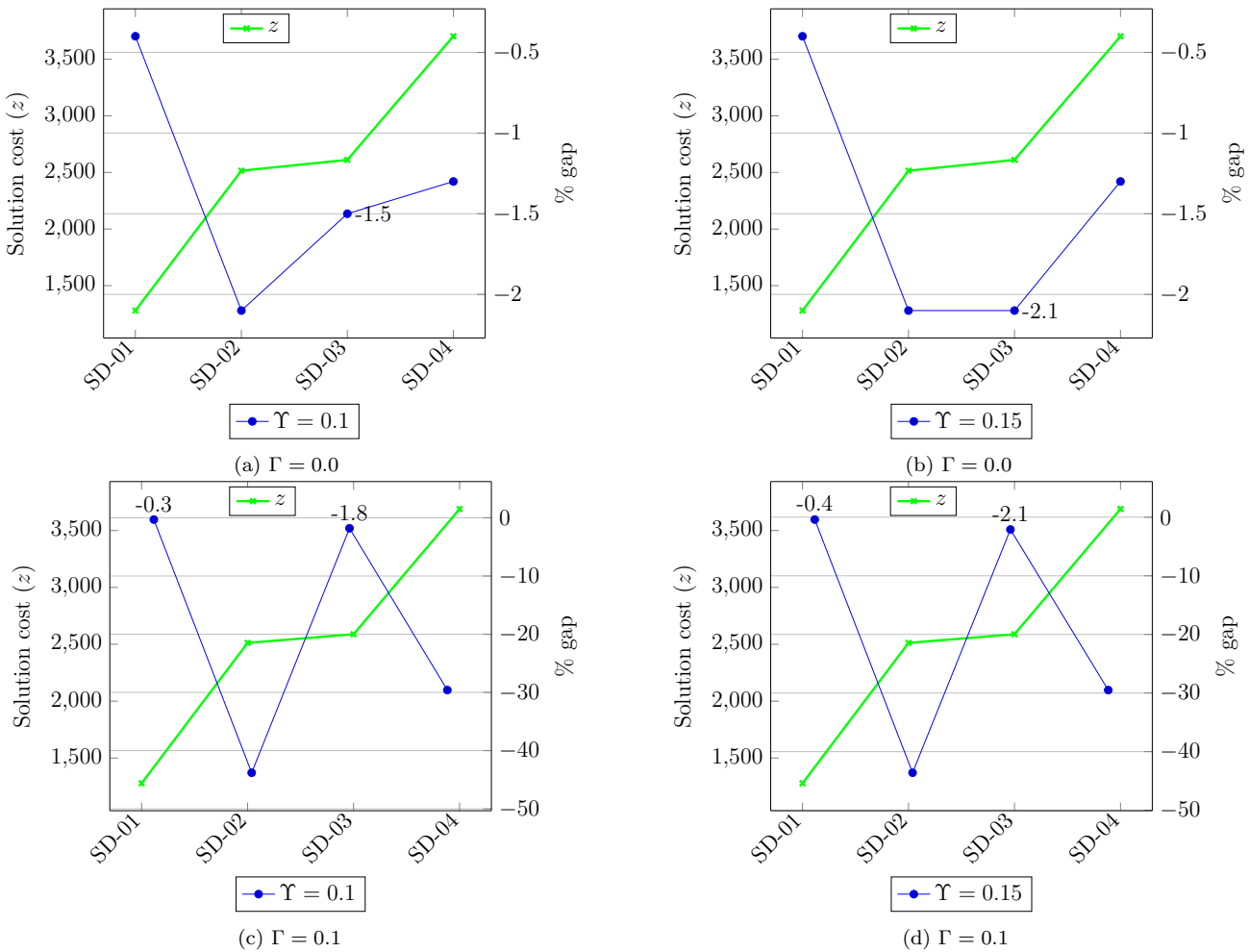
Figure 11 reports the results obtained with $\Upsilon \in \{0.1, 0.15\}$ (i.e., the visit frequencies are revised for either 10% or 15% of the customers). Detailed results can be found in Table C.17 in Appendix C. Figures (a) and (b) give the results with $\Gamma = 0.0$; Figures (c) and (d) show the results with $\Gamma = 0.1$ (the time window widths of all the customers are modified as described in the previous section). The figures report the solution costs (z) and the percent gaps of the different scenarios.

Figure 11-a shows that cost savings (up to about 2%) can be achieved for all the instances. Increasing the value of Υ to 0.15 achieves an additional savings for instance SD-03. Clearly, savings achieved on a weekly basis can lead to consistent savings over a medium-term planning horizon. The results with $\Gamma = 0.1$ show that combining changes to the customer frequencies and the time window widths leads to additional savings, by also reducing the number of areas used in the solutions.

Finally, with respect to our previous analysis, the impact of the changes on the data associated with the customers can be gradually tailored by the value of Υ. Moreover, the definition of the set W can be driven by the expert knowledge of the company's operators, making the changes even more effective in practice.

5. Conclusions and future work

In this paper, we investigated a real-world territory design problem in the dairy industry that requires the joint optimisation of distribution districts and periodic vehicle routes. For its solution, we designed a heuristic algorithm based on a hybrid genetic search solution framework enriched with new features, such as a local search improvement procedure based on a simulated annealing algorithm framework and a mechanism to dynamically change the maximum subpopulation sizes.



(a) $\Gamma = 0.0$ (b) $\Gamma = 0.0$

(c) $\Gamma = 0.1$ (d) $\Gamma = 0.1$

Figure 11: Sensitivity analysis on the customer frequencies.

The proposed algorithm was tested on instances from the literature and real-world instances. More precisely, we evaluated the performance of the algorithm on the set of instances of the periodic vehicle routing problem with driver consistency for which an exact branch-and-cut method has been proposed in the literature. Our results show that the heuristic computed optimal solutions for about 96% of the instances solved to optimality by the exact method, with an average running time of about one third of that of the exact method. Further, for the instances not solved to optimality by the exact method, the heuristic algorithm computed better solutions in a fraction of the computing time of the exact method. The algorithm was also extensively tested on real-world instances involving more than 1000 customers. Firstly, an analysis of the impact of the main algorithm's components showed the effectiveness of all the different algorithm's features. Secondly, there are several managerial insights, based on the results of the sensitivity analysis, that can be summarised as follows.

- The sensitivity analysis of the fixed cost associated with the areas showed that the (weighted) objective function in our problem is effective, and that a trade-off can exist between the area cost and the routing cost.

- Varying the time window widths can lead to significant cost savings. However, the company has very little power to change them.

- To evaluate scenarios with varying customer frequencies, we introduce a metric called the *Marginal Routing Cost* (MRC). For a given solution, the metric gives a measure of the impact of each customer's service cost on the total routing cost. We design a procedure that makes use of the MRC values to identify critical customers (in terms of distribution cost) and revises their visit frequencies in order to reduce the distribution cost while preserving the level of service. The experiments provide empirical evidence that the procedure has the potential to contribute to cost savings.

A preliminary version of the optimization algorithm described in this paper has been used by the company to evaluate different distribution scenarios. In addition to the global optimisation of a scenario, the optimisation tool has been enriched with additional functionalities, such as evaluating the addition of new customers to an existing distribution plan. Based on the

feedback from the company, the main benefits achieved can be summarised as follows.

- The optimisation provides a highly operational and cost-efficient distribution plan.
- A distribution plan is created in less time. Moreover, its definition no longer fully depends on the knowledge of the operators involved in the planning process, so the company is less dependent on these employees.
- The tool permits to evaluate what-if scenarios. In particular, the MRC are used to identify critical customers in terms of distribution cost and revise their visit frequencies. Moreover, what-if scenarios can be used to enhance the decision-making process.

We see three main topics for future work. First, the algorithm was used as a stand-alone tool without any graphical user interface. Therefore, the algorithm needs to be integrated into a decision support system that presents the information graphically to help the users make their decisions more easily. Second, balance requirements are important in real applications, hence the territories should also be balanced with respect to some measure (e.g., route duration or route profit), so we are planning to integrate route balancing in our solution framework. Finally, the assignment of customers to depots or transit points greatly impacts the distribution cost, and the results analysed by the company show that improvements can be made by revising the distribution areas of the different depots. This will involve partitioning customers into groups of depots, adding a higher decision level to our problem; the result will be a complex location-routing problem. Our aim is to tackle this challenging optimisation problem by combining different optimisation techniques.

Acknowledgment

The authors would like to thank the anonymous reviewers for their helpful suggestions and thorough review of the paper.

The authors thank Inmaculada Rodríguez-Martín for providing the instances and the corresponding details used in Rodríguez-Martín et al. [36].

This research was partially supported by the National Nature Science Foundation of China under Grant 71801025, and Scientific and Technological

Research Program of Chongqing Municipal Education Commission under Grant KJQN20181111.

Appendix A. Parameter calibration

Table A.6: Parameters used and final values found.

Reference	Parameter	Description	Search interval	Final value
Algorithm 1	n_{sp}	Number of subpopulations	3-8	6
	IT_{nimp}	Maximum number of non-improving iterations		200
	p_{ls}	Local search probability (%)	0-100	55
	Δ_0	Initial maximum population size	10-100	80
	IT_{sf}	Updating frequency of the subpopulation size (iterations)	10-100	50
Algorithm 2	ω	Working time adjust coefficient		0.8
Fitness function	α_1	Penalty term for time-window constraints	$0.01M - 0.15M$	$0.02M$
	α_2	Penalty term for working time constraints	$0.01M - 0.15M$	$0.1M$
	α_3	Penalty term for capacity constraints	$0.01M - 0.15M$	$0.1M$
	β_1	Fitness function weight		0.95
	β_2	Fitness function weight		0.05
Mutation	NP_{MUT}	Number of pairs of customers		$\max\{10, \lceil 0.1n_c \rceil\}$
	NC_{MUT}	Number of customers selected		$\max\{2, \lceil 0.05n_c \rceil\}$
Local improvement	T_0	Initial temperature	10-1000	100
	α_T	Cooling parameter	0.900-0.999	0.996
	T_f	Minimum temperature	0.001-0.1	0.02
Neighborhood structures	NC_{LP}	Number of selected customers		$\max\{2, \lceil 0.05n_c \rceil\}$
	NCC_{LP}	Number of nearest customers		$\max\{5, \lceil 0.01n_c \rceil\}$
	α_{LP}	Area elimination coefficient	0.01-0.05	0.02
Population management	ST_{PM}	Threshold for defining the number of seeds		$\lceil 0.1n_v \rceil$
	NS_{PM}	Number of initial solutions		100
	ISF_{PM}	Number of iterations (Shrinking function)	10-100	50
	η	Shrinking function parameter	3-6	5
	H	Shrinking function parameter	50-500	100

To produce suitable values for the parameters of the algorithm, we first defined the values of a set of parameters based on the results of preliminary

experiments. Then, we selected a second set of parameters to further extend the parameter calibration. For the second set of parameters, we used a preliminary meta-calibration based on the covariance matrix adaptation evolution strategy of Hansen [51]. During meta-calibration, the parameters are considered to be the decision variables, and the associated objective corresponds to the average solution quality of `HMPG` over ten runs on a set of training instances. This training set includes eight instances from the set of real-world instances.

Table A.6 lists the parameters used in our approach, the allowed range for each parameter, and the final values found by the meta-calibration process. When a range for a parameter is not reported, the value of the parameter was fixed during the preliminary experiments. The value of M reported in the table was set to 1000.

Appendix B. Detailed results on the set of instances of Rodríguez-Martín et al. [36]

Tables B.7-B.13 give the detailed results about the instances of the PVRP with driver consistency grouped for the number of customers in $\{10, 20, 30, 40, 50, 60, 70\}$.

The tables report for each instances, the relevant data of the instance (n_d, n_v, and Q), the cost of the best solution found by the branch-and-cut and the Algorithm `HMPG` ("z^*"), and for each method the solution cost ("z"), the percentage deviation ("$\%dev$") computed as $100.0 \times z/z^*$. In the tables, an entry marked with "-" under the branch-and-cut section means that the algorithm failed to compute a feasible solution for the corresponding instance. Moreover, for method `HMPG`, the last four sections of each table report: the maximum and the average values of the solution cost (section "z"), the minimum, maximum and average number of iterations performed to compute the best solution ("Its. to best"), the minimum, maximum and average time spent to compute the best solution ("Time to best") and the minimum, maximum and average total time ("Total time"). The last row of each table reports (i) the total sum of the costs of the best solutions found and (ii) the average of the solution costs and of time to best and the total time.

Table B.7: Results on instances from the literature with $n_c = 10$.

				Branch-and-cut			HMPG		z		Its. to best			Time to best			Total time		
n_d	n_v	Q	z^*	z	$\%dev$	t	z	$\%dev$	max	avg	min	max	avg	min	max	avg	min	max	avg
2	2	4	698.0	698.0	100.0	0.1	698.0	100.0	698.0	698.0	1	10	6.0	0.02	0.18	0.13	1.6	1.7	1.6
2	2	4	661.0	661.0	100.0	0.1	661.0	100.0	661.0	661.0	1	13	5.8	0.02	0.19	0.08	1.3	1.8	1.5
2	2	4	657.0	657.0	100.0	0.0	657.0	100.0	657.0	657.0	4	11	7.3	0.05	0.17	0.11	1.2	1.8	1.5
2	3	3	790.1	790.1	100.0	0.2	790.1	100.0	790.1	790.1	4	55	34.0	0.05	1.21	0.59	1.9	1.9	1.9
2	3	3	701.3	701.3	100.0	0.2	701.3	100.0	701.3	701.3	4	24	11.5	0.05	0.42	0.18	1.2	1.9	1.6
2	3	3	744.3	744.3	100.0	0.2	744.3	100.0	744.3	744.3	2	15	9.0	0.07	0.21	0.16	1.4	1.7	1.6
3	2	4	936.6	936.6	100.0	0.2	936.6	100.0	936.6	936.6	2	8	5.3	0.03	0.17	0.11	1.6	2.3	1.9
3	2	4	1032.3	1032.3	100.0	0.1	1032.3	100.0	1032.3	1032.3	1	1	1.0	0.01	0.08	0.04	1.6	1.9	1.8
3	2	4	759.6	759.6	100.0	0.0	759.6	100.0	759.6	759.6	1	31	17.8	0.02	0.34	0.21	1.3	1.8	1.6
3	3	3	1123.9	1123.9	100.0	0.6	1123.9	100.0	1123.9	1123.9	1	6	3.3	0.03	0.12	0.05	1.8	1.9	1.9
3	3	3	1189.5	1189.5	100.0	0.3	1189.5	100.0	1189.5	1189.5	1	2	1.5	0.01	0.02	0.02	1.5	1.9	1.7
3	3	3	832.3	832.3	100.0	0.1	832.3	100.0	832.3	832.3	1	4	2.5	0.03	0.07	0.05	1.4	1.6	1.6
4	2	4	1276.7	1276.7	100.0	0.3	1276.7	100.0	1276.7	1276.7	1	12	5.5	0.02	0.26	0.12	2.0	2.3	2.2
4	2	4	1400.1	1400.1	100.0	0.2	1400.1	100.0	1400.1	1400.1	1	3	1.3	0.01	0.04	0.03	1.8	2.1	1.9
4	2	4	1104.3	1104.3	100.0	0.4	1104.3	100.0	1104.3	1104.3	2	16	8.5	0.03	0.37	0.20	1.9	2.3	2.1
4	3	3	1457.3	1457.3	100.0	0.7	1457.3	100.0	1457.3	1457.3	1	20	10.8	0.06	0.74	0.37	2.3	2.9	2.4
4	3	3	1502.9	1502.9	100.0	0.5	1502.9	100.0	1502.9	1502.9	14	127	64.5	0.67	2.22	1.23	2.5	5.8	3.7
4	3	3	1203.3	1203.3	100.0	1.1	1203.3	100.0	1203.3	1203.3	11	44	24.0	0.06	1.44	0.91	3.7	7.7	6.1
5	2	4	1242.9	1242.9	100.0	1.3	1242.9	100.0	1242.9	1242.9	5	23	12.8	0.33	1.41	0.82	3.0	3.8	3.3
5	2	4	1759.9	1759.9	100.0	1.0	1759.9	100.0	1759.9	1759.9	7	18	13.0	0.33	0.69	0.50	3.0	3.3	3.1
5	2	4	1248.4	1248.4	100.0	0.4	1248.4	100.0	1248.4	1248.4	13	47	29.5	0.73	1.02	0.87	2.9	4.0	3.3
5	3	3	1432.1	1432.1	100.0	30.6	1432.1	100.0	1432.1	1432.1	7	28	16.5	0.48	0.67	0.56	2.3	2.9	2.6
5	3	3	1894.5	1894.5	100.0	3.8	1894.5	100.0	1894.5	1894.5	3	7	5.3	0.13	0.44	0.33	6.9	7.3	7.0
5	3	3	1415.0	1415.0	100.0	5.3	1415.0	100.0	1415.0	1415.0	3	8	5.3	0.03	0.16	0.09	2.0	2.4	2.2
							27063.0			27063.0						0.32			2.50

Branch-and-cut: exact method of Rodríguez-Martín et al. [36].

Table B.8: Results on instances from the literature with $n_c = 20$.

				Branch-and-cut			HMPG		z		Its. to best			Time to best			Total time		
n_d	n_v	Q	z^*	z	$\%dev$	t	z	$\%dev$	max	avg	min	max	avg	min	max	avg	min	max	avg
2	2	8	834.5	834.5	100.0	0.1	834.5	100.0	834.5	834.5	58	67	61.8	1.1	1.2	1.2	3.2	3.2	3.2
2	2	8	783.3	783.3	100.0	1.4	783.3	100.0	783.3	783.3	8	32	25.5	0.8	0.8	0.8	2.9	3.2	3.1
2	2	8	801.1	801.1	100.0	0.3	801.1	100.0	801.1	801.1	19	30	24.0	1.5	2.3	1.8	5.6	8.4	6.9
2	3	5	923.7	923.7	100.0	0.8	923.7	100.0	923.7	923.7	67	84	74.5	2.3	2.4	2.3	5.4	6.0	5.7
2	4	4	1138.1	1138.1	100.0	3.1	1138.1	100.0	1138.1	1138.1	11	34	20.5	1.4	2.2	1.9	8.0	9.5	8.6
2	4	4	1034.8	1034.8	100.0	30.7	1034.8	100.0	1034.8	1034.8	71	409	196.5	8.8	20.8	14.4	14.0	30.0	23.6
2	4	4	1003.3	1003.3	100.0	2.1	1003.3	100.0	1003.3	1003.3	33	54	40.5	4.0	6.2	4.8	14.8	17.8	16.0
3	2	8	1085.9	1085.9	100.0	0.5	1085.9	100.0	1085.9	1085.9	25	38	31.8	2.1	3.0	2.6	9.3	9.6	9.5
3	2	8	1160.1	1160.1	100.0	1.0	1160.1	100.0	1160.1	1160.1	22	40	30.5	2.0	2.5	2.2	8.8	9.5	9.2
3	2	8	1009.7	1009.7	100.0	0.7	1009.7	100.0	1009.7	1009.7	6	18	11.5	0.7	1.4	1.0	8.2	10.0	9.2
3	3	5	1231.6	1231.6	100.0	1.2	1231.6	100.0	1231.6	1231.6	15	40	28.3	1.9	4.1	2.8	12.6	15.5	13.6
3	3	5	1383.5	1383.5	100.0	7.7	1383.5	100.0	1383.5	1383.5	21	42	32.8	3.5	5.4	4.5	15.8	17.7	16.7
3	3	5	1193.7	1193.7	100.0	15.9	1193.7	100.0	1193.7	1193.7	45	85	65.5	7.2	12.2	9.7	20.3	26.3	23.8
3	4	4	1367.2	1367.2	100.0	7.8	1367.2	100.0	1368.0	1367.6	45	80	57.0	4.7	5.7	5.3	16.6	20.1	18.5
3	4	4	1504.2	1504.2	100.0	15.9	1504.2	100.0	1504.2	1504.2	55	110	80.8	9.2	16.4	12.2	19.0	32.6	24.9
3	4	4	1339.5	1339.5	100.0	124.1	1339.5	100.0	1343.9	1342.2	169	366	239.3	21.4	45.9	29.7	36.1	65.9	48.6
4	2	8	1335.9	1335.9	100.0	2.0	1335.9	100.0	1335.9	1335.9	159	288	205.5	6.0	10.0	7.4	12.2	16.1	13.6
4	2	8	1310.1	1310.1	100.0	2.0	1310.1	100.0	1310.1	1310.1	52	71	58.8	7.5	9.3	8.2	19.6	22.1	20.5
4	2	8	1208.8	1208.8	100.0	2.0	1208.8	100.0	1208.8	1208.8	45	86	64.3	6.7	8.9	7.9	12.7	13.8	13.2
4	3	5	1436.1	1436.1	100.0	2.3	1436.1	100.0	1436.1	1436.1	43	57	48.8	6.2	9.2	7.6	19.5	24.5	22.6
4	3	5	1538.9	1538.9	100.0	16.8	1538.9	100.0	1546.1	1541.3	68	98	87.5	15.2	18.2	17.0	31.7	33.4	32.9
4	3	5	1377.7	1377.7	100.0	10.7	1377.7	100.0	1377.7	1377.7	48	68	58.0	12.3	16.2	14.4	37.9	40.9	39.4
4	4	4	1676.0	1676.0	100.0	175.6	1676.0	100.0	1717.5	1686.4	28	137	82.8	4.1	19.5	11.6	22.4	33.0	26.9
4	4	4	1711.6	1711.6	100.0	1392.5	1711.6	100.0	1711.6	1711.6	57	662	466.3	9.7	100.4	76.8	28.8	129.4	104.2
4	4	4	1513.6	1513.6	100.0	97.2	1513.6	100.0	1513.6	1513.6	117	370	243.5	25.6	65.8	49.2	65.6	99.3	83.1
5	2	8	1838.3	1838.3	100.0	9.9	1838.3	100.0	1838.3	1838.3	17	27	23.5	4.4	7.0	5.3	26.8	28.7	27.7
5	2	8	1693.3	1693.3	100.0	2.0	1693.3	100.0	1693.3	1693.3	42	96	61.0	5.5	16.7	11.0	21.9	47.1	34.6
5	2	8	1892.9	1892.9	100.0	15.5	1892.9	100.0	1993.9	1929.3	55	354	245.8	4.3	25.0	17.5	18.2	41.3	33.0
5	3	5	2035.6	2035.6	100.0	18.9	2035.6	100.0	2049.2	2039.0	73	119	94.0	19.6	35.6	27.6	76.1	88.9	78.1
5	3	5	1942.3	1942.3	100.0	13.8	1942.3	100.0	2001.9	1963.8	551	622	576.3	34.1	125.2	67.0	48.1	157.1	87.4
5	3	5	2180.0	2180.0	100.0	142.8	2180.0	100.0	2180.0	2180.0	306	426	371.8	68.1	97.2	84.3	117.3	134.7	123.8
5	4	4	2218.7	2218.7	100.0	51.7	2218.7	100.0	2218.7	2218.7	182	255	206.3	34.5	45.6	38.2	70.5	74.1	71.7
5	4	4	2111.6	2111.6	100.0	78.9	2111.6	100.0	2111.6	2111.6	358	408	382.8	65.3	71.0	67.9	93.1	99.7	96.5
5	4	4	2317.7	2317.7	100.0	318.8	2317.7	100.0	2317.7	2317.7	253	814	452.5	55.0	143.2	88.8	94.5	174.5	125.8
							49133.1			49210.4						20.7			37.5

Branch-and-cut: exact method of Rodríguez-Martín et al. [36].

Table B.9: Results on instances from the literature with $n_c = 30$.

				Branch-and-cut			HMPG		z		Its. to best			Time to best			Total time		
n_d	n_v	Q	z^*	z	$\%dev$	t	z	$\%dev$	max	avg	min	max	avg	min	max	avg	min	max	avg
2	2	12	834.0	834.0	100.0	2.2	834.0	100.0	834.0	834.0	10	34	16.5	0.2	0.8	0.4	2.4	3.2	2.7
2	2	12	955.1	955.1	100.0	1.5	955.1	100.0	955.1	955.1	197	240	214.9	5.2	6.6	5.8	9.9	11.9	10.7
2	2	12	815.1	815.1	100.0	1.6	815.1	100.0	815.1	815.1	94	327	224.3	7.3	23.2	16.0	20.3	36.9	29.7
2	3	8	964.6	964.6	100.0	23.4	964.6	100.0	967.6	965.3	161	339	264.8	9.8	14.7	11.8	15.5	22.7	18.5
2	3	8	1096.0	1096.0	100.0	13.2	1096.0	100.0	1096.0	1096.0	16	221	155.0	0.5	7.5	5.4	7.7	14.4	12.4
2	3	8	924.8	924.8	100.0	7.3	924.8	100.0	938.1	933.0	344	494	448.8	10.9	37.5	21.3	16.8	52.1	30.1
2	4	6	1075.2	1075.2	100.0	51.5	1075.2	100.0	1090.4	1079.0	44	353	198.3	4.9	34.5	20.1	24.5	52.3	38.3
2	4	6	1244.0	1244.0	100.0	90.8	1244.0	100.0	1253.1	1246.3	220	632	361.5	47.5	105.9	66.8	78.6	132.4	97.7
2	4	6	1033.1	1033.1	100.0	11.1	1033.1	100.0	1033.1	1033.1	75	207	159.5	8.3	29.5	20.0	35.3	58.9	46.0
3	2	12	1193.6	1193.6	100.0	1.7	1193.6	100.0	1193.6	1193.6	78	601	397.8	9.4	60.9	42.5	30.9	77.1	61.0
3	2	12	1328.1	1328.1	100.0	6.4	1328.1	100.0	1328.1	1328.1	31	425	277.3	7.3	35.0	24.8	46.5	55.2	50.5
3	2	12	1160.4	1160.4	100.0	1.9	1160.4	100.0	1161.7	1160.9	240	706	433.8	26.4	72.1	45.2	45.6	87.3	62.7
3	3	8	1386.1	1386.1	100.0	32.6	1386.1	100.0	1386.1	1386.1	99	147	127.5	13.2	22.9	18.1	39.1	52.4	45.3
3	3	8	1477.9	1477.9	100.0	20.4	1477.9	100.0	1482.2	1479.0	58	234	148.5	7.2	26.5	16.9	32.0	47.6	40.0
3	3	8	1305.1	1305.1	100.0	13.8	1305.1	100.0	1306.9	1305.6	150	397	282.5	18.9	47.7	33.5	44.1	78.5	60.8
3	4	6	1564.9	1564.9	100.0	85.9	1564.9	100.0	1573.0	1568.9	129	175	145.8	22.8	28.3	24.9	52.5	58.2	56.1
3	4	6	1596.0	1596.0	100.0	32.1	1596.0	100.0	1607.3	1601.6	90	176	134.3	14.4	23.6	19.2	42.1	48.3	45.2
3	4	6	1532.6	1532.6	100.0	1130.0	1532.6	100.0	1538.2	1535.0	177	879	444.5	22.8	112.5	58.9	48.1	138.1	83.9
4	2	12	1626.4	1626.4	100.0	22.7	1626.4	100.0	1626.4	1626.4	93	345	242.0	3.4	14.6	10.1	11.1	22.0	17.6
4	2	12	1566.6	1566.6	100.0	12.0	1566.6	100.0	1566.6	1566.6	153	282	225.7	16.9	29.7	25.4	36.4	53.3	46.3
4	2	12	1546.2	1546.2	100.0	9.7	1546.2	100.0	1554.5	1550.4	134	926	368.0	22.3	32.6	25.0	49.7	57.4	53.6
4	3	8	1844.3	1844.3	100.0	45.8	1844.3	100.0	1854.1	1847.6	181	200	189.0	44.3	51.4	47.9	87.2	96.2	92.4
4	3	8	1728.3	1728.3	100.0	88.3	1728.3	100.0	1757.6	1738.0	234	1003	434.5	42.3	78.4	52.3	75.2	89.8	79.3
4	3	8	1783.5	1783.5	100.0	59.1	1783.5	100.0	1789.6	1785.6	111	692	488.0	25.3	109.2	79.3	61.2	140.0	108.3
4	4	6	2031.0	2031.0	100.0	675.5	2031.0	100.0	2031.0	2031.0	192	438	313.8	49.8	78.9	65.0	95.3	124.8	115.1
4	4	6	1867.0	1867.0	100.0	200.8	1867.0	100.0	1887.6	1878.7	951	1221	1081.0	149.0	178.1	164.2	174.2	202.0	184.9
4	4	6	1989.1	1989.1	100.0	241.7	1989.1	100.0	1989.1	1989.1	384	555	469.0	85.9	114.7	100.5	124.6	151.0	138.0
5	2	12	1690.4	1690.4	100.0	10.0	1690.4	100.0	1690.4	1690.4	84	176	126.0	10.3	22.7	15.9	33.8	45.4	38.7
5	2	12	2144.7	2144.7	100.0	11.0	2144.7	100.0	2149.5	2146.5	165	384	245.3	26.5	59.0	39.7	56.2	85.4	69.0
5	2	12	1656.8	1656.8	100.0	7.3	1656.8	100.0	1660.8	1658.2	97	850	368.8	15.9	83.1	40.1	45.0	98.1	64.1
5	3	8	1852.0	1852.0	100.0	233.7	1852.0	100.0	1859.3	1856.9	340	1443	655.8	74.1	170.5	105.0	92.1	185.9	128.4
5	3	8	2410.7	2410.7	100.0	148.5	2410.7	100.0	2410.7	2410.7	334	717	470.5	76.3	160.3	111.8	118.0	196.3	153.0
5	3	8	1829.8	1829.8	100.0	655.6	1829.8	100.0	1844.5	1834.7	1408	2234	1935.6	257.0	372.0	330.5	278.3	390.6	350.0
5	4	6	1978.1	1978.1	100.0	293.3	1978.1	100.0	1998.8	1988.3	577	1326	1014.3	126.1	272.0	211.0	166.7	581.6	316.9
5	4	6	2676.9	2676.9	100.0	3813.7	2676.9	100.0	2676.9	2676.9	466	1029	668.8	128.9	217.7	164.8	173.4	253.8	205.9
5	4	6	2013.3	2013.3	100.0	1372.6	2013.3	100.0	2024.3	2017.9	417	1872	1132.8	93.7	301.9	201.2	130.5	346.0	240.1
							55721.5			55809.5						62.3			88.7

Branch-and-cut: exact method of Rodríguez-Martín et al. [36].

Table B.10: Results on instances from the literature with $n_c = 40$.

				Branch-and-cut			HMPG		z		Its. to best			Time to best			Total time		
n_d	n_v	Q	z^*	z	$\%dev$	t	z	$\%dev$	max	avg	min	max	avg	min	max	avg	min	max	avg
2	2	15	897.4	897.4	100.0	9.7	897.4	100.0	901.7	898.5	125	145	138.0	16.2	18.9	17.8	28.7	30.9	30.2
2	2	15	1049.8	1049.8	100.0	15.0	1049.8	100.0	1061.7	1052.8	139	186	155.0	10.0	17.5	13.9	23.9	35.9	31.5
2	3	10	993.0	993.0	100.0	38.6	993.0	100.0	1006.3	998.6	382	798	588.0	64.5	154.7	107.0	94.5	170.2	130.4
2	3	10	1189.7	1189.7	100.0	125.2	1189.7	100.0	1195.1	1191.5	714	1327	955.0	82.3	139.4	104.5	104.6	160.8	126.0
2	4	8	1103.5	1103.5	100.0	171.1	1103.5	100.0	1104.0	1103.6	73	592	217.5	18.0	57.2	32.0	60.9	98.5	77.2
2	4	8	1208.5	1208.5	100.0	108.3	1208.5	100.0	1208.5	1208.5	154	1028	607.8	39.0	177.4	115.7	82.7	202.7	149.6
2	4	8	1249.0	1249.0	100.0	294.5	1249.0	100.0	1298.5	1273.4	189	741	381.5	48.6	65.6	56.2	85.5	97.1	90.7
3	2	15	1285.0	1285.0	100.0	19.5	1285.0	100.0	1308.0	1297.9	320	14593	4028.5	36.4	1614.7	465.6	56.0	1637.7	489.2
3	2	15	1500.0	1500.0	100.0	34.7	1500.0	100.0	1516.1	1511.7	461	3008	1515.8	43.6	290.4	145.9	62.7	310.7	166.0
3	2	15	1520.6	1520.6	100.0	27.6	1520.6	100.0	1545.6	1532.4	444	3898	1638.0	38.9	348.8	146.1	61.6	367.0	168.1
3	3	10	1412.6	1412.6	100.0	38.0	1412.6	100.0	1440.2	1428.4	539	1404	957.8	54.2	148.3	100.1	76.7	169.7	121.8
3	3	10	1618.0	1618.0	100.0	111.5	1618.0	100.0	1647.7	1628.7	35	1890	734.8	4.0	247.9	90.2	29.5	274.1	114.1
3	3	10	1686.3	1686.3	100.0	84.9	1686.3	100.0	1688.9	1687.8	284	1718	909.5	51.8	229.6	134.5	83.9	246.5	158.8
3	4	8	1575.0	1575.0	100.0	4964.2	1575.0	100.0	1634.3	1597.6	2538	6289	3893.8	345.9	1011.4	586.3	371.1	1041.0	614.5
3	4	8	1743.0	1743.0	100.0	330.8	1743.0	100.0	1743.0	1743.0	97	209	179.0	44.8	89.6	58.3	92.7	168.4	114.3
3	4	8	1856.6	1856.6	100.0	3982.3	1856.6	100.0	1862.4	1860.6	61	791	252.5	12.2	136.9	45.5	50.9	161.4	79.0
4	2	15	1869.4	1869.4	100.0	85.4	1869.4	100.0	1874.7	1871.1	213	1001	638.8	36.2	145.9	95.1	65.4	168.6	120.2
4	2	15	1962.4	1962.4	100.0	210.1	1986.4	101.2	1986.4	1979.3	865	6197	3435.3	87.2	650.2	386.3	108.4	660.1	404.0
4	3	10	1750.8	1750.8	100.0	182.4	1750.8	100.0	1780.6	1767.0	537	2066	1310.9	68.0	283.7	175.1	93.4	310.4	201.7
4	3	10	2062.2	2062.2	100.0	3851.2	2062.2	100.0	2094.2	2078.4	1190	4019	2314.8	186.9	568.6	332.8	199.5	598.4	344.5
4	3	10	2158.2	2158.2	100.0	1691.6	2158.2	100.0	2169.2	2164.3	240	1742	1218.8	89.3	284.7	218.5	200.2	315.4	273.7
4	4	8	1797.9	1797.9	100.0	271.1	1797.9	100.0	1814.1	1808.1	1003	2493	1659.8	256.3	284.6	270.6	283.9	313.9	302.6
4	4	8	2100.4	2100.4	100.0	7200.0	2131.2	101.5	2187.0	2174.3	1629	5620	4204.0	255.3	1567.4	891.1	286.0	1613.3	927.2
4	4	8	2314.8	2314.8	100.0	7200.0	2335.4	100.9	2368.4	2364.8	5842	5977	5943.3	859.1	1623.9	1432.7	889.1	1682.9	1484.4
5	2	15	2100.5	2100.5	100.0	187.7	2100.5	100.0	2128.6	2116.8	1249	1931	1568.0	163.7	255.3	206.1	194.2	279.9	236.0
5	2	15	2240.7	2240.7	100.0	112.9	2240.7	100.0	2253.7	2246.1	517	1216	811.8	98.2	174.4	130.2	129.7	194.9	157.2
5	2	15	2165.4	2165.4	100.0	1375.5	2165.4	100.0	2189.1	2181.2	3081	5646	4872.0	492.7	808.3	707.3	525.3	832.6	735.6
5	3	10	2339.7	2339.7	100.0	2606.4	2339.7	100.0	2350.5	2347.6	1366	2738	2269.3	252.8	479.5	411.0	293.8	514.3	447.6
5	3	10	2468.9	2468.9	100.0	2344.1	2468.9	100.0	2484.8	2476.0	1898	3104	2507.3	279.9	571.2	433.9	305.0	610.2	467.9
5	3	10	2426.1	2427.1	100.0	7200.0	2426.1	100.0	2429.0	2426.5	2747	7327	4876.0	481.9	1211.3	815.1	516.9	1244.2	848.5
5	4	8	2508.3	2508.3	100.0	7200.0	2510.1	100.1	2655.4	2551.8	8865	10340	9916.6	1796.9	2215.5	2094.8	1836.4	2266.3	2136.2
5	4	8	2530.5	2530.5	100.0	7200.0	2566.1	101.4	2666.1	2594.8	1279	4327	3452.0	257.0	846.5	677.3	294.4	886.6	716.6
5	4	8	2468.8	2468.8	100.0	7200.0	2489.1	100.8	2537.0	2506.4	1593	1988	1735.3	322.4	410.3	354.1	357.3	450.5	391.0
							59285.9			59669.2						359.1			389.6

Branch-and-cut: exact method of Rodríguez-Martín et al. [36].

Table B.11: Results on instances from the literature with $n_c = 50$.

				Branch-and-cut			HMPG		z		Its. to best			Time to best			Total time		
n_d	n_v	Q	z^*	z	$\%dev$	t	z	$\%dev$	max	avg	min	max	avg	min	max	avg	min	max	avg
2	2	19	1109.1	1109.1	100.0	36.4	1109.1	100.0	1133.1	1117.1	435	644	535.0	56.5	72.8	62.0	74.2	99.5	85.2
2	2	19	1053.5	1053.5	100.0	31.5	1053.5	100.0	1057.4	1054.8	960	1656	1203.0	96.5	144.7	112.6	117.2	162.7	133.6
2	2	19	1050.5	1050.5	100.0	15.5	1050.5	100.0	1070.8	1060.7	677	934	845.0	62.1	80.8	72.9	78.1	97.9	90.1
2	3	13	1213.9	1213.9	100.0	76.5	1213.9	100.0	1227.0	1221.4	169	1793	1331.0	47.0	244.6	181.9	93.1	249.7	203.9
2	3	13	1148.0	1148.0	100.0	137.8	1148.0	100.0	1175.2	1157.1	875	966	922.8	95.5	104.0	100.0	114.5	123.2	119.2
2	3	13	1129.2	1129.2	100.0	47.0	1129.2	100.0	1136.0	1131.7	344	467	397.5	88.9	115.5	100.6	131.9	155.2	142.0
2	4	10	1335.6	1335.6	100.0	580.8	1335.6	100.0	1335.6	1335.6	306	654	493.8	36.4	77.2	58.1	61.0	101.3	82.8
2	4	10	1231.8	1231.8	100.0	1786.9	1231.8	100.0	1231.8	1231.8	3122	4543	3632.3	365.5	548.5	431.1	392.3	571.7	456.3
2	4	10	1229.2	1229.2	100.0	1451.8	1229.2	100.0	1229.2	1229.2	1243	2590	1855.8	152.4	308.3	222.7	176.5	332.1	246.8
3	2	19	1690.0	1690.0	100.0	85.6	1690.0	100.0	1729.6	1705.2	407	686	488.8	50.1	85.4	66.9	76.1	109.0	93.6
3	2	19	1581.6	1581.6	100.0	55.1	1581.6	100.0	1649.4	1615.1	204	327	270.3	32.1	39.7	37.7	55.6	63.4	59.2
3	2	19	1638.7	1638.7	100.0	404.3	1638.7	100.0	1649.1	1643.6	590	1494	1190.5	81.4	228.8	190.1	111.5	256.2	209.3
3	3	13	1815.1	1815.1	100.0	237.1	1815.1	100.0	1847.0	1833.6	1418	1668	1523.8	212.5	251.3	228.9	244.2	281.1	259.8
3	3	13	1714.3	1714.3	100.0	289.4	1714.3	100.0	1715.5	1715.1	1977	2336	2153.7	315.0	369.9	342.6	347.2	404.1	373.8
3	3	13	1691.4	1691.4	100.0	254.6	1691.4	100.0	1695.0	1693.4	1122	4077	2319.3	186.6	681.1	390.9	221.6	714.3	426.6
3	4	10	2012.7	2018.2	100.3	7200.0	2012.7	100.0	2021.9	2017.9	4435	7379	6520.5	768.7	1279.9	1130.9	800.5	1312.1	1164.2
3	4	10	1845.9	1847.0	100.1	7200.0	1845.9	100.0	1847.2	1845.9	4617	9684	6173.5	763.0	1630.9	1031.9	799.2	1667.8	1067.4
3	4	10	1859.6	1866.2	100.4	7200.0	1859.6	100.0	1868.0	1865.4	1615	2832	2111.5	283.1	487.7	368.8	322.5	520.4	404.0
4	2	19	1968.5	1968.5	100.0	782.2	1968.5	100.0	2000.9	1987.1	3922	5120	4404.5	554.1	687.6	604.9	581.8	714.7	633.2
4	2	19	1821.9	1821.9	100.0	1660.5	1821.9	100.0	1888.2	1851.4	3021	3865	3396.0	433.2	534.0	478.0	460.0	560.4	504.6
4	2	19	1815.8	1815.8	100.0	71.8	1815.8	100.0	1868.7	1846.5	1780	12151	5454.3	238.4	1526.5	695.5	265.4	1550.1	720.6
4	3	13	2120.5	2129.3	100.4	7200.0	2120.5	100.0	2129.3	2125.1	3929	9780	6152.5	582.9	1175.6	878.3	606.2	1227.2	913.9
4	3	13	2015.7	2024.0	100.4	7200.0	2015.7	100.0	2034.1	2021.8	2173	10323	6406.3	376.6	1810.3	1120.6	411.8	1846.5	1155.2
4	3	13	1928.4	1928.4	100.0	7200.0	1934.6	100.3	1938.2	1935.8	1418	4534	2456.7	230.9	769.7	410.5	266.3	801.0	444.6
4	4	10	2230.0	2234.4	100.2	7200.0	2230.0	100.0	2270.0	2250.5	5302	11688	8967.5	940.2	2156.7	1654.0	976.2	2194.7	1690.5
4	4	10	2083.6	2083.6	100.0	7200.0	2116.2	101.6	2134.9	2129.5	5894	14850	11615.5	1187.9	2852.3	2251.3	1228.9	2897.5	2295.0
4	4	10	1957.2	1957.2	100.0	4908.7	1957.2	100.0	1959.7	1958.9	4932	5274	5074.0	911.5	930.0	923.1	948.3	964.4	958.4
5	2	19	2486.6	2486.6	100.0	1844.7	2535.2	102.0	2535.2	2534.7	3820	4349	4039.7	611.2	708.1	658.4	642.4	740.9	690.7
5	2	19	2450.3	2450.3	100.0	1930.8	2450.3	100.0	2555.9	2488.4	3452	7870	6274.5	706.0	1435.2	1171.9	747.4	1469.7	1208.9
5	2	19	2466.6	2466.6	100.0	267.4	2466.6	100.0	2551.7	2497.3	2256	4350	3727.8	378.6	705.0	612.0	412.5	741.1	647.4
5	3	13	2719.7	2732.8	100.5	7200.0	2719.7	100.0	2732.8	2719.7	3666	7899	5119.3	766.4	1731.6	1103.5	809.4	1777.2	1148.1
5	3	13	2723.4	2727.8	100.2	7200.0	2723.4	100.0	2778.8	2737.3	2981	6954	4056.0	678.9	1669.7	946.7	722.6	1723.1	994.4
5	3	13	2706.2	2706.2	100.0	7200.0	2706.8	100.0	2706.9	2706.7	1045	5726	2758.8	203.5	1206.0	574.7	248.2	1251.5	613.9
5	4	10	2831.1	2831.1	100.0	7200.0	2831.6	100.0	2859.8	2846.1	957	5325	3303.0	218.6	1224.0	753.7	260.9	1267.9	797.1
5	4	10	3053.7	3071.0	100.6	7200.0	3053.7	100.0	3071.0	3026.4	2081	9010	3974.3	521.4	2185.7	966.3	563.3	2237.0	1015.1
5	4	10	2750.0	2750.0	100.0	7200.0	2791.2	101.5	2845.9	2815.5	3999	4577	4312.0	1000.1	1057.2	1019.2	1044.0	1102.8	1063.6
							68609.0			68953.3						609.8			642.0

Branch-and-cut: exact method of Rodríguez-Martín et al. [36].

Table B.12: Results on instances from the literature with $n_c = 60$.

				Branch-and-cut			HMPG		z		Its. to best			Time to best			Total time		
n_d	n_v	Q	z^*	z	$\%dev$	t	z	$\%dev$	max	avg	min	max	avg	min	max	avg	min	max	avg
2	2	23	1183.5	1183.5	100.0	152.8	1183.5	100.0	1227.1	1195.6	1577	1633	1612.5	245.8	282.9	259.2	275.3	314.6	289.5
2	2	23	1151.3	1151.3	100.0	70.3	1151.3	100.0	1158.5	1153.1	1122	1263	1181.8	101.3	109.2	105.9	119.4	130.0	124.0
2	2	23	1141.7	1141.7	100.0	75.6	1141.7	100.0	1212.9	1183.2	544	1010	719.8	53.4	97.4	69.6	74.3	118.1	90.0
2	3	15	1224.4	1224.4	100.0	419.1	1224.4	100.0	1235.5	1227.1	2543	3044	2918.8	300.2	361.1	345.9	323.9	386.2	370.6
2	3	15	1240.2	1240.2	100.0	2970.3	1240.2	100.0	1324.2	1263.5	3061	3315	3195.1	390.4	435.8	414.3	415.8	460.9	439.6
2	4	12	1348.4	1348.4	100.0	7200.0	1356.5	100.6	1358.8	1356.0	915	4796	3343.0	125.6	669.2	458.0	152.8	694.8	484.1
2	4	12	1332.2	1332.2	100.0	3365.2	1332.2	100.0	1364.5	1343.8	968	2332	1839.3	117.3	655.4	461.1	142.5	716.5	509.2
2	4	12	1332.2	1332.2	100.0	1564.2	1332.2	100.0	1335.3	1333.0	1347	6555	4149.0	167.0	841.8	528.2	192.7	868.1	550.8
3	2	23	1756.6	1756.6	100.0	440.4	1756.6	100.0	1865.5	1854.5	903	2926	1701.3	124.5	396.5	232.2	151.8	424.2	259.5
3	2	23	1710.4	1710.4	100.0	665.0	1710.4	100.0	1775.0	1728.4	2580	4080	2996.5	325.5	376.8	339.7	351.1	394.4	362.9
3	2	23	1606.9	1606.9	100.0	440.9	1606.9	100.0	1632.1	1625.0	1873	4490	3447.5	233.0	1031.9	573.1	257.1	1082.3	596.5
3	3	15	1846.1	1846.1	100.0	1789.8	1846.1	100.0	1907.8	1873.6	3107	3742	3459.7	544.8	647.5	601.9	580.4	682.0	636.8
3	3	15	1856.5	1856.5	100.0	5737.7	1856.5	100.0	1897.9	1895.2	555	4386	2745.0	116.2	733.9	471.0	160.4	766.6	507.7
3	3	15	1748.1	1748.1	100.0	5201.3	1781.0	101.9	1798.5	1795.9	1783	3027	2468.3	290.1	503.7	409.5	323.7	536.7	442.4
3	4	12	2036.0	2081.0	102.2	7200.0	2036.0	100.0	2081.0	2040.2	1329	2973	2187.5	241.8	542.9	399.3	274.6	581.1	434.3
3	4	12	1984.9	1987.7	100.1	7200.0	1984.9	100.0	1987.7	1985.7	4502	6100	4943.0	766.3	1020.4	832.9	803.9	1052.3	867.9
3	4	12	1903.2	1903.2	100.0	7200.0	1927.9	101.3	1933.3	1929.9	1767	3032	2281.0	489.9	534.1	511.6	549.0	629.0	583.2
4	2	23	2217.5	2217.5	100.0	1003.6	2217.5	100.0	2235.4	2223.3	3149	6439	4675.7	522.0	1107.7	796.2	557.0	1142.2	831.5
4	2	23	1991.9	1991.9	100.0	773.9	1991.9	100.0	2001.0	1999.9	6921	10452	9340.8	749.8	1602.8	1355.5	769.6	1630.7	1382.6
4	2	23	2007.9	2007.9	100.0	974.8	2007.9	100.0	2045.6	2024.6	6695	7524	7155.3	961.6	2061.7	1572.8	991.7	2117.0	1616.8
4	3	15	2382.0	2382.0	100.0	6978.9	2382.0	100.0	2410.8	2394.8	9475	9882	9701.0	2411.2	2688.0	2565.0	2462.6	2742.4	2618.1
‘	3	15	2176.4	2177.2	100.0	7200.0	2176.4	100.0	2177.4	2176.6	4021	9176	6217.3	770.2	1774.4	1198.9	820.3	1813.2	1241.4
4	3	15	2195.6	2195.6	100.0	7200.0	2198.4	100.1	2202.0	2199.6	4007	5751	4700.0	697.6	1027.8	829.3	736.0	1059.8	866.0
4	4	12	2549.5	2728.0	107.0	7200.0	2549.5	100.0	2728.0	2547.2	1068	3096	2419.7	239.8	705.1	549.9	283.1	751.0	594.9
4	4	12	2305.0	2316.4	100.5	7200.0	2305.0	100.0	2316.4	2308.8	2258	8515	5734.1	432.5	1627.0	1096.1	471.5	1667.4	1135.9
4	4	12	2238.5	2238.5	100.0	7200.0	2249.1	100.5	2263.2	2253.8	4754	4807	4771.7	891.3	902.1	894.9	928.8	939.3	932.3
5	2	23	2482.6	2482.6	100.0	364.0	2486.4	100.2	2536.4	2518.0	3182	3404	3330.0	624.4	667.6	653.2	663.1	706.0	691.7
5	2	23	2430.1	2430.1	100.0	505.1	2430.1	100.0	2432.1	2430.9	4462	6627	5767.0	732.1	1098.0	953.6	764.9	1129.9	986.7
5	2	23	2617.3	2617.3	100.0	7200.0	2670.0	102.0	2698.4	2682.7	1459	6063	4016.7	264.8	1648.5	1033.5	304.1	1707.8	1083.9
5	3	15	2721.4	2722.2	100.0	7200.0	2721.4	100.0	2726.2	2723.6	704	6393	2548.0	236.7	1526.7	655.1	295.6	1574.1	703.1
5	3	15	2648.9	2648.9	100.0	7200.0	2707.2	102.2	2728.1	2712.4	8193	9750	9360.8	1715.2	2655.3	2420.3	1756.1	2707.8	2469.9
5	3	15	2847.5	2847.5	100.0	7200.0	2888.9	101.5	2909.9	2896.7	5274	7307	6026.7	803.5	1800.2	1172.7	845.2	1849.2	1217.0
5	4	12	2891.7	2891.7	100.0	7200.0	2947.7	101.9	2949.7	2948.1	813	5514	3538.0	174.6	1227.2	796.0	222.3	1273.9	842.7
5	4	12	2757.7	2757.7	100.0	7200.0	2816.3	102.1	2817.1	2815.7	2930	8239	4723.3	701.8	2256.8	1222.6	728.4	2310.2	1259.4
5	4	12	3099.8	0.0	0.0	7200.0	3099.8	100.0	3101.8	3100.8	5046	5082	5058.3	1223.0	1413.6	1286.6	1276.1	5282.0	2611.5
							71313.7			71741.1						801.9			875.3

Branch-and-cut: exact method of Rodríguez-Martín et al. [36].

Table B.13: Results on instances from the literature with $n_c = 70$.

				Branch-and-cut			HMPG		z		Its. to best			Time to best			Total time		
n_d	n_v	Q	z^*	z	$\%dev$	t	z	$\%dev$	max	avg	min	max	avg	min	max	avg	min	max	avg
2	2	27	1266.3	1266.3	100.0	143.6	1266.3	100.0	1269.4	1267.1	981	1052	1026.3	114.7	122.9	119.9	137.9	146.0	143.1
2	2	27	1219.5	1219.5	100.0	329.5	1219.5	100.0	1243.8	1233.3	598	4169	1827.8	69.2	1155.4	378.8	89.5	1213.5	410.0
2	2	27	1199.9	1199.9	100.0	89.2	1199.9	100.0	1265.9	1242.0	980	6741	2945.3	98.5	973.4	515.4	120.7	1120.9	568.7
2	3	18	1371.1	1371.1	100.0	1295.8	1396.5	101.9	1396.5	1379.0	2218	2265	2135.3	319.8	326.9	324.9	345.4	351.9	350.1
2	3	18	1302.4	1302.4	100.0	4894.7	1302.4	100.0	1318.7	1311.6	719	5300	2852.3	92.6	692.2	373.8	117.8	717.6	399.7
2	3	18	1278.2	1278.2	100.0	343.3	1278.2	100.0	1284.4	1281.7	1909	2023	1968.5	245.4	251.8	249.6	269.3	277.6	274.5
2	4	14	1457.6	1457.6	100.0	7200.0	1457.6	100.0	1458.9	1458.4	1885	10428	6374.3	277.6	1403.4	903.4	309.0	1427.9	933.0
2	4	14	1405.4	1406.9	100.1	7200.0	1405.4	100.0	1406.9	1406.3	6184	12063	8096.5	822.1	1557.9	1063.8	848.6	1582.5	1089.6
2	4	14	1351.4	1351.4	100.0	1839.8	1351.4	100.0	1383.9	1375.0	2005	4774	4023.8	259.9	660.4	537.4	289.0	690.1	564.6
3	2	27	1666.7	1666.7	100.0	596.6	1666.7	100.0	1699.6	1681.8	3282	7120	4614.3	453.5	1120.1	645.9	474.7	1169.3	679.0
3	2	27	1650.9	1650.9	100.0	480.6	1650.9	100.0	1661.8	1655.6	1734	3766	2734.3	226.9	498.1	360.4	255.3	523.4	388.1
3	2	27	1602.4	1602.4	100.0	315.6	1602.4	100.0	1617.0	1611.9	2679	4177	3227.0	335.2	538.8	419.6	360.1	554.7	441.7
3	3	18	1799.0	1799.0	100.0	5929.9	1799.0	100.0	1808.5	1805.1	2191	3418	2746.3	359.5	582.1	464.8	394.4	613.8	498.7
3	3	18	1761.3	1761.3	100.0	3150.7	1761.3	100.0	1777.8	1767.9	2640	12884	9552.8	413.3	1937.6	1561.8	446.4	2030.1	1507.5
3	3	18	1720.3	1720.3	100.0	709.0	1720.3	100.0	1821.5	1753.3	914	3198	2152.0	132.9	489.2	327.0	165.2	518.6	358.2
3	4	14	1960.2	1964.2	100.2	7200.0	1960.2	100.0	1979.7	1966.7	3568	14886	7217.0	610.9	4078.7	1627.6	647.1	4134.4	1669.5
3	4	14	1945.4	1945.4	100.0	7200.0	1954.9	100.5	1976.2	1961.5	3890	6059	5122.8	646.8	1023.8	861.3	677.9	1060.7	895.5
3	4	14	1882.6	1882.6	100.0	7200.0	1897.2	100.8	1943.1	1914.8	1424	5455	3670.8	218.8	573.7	448.8	259.7	595.4	478.0
4	2	27	2186.3	2186.3	100.0	1678.4	2186.3	100.0	2196.4	2188.8	6417	9529	7351.0	1005.8	1442.7	1131.5	1037.6	1474.1	1162.9
4	2	27	2201.7	2201.7	100.0	7200.0	2222.0	100.9	2276.5	2261.1	3057	4538	3816.8	494.7	708.7	603.2	525.0	737.8	634.0
4	2	27	2189.5	2189.5	100.0	284.2	2221.9	101.5	2234.5	2226.1	4224	5364	5036.5	702.5	1490.4	1264.2	735.5	1547.2	1314.2
4	3	18	2372.4	2465.6	103.9	7200.0	2372.4	100.0	2465.6	2373.0	2899	4299	3407.0	559.4	850.1	666.8	619.5	888.9	711.1
4	3	18	2406.0	2525.0	104.9	7200.0	2406.0	100.0	2525.0	2401.5	3262	4202	3597.0	645.4	848.3	719.7	687.8	890.4	761.2
4	3	18	2381.9	2381.9	100.0	7200.0	2385.0	100.1	2395.8	2390.1	3428	6301	4365.8	707.4	1282.7	898.5	748.0	1321.9	940.8
4	4	14	2609.9	2633.7	100.9	7200.0	2609.9	100.0	2633.7	2609.1	1820	2535	2148.8	364.5	515.0	431.8	405.8	553.2	471.8
4	4	14	2595.6	0.0	0.0	7200.0	2595.6	100.0	2595.6	2588.7	4374	5624	5223.0	931.6	1194.7	1097.6	974.4	1240.8	1140.5
4	4	14	2576.5	2589.4	100.5	7200.0	2576.5	100.0	2589.4	2545.0	1117	8539	5695.5	214.5	1645.6	1007.5	254.3	1689.6	1046.5
5	2	27	2885.6	2885.6	100.0	7108.7	2895.2	100.3	2963.9	2912.4	2157	7126	3812.0	479.1	984.7	702.9	661.7	2257.0	1178.0
5	2	27	2621.0	2621.0	100.0	7200.0	2629.0	100.3	2634.8	2632.9	3334	7867	6031.3	742.6	1704.0	1318.1	786.2	1748.5	1361.8
5	2	27	2580.4	2580.4	100.0	5430.7	2580.4	100.0	2636.6	2616.7	396	13918	5140.8	62.7	3443.9	1190.1	103.5	3493.9	1233.5
5	3	18	3418.0	0.0	0.0	7200.0	3418.0	100.0	3418.0	3405.2	2178	11006	6501.3	593.6	4027.1	2274.1	651.1	4103.4	2339.3
5	3	18	3190.8	0.0	0.0	7200.0	3190.8	100.0	3190.8	2935.7	4788	7352	6051.8	1242.3	1915.5	1568.5	1290.6	1968.6	1620.6
5	3	18	2751.9	2773.1	100.8	7200.0	2751.9	100.0	2773.1	2750.2	1305	11245	8033.8	290.0	2829.9	2011.6	339.5	2880.7	2061.3
5	4	14	3416.9	0.0	0.0	7200.0	3416.9	100.0	3423.8	3420.0	2117	4735	3576.5	629.1	1446.8	1077.5	685.3	1509.4	1136.9
5	4	14	3192.0	0.0	0.0	7200.0	3192.0	100.0	3196.5	3191.2	4510	7018	5694.5	1285.6	1921.2	1546.5	1342.4	1976.4	1600.6
5	4	14	2888.0	2888.0	100.0	7200.0	2919.4	101.1	2976.1	2940.7	4040	12209	7754.8	991.5	2994.0	1896.2	1026.9	3045.3	1942.4
							76459.1			76461.2						905.3			953.0

Branch-and-cut: exact method of Rodríguez-Martín et al. [36].

Table C.15: Sensitivity analysis on the time windows width (group A).

	$\Gamma = -0.1$			$\Gamma = 0.0$			$\Gamma = 0.1$		
Name	z	rc	ac	z	rc	ac	z	rc	ac
MA-01	13651.0	3651.0	10000.0	10955.7	2955.7	8000.0	9733.6	2733.6	7000.0
MA-02	25570.1	5570.1	20000.0	19917.8	4917.8	15000.0	18408.4	4408.4	14000.0
MA-03	33106.1	7106.1	26000.0	28308.9	6308.9	22000.0	23611.6	5611.6	18000.0
LA-01	37132.1	8132.1	29000.0	32545.2	7545.2	25000.0	28561.1	7561.1	21000.0
LA-02	63582.6	13582.6	50000.0	50710.1	10710.1	40000.0	40978.1	8978.1	32000.0

Appendix C. Detailed results on the sensitivity analyses

Tables C.14-C.17 gives detailed results of the sensitivity analyses. The tables reports the total solution cost ("z"), the routing cost ("rc"), the area cost ("ac"), and the number of areas ("$\#ar$").

Table C.14: Sensitivity analysis on the value of F.

	$F = 1000$				$F = 0$			
Name	z	rc	ac	$\#ar$	z	rc	ac	$\#ar$
MA-02	19917.8	4917.8	15000.0	15	21872.2	4872.2	17000.0	17
MA-03	28308.9	6308.9	22000.0	22	28826.4	4826.4	24000.0	24
LA-01	32545.2	7545.2	25000.0	25	34231.1	7231.1	27000.0	27
MB-02	23099.2	6099.2	17000.0	17	25654.1	5654.1	20000.0	20
MC-01	13779.8	4779.8	9000.0	9	14745.2	4745.2	10000.0	10
MC-03	23441.8	7441.8	16000.0	16	24366.3	7366.3	17000.0	17
LC-01	30374.2	10374.2	20000.0	20	32923.6	9923.6	23000.0	23
MD-03	22424.8	5424.8	17000.0	17	24266.3	5266.3	19000.0	19

Table C.16: Sensitivity analysis on the time windows width (group D).

	$\Gamma = -0.1$			$\Gamma = 0.0$			$\Gamma = 0.1$		
Name	z	rc	ac	z	rc	ac	z	rc	ac
MD-01	2372.2	372.2	2000.0	1280.0	280.0	1000.0	1277.7	277.7	1000.0
MD-02	2519.2	519.2	2000.0	2515.4	515.4	2000.0	2512.4	512.4	2000.0
MD-03	3675.5	675.5	3000.0	2611.0	611.0	2000.0	2588.4	588.4	2000.0
LD-01	4808.8	808.8	4000.0	3702.6	702.6	3000.0	3689.8	689.8	3000.0
LD-02	7102.9	1102.9	6000.0	6004.8	1004.8	5000.0	4927.3	927.3	4000.0

Table C.17: Sensitivity analysis on the customer frequencies.

		$\Gamma = 0.0$			$\Gamma = 0.1$		
	Name	z	rc	ac	z	rc	ac
$\Upsilon = 0.0$	SD-01	1280.0	280.0	1000.0	1277.7	277.7	1000.0
	SD-02	2515.4	515.4	2000.0	2512.4	512.4	2000.0
	SD-03	2611.0	611.0	2000.0	2588.4	588.4	2000.0
	SD-04	3702.6	702.6	3000.0	3689.8	689.8	3000.0
$\Upsilon = 0.10$	SD-01	1275.2	275.2	1000.0	1273.3	273.3	1000.0
	SD-02	2462.5	462.5	2000.0	1412.1	412.1	1000.0
	SD-03	2572.3	572.3	2000.0	2540.7	540.7	2000.0
	SD-04	3656.2	656.2	3000.0	2597.6	597.6	2000.0
$\Upsilon = 0.15$	SD-01	1274.3	274.3	1000.0	1272.2	272.2	1000.0
	SD-02	2461.5	461.5	2000.0	1417.5	417.5	1000.0
	SD-03	2555.6	555.6	2000.0	2534.9	534.9	2000.0
	SD-04	3654.4	654.4	3000.0	2600.3	600.3	2000.0

References

[1] N. Christofides, J. Beasley, The period routing problem, Networks 14 (1984) 237–256.

[2] M. Gaudioso, G. Paletta, A heuristic for the periodic vehicle routing problem, Transportation Science 26 (1992) 86–92.

[3] H. Zhong, R. W. Hall, M. Dessouky, Territory planning and vehicle dispatching with driver learning, Transportation Science 41 (2007) 74–89.

[4] K. Smilowitz, M. Nowak, T. Jiang, Workforce management in periodic delivery operations, Transportation Science 47 (2013) 214–230.

[5] M. Schneider, A. Stenger, F. Schwahn, D. Vigo, Territory-based vehicle routing in the presence of time-window constraints, Transportation Science 49 (2014) 732–751.

[6] J. G. Carlsson, Dividing a territory among several vehicles, INFORMS Journal on Computing 24 (2012) 565–577.

[7] J. G. Carlsson, E. Delage, Robust partitioning for stochastic multivehicle routing, Operations research 61 (2013) 727–744.

[8] J. Kalcsics, S. Nickel, M. Schröder, Towards a unified territorial design approach – applications, algorithms and gis integration, Top 13 (2005) 1–56.

[9] J. Kalcsics, Districting problems, in: G. Laporte, S. Nickel, F. Saldanha da Gama (Eds.), Location Science, Springer International Publishing, Cham, 2015, pp. 595–622.

[10] G. Laporte, S. Nickel, F. Saldanha da Gama, Location Science, Springer International Publishing, Cham, 2015.

[11] B. Golden, S. Raghavan, E. Wasil, The Vehicle Routing Problem: Latest Advances and New Challenges, volume 43, Springer US, Boston, MA, 2008.

[12] P. Toth, D. Vigo, Vehicle routing: problems, methods, and applications, second edition ed., Society for Industrial and Applied Mathematics : Mathematical Optimization Society, Philadelphia, 2014.

[13] E. J. Beltrami, L. D. Bodin, Networks and vehicle routing for municipal waste collection, Networks 4 (1974) 65–94.

[14] P. M. Francis, K. R. Smilowitz, M. Tzur, The period vehicle routing problem and its extensions, in: The vehicle routing problem: latest advances and new challenges, Springer, 2008, pp. 73–102.

[15] S. Irnich, M. Schneider, D. Vigo, Chapter 9: Four variants of the vehicle routing problem, in: Vehicle Routing: Problems, Methods, and Applications, Second Edition, SIAM, 2014, pp. 241–271.

[16] A. M. Campbell, J. H. Wilson, Forty years of periodic vehicle routing, Networks 63 (2014) 2–15.

[17] T. Vidal, T. G. Crainic, M. Gendreau, N. Lahrichi, W. Rei, A hybrid genetic algorithm for multidepot and periodic vehicle routing problems, Operations Research 60 (2012) 611–624.

[18] C. F. Daganzo, The distance traveled to visit n points with a maximum of c stops per vehicle: An analytic model and an application, Transportation Science 18 (1984) 331–350.

[19] Y. Ouyang, C. F. Daganzo, Discretization and validation of the continuum approximation scheme for terminal system design, Transportation Science 40 (2006) 89–98.

[20] H. Calik, M. Labbé, H. Yaman, p-center problems, in: G. Laporte, S. Nickel, F. Saldanha da Gama (Eds.), Location Science, Springer International Publishing, Cham, 2015, pp. 79–92.

[21] M. G. Sandoval, J. A. Díaz, R. Z. Ríos-Mercado, An improved exact algorithm for a territory design problem with p-center-based dispersion minimization, Expert Systems with Applications 146 (2020) 113150.

[22] S. Moreno, J. Pereira, W. Yushimito, A hybrid k-means and integer programming method for commercial territory design: a case study in meat distribution, Annals of Operations Research 286 (2020) 87–117.

[23] M. A. Solana, J. A. Díaz, D. E. Luna, Math-heuristic for a territory design problem, in: C. Paternina-Arboleda, S. Voß (Eds.), Computational Logistics, Springer International Publishing, Cham, 2019, pp. 67–82.

[24] M. A. Salazar-Aguilar, R. Z. Ríos-Mercado, M. Cabrera-Ríos, New models for commercial territory design, Networks and Spatial Economics 11 (2011) 487–507.

[25] R. Z. Ríos-Mercado, Optimal Districting and Territory Design, Springer International Publishing, 2020.

[26] I. Sungur, Y. Ren, F. Ordóñez, M. Dessouky, H. Zhong, A model and algorithm for the courier delivery problem with uncertainty, Transportation Science 44 (2010) 193–205.

[27] A. M. Rodrigues, J. S. Ferreira, Sectors and routes in solid waste collection, in: J. P. Almeida, J. F. Oliveira, A. A. Pinto (Eds.), Operational Research, Springer International Publishing, Cham, 2015, pp. 353–375.

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

[29] A. A. Kovacs, B. L. Golden, R. F. Hartl, S. N. Parragh, The generalized consistent vehicle routing problem, Transportation Science 49 (2015a) 796–816.

[30] A. Subramanyam, C. E. Gounaris, A decomposition algorithm for the consistent traveling salesman problem with vehicle idling, Transportation Science 52 (2018) 386–401.

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

[32] A. A. Kovacs, B. L. Golden, R. F. Hartl, S. N. Parragh, Vehicle routing problems in which consistency considerations are important: A survey, Networks 64 (2014) 192–213.

[33] H. Lei, G. Laporte, Y. Liu, T. Zhang, Dynamic design of sales territories, Computers & Operations Research 56 (2015) 84–92.

[34] H. Lei, R. Wang, G. Laporte, Solving a multi-objective dynamic stochastic districting and routing problem with a co-evolutionary algorithm, Computers & Operations Research 67 (2016) 12–24.

[35] M. Bender, A. Meyer, J. Kalcsics, S. Nickel, The multi-period service territory design problem - an introduction, a model and a heuristic approach, Transportation Research Part E: Logistics and Transportation Review 96 (2016) 135–157.

[36] I. Rodríguez-Martín, J.-J. Salazar-González, H. Yaman, The periodic vehicle routing problem with driver consistency, European Journal of Operational Research 273 (2019) 575–584.

[37] M. Bender, J. Kalcsics, Multi-period service territory design, in: R. Z. Ríos-Mercado (Ed.), Optimal Districting and Territory Design, Springer International Publishing, Cham, 2020, pp. 129–152.

[38] H. Derbel, B. Jarboui, S. Hanafi, H. Chabchoub, Genetic algorithm with iterated local search for solving a location-routing problem, Expert Systems with Applications 39 (2012) 2865–2871.

[39] L. Zhou, R. Baldacci, D. Vigo, X. Wang, A multi-depot two-echelon vehicle routing problem with delivery options arising in the last mile distribution, European Journal of Operational Research 265 (2018) 765–778.

[40] M. Gendreau, J.-Y. Potvin, Handbook of Metaheuristics, volume 272, Springer International Publishing, Cham, 2019.

[41] A. E. Eiben, J. E. Smith, Introduction to evolutionary computing, 2. ed ed., Springer, Heidelberg, 2015.

[42] T. Vidal, T. G. Crainic, M. Gendreau, C. Prins, Heuristics for multi-attribute vehicle routing problems: A survey and synthesis, European Journal of Operational Research 231 (2013) 1–21.

[43] J. Mulvey, M. Beck, Solving capacitated clustering problems, European Journal of Operational Research 18 (1984) 339–348.

[44] Y. Chu, T. Liu, On the shortest arborescence of a directed graph, Scientia Sinica 14 (1965) 1396–1400.

[45] M. M. Solomon, Algorithms for the vehicle routing and scheduling problems with time window constraints, Operations Research 35 (1987) 254–265.

[46] M. W. P. Savelsbergh, Local search in routing problems with time windows, Annals of Operations Research 4 (1985) 285–305.

[47] M. Gendreau, A. Hertz, G. Laporte, M. Stan, A generalized insertion heuristic for the traveling salesman problem with time windows, Operations Research 46 (1998) 330–335.

[48] S. Kirkpatrick, C. Gelatt, M. Vecchi, Optimization by simulated annealing, Technical Report, 1982.

[49] D. Delahaye, S. Chaimatanan, M. Mongeau, Simulated annealing: From basics to applications, in: M. Gendreau, J.-Y. Potvin (Eds.), Handbook of Metaheuristics, Springer International Publishing, Cham, 2019, pp. 1–35.

[50] S. Lin, B. W. Kernighan, An effective heuristic algorithm for the traveling-salesman problem, Operations Research 21 (1973) 498–516.

[51] N. Hansen, The CMA evolution strategy: a comparing review, in: J. Lozano, P. Larranaga, I. Inza, E. Bengoetxea (Eds.), Towards a new evolutionary computation. Advances on estimation of distribution algorithms, Springer Berlin Heidelberg, 2006, pp. 75–102.

Author statement

Lin Zhou: Conceptualization, Methodology, Software, Writing

Lu Zhen: Conceptualization, Reviewing and Editing, Funding acquisition

Roberto Baldacci: Conceptualization, Methodology, Writing, Project administration

Marco Boschetti: Methodology, Reviewing and Editing

Ying Dai: Methodology, Reviewing and Editing

Andrew Lim: Reviewing and Editing, Funding acquisition