Territorial design for customers with demand frequency

Highlights

- An MIP model is formulated for territorial design problem.
- The problem is based on demand frequency of customers.
- A novel column generation solution method is proposed.
- Some algorithmic acceleration techniques are implemented.
- The solution time is 55% shorter than that of CPLEX.

Territorial design for customers with demand frequency

Lu Zhen [a], Jiajing Gao [a], Zheyi Tan [a], Gilbert Laporte [b], Roberto Baldacci [c]

[a] *School of Management, Shanghai University, Shanghai, China*
[b] *Department of Decision Sciences, HEC Montréal, Montréal, Canada*
[c] *College of Science and Engineering, Hamad Bin Khalifa University, Doha, Qatar*

Abstract: Territorial design is an important long-term decision for urban delivery service companies, in contexts where customers are partitioned into districts. This study focuses on a territorial design problem given the demand frequency of each customer, i.e., the estimated percentage of days with demand, over the planning horizon. This study formulates a set partitioning model and designs a column generation based algorithm to solve the problem. The algorithm decomposes the original problem into a restricted master problem (RMP) and a series of pricing problems (PPs), each limited to one district. A dynamic programming based method is designed to solve the PPs efficiently. To further accelerate the solution processes of the PPs and of the RMP, some tailored strategies are also embedded within the algorithm. Numerical experiments are conducted to validate the contributions of the dynamic programming and of the acceleration strategies. Some tests based on real-world cases are also performed in order to derive some managerial insights to support the practitioners' decisions on service territory design.

Keywords: OR in service industries; territorial design problem; set partitioning; column generation.

1 Introduction

The last-mile delivery activities are usually conducted in districts, usually defined on the basis of the administrative regions established by governments, or of the polygons formed by some main streets. A good territorial design should be based on the joint optimization of the districting plan and of the periodic routes within each district (Kalcsics et al., 2005; Kalcsics and Ríos-Mercado, 2019). The districting plan is a long-term decision which does not change over the planning horizon. However, customer demands over the planning horizon are uncertain, which may affect the delivery routes. Here uncertainty is reflected by the fact that the days for which a customer has a demand are unknown. Such a decision problem under uncertainty is intractable if formulated as a standard two-stage stochastic programming model with a multiplicity of demand scenarios. If there are n customers and t days in the planning horizon, then there are 2^{nt} possible scenarios, and it is also difficult to calibrate the probabilities of the scenarios. Hence this paper proposes an alternative way of modeling this territorial design problem as a deterministic model. A first step is to analyze the historical data for each customer and estimate the number of days f_i on which customer i has a demand over the planning horizon. Then the territorial design problem is to partition the customers into districts and construct a delivery route in each district f_i times over the planning horizon, with the aim of minimizing the sum of the total travel cost and of the fixed cost related to the number of districts. In addition, this problem also considers some realistic factors such as the maximum daily working time, the carrying capacity for each district, the customer preferred service time window on each day, and a preferred set of days to be served. These preferences can be calibrated on the basis of historical data.

This paper investigates a territorial design problem on the basis of the estimated information about each customer's demand frequency, i.e., the percentage of days with demand over the planning horizon. We formulate a set partitioning model, and we design a column generation algorithm. The algorithmic efficiency is validated through extensive numerical experiments, from which some managerial insights are derived to guide the service territory design.

2 Related works

This paper studies an optimization problem combining territory partitioning (Carlsson, 2012) and routing (Carlsson and Delage, 2013). It proposes a model and an algorithm for the partitioning of customers into districts and the routing within the districts over multiple periods of planning horizon in each district under some operational constraints. The first problem is referred to as "territorial design" (Kalcsics et al., 2005), "territory planning" (Zhong et al., 2007), or "districting" (Kalcsics and Ríos-Mercado, 2019; Rios-Mercado, 2020), while the second problem is similar to the periodic vehicle routing problem (Cordeau et al.,1997;Francis et al., 2006). This section reviews the literature on each of these two streams.

The territorial design problem can be formulated as a continuous model (Daganzo, 1984; Ouyang and Daganzo, 2006) or as a discrete model (Kalcsics and Ríos-Mercado, 2019). Continuous models usually favor contiguity and compactness of the districts but do not take routing decisions into account (Laporte et al., 2019). For a delivery service company, the shape of the districts may affect the routing cost. Discrete models can be further classified into two categories: one category does not consider routing decisions, such as the *p*-median or the *p*-center problem (Elloumi et al., 2004; Mladenović et al., 2007), and the other category considers routing decisions (Bard and Jarrah, 2009; Carlsson and Delage, 2013; Sandoval et al., 2022). Different from the above studies that apply to a single-period context, this study concerns discrete territorial design problem with routing considerations over a multi-period planning horizon. A simple solution method for such problems is to solve them in two stages (Sungur et al., 2010; Schneider et al., 2014), but it preferable to optimize the partitioning and the routing simultaneously (Smilowitz et al., 2013). In this spirit, Lei et al. (2015) proposed such an algorithm that also considers multiple depots and territory similarity in subsequent periods. Lei et al. (2016) further extended the above work by including uncertainties and multi-objectives. In the context of districting under uncertainty,

Haugland et al. (2007) proposed a two-stage stochastic programing model for a vehicle routing and districting problem with stochastic demands; a tabu search algorithm was designed and shown to be superior to other methods such as multistart heuristics. Lei et al. (2012) also formulated a two-stage model for the routing and districting problem under uncertainty; a large neighborhood search (LNS) heuristic was designed to solve their model. Nikzad et al. (2021) investigated a specific districting, resource assignment and routing problem under uncertainties in the context of home health care planning; the uncertainties include the random travel and service times; a two-stage model as well as a four-phase matheuristic was implemented to solve the model. Darmian et al. (2021) proposed a robust optimization model for a health services districting problem considering demand uncertainty; in order to solve the problem efficiently, an improved genetic algorithm was developed in accordance with the graph-based problem nature. The study considers customer service frequency and differs from the above three papers in which each customer is visited in every period. In the multi-period territorial design problem by Bender et al. (2016), Bender et al. (2018), Bender and Kalcsics (2020), the customers need not be visited every day, but specify on which days they should be visited, while in our study, they only specify a frequency of visits, i.e., the number of visiting days per week. For the multi-period problem, Bender et al. (2018) designed an exact algorithm with an accelerated pricing method and symmetry reduction techniques. Bender et al. (2020) proposed a two-stage mathematical model that determines districts in the first stage and adapts them to the daily demand realizations in the second stage, and apply their methodology to a European parcel delivery company.

Regarding the second related research stream, the periodic vehicle routing problem was introduced by Beltrami and Bodin (1974). This problem optimizes routes over multi-period planning horizon with customers requiring multiple visits. Cordeau et al. (1997) and Francis et al. (2006) solved a periodic vehicle routing problem with service frequency exactly and heuristically. Wen et al. (2010) studied a multi-period vehicle routing problem in which customer orders and their feasible service periods are dynamically revealed over a multi-period time horizon. More recent papers also consider service consistency. Groër et al. (2009) defined the consistent vehicle routing problem by addressing the time consistency and driver consistency, which states each customer must be served by the same driver over time, which is similar to a districting decision. For the problem, Tarantilis et al. (2012) designed a tabu search algorithm, and Kovacs et al. (2014) designed an adaptive LNS algorithm to iteratively improve the routing decisions embedded within the consistent vehicle routing problems. Kovacs et al. (2015) further relaxed the driver consistency, so that each customer can be serviced by a few drivers instead of only one. Similarly, Luo et al. (2015) also assumed that each customer can be served by at most a certain number of different vehicles over the multi-period time horizon. In Rodríguez-Martín et al. (2019), each customer is visited by the same vehicle on each visit, and the problem is solved exactly by branch-and-cut. Mancini et al. (2021) consider both visiting time consistency and service consistency over a multi-period time horizon. Wang et al. (2022) incorporate a more comprehensive set of consistency constraints (i.e., time, driver, and route consistencies) in the periodical vehicle routing problem and develop an exact column-and-cut generation algorithm. Zhen et al. (2020) considered both delivery and pickup in a consistent vehicle routing problem. Uncertainty has also been taken into account in some studies. For example, Alinaghian et al. (2018) propose a bi-objective model for a periodic vehicle routing problem in an uncertain competitive environment, in which the time needed to reach the customers affects market share, and the travel time between pairs of customers is uncertain; a multi-objective particle swarm optimization algorithm was developed to solve the model. For a similar problem, Salamatbakhsh-Varjovi et al. (2018) proposed an improved differential evolution algorithm to obtain a robust periodic vehicle routing plan with time windows under uncertainty for companies competing to provide services to customers.

This study differs significantly from the above contributions. It proposes a mathematical model that considers customer demand frequencies and optimizes both the territorial design and the routing in each district. To our knowledge, the study by Zhou et al. (2021) is the most similar to this paper. However, the difference between our study and that of Zhou et al. is that we establish a mixed integer programming model explicitly; in addition, we design a more advanced algorithm.

3 Problem description

This study investigates a territorial design problem with given customer demand frequencies. The problem is formulated on a complete directed graph $G = (V', A)$ with a planning horizon. Let $T = \{1, \ldots, |T|\}$ be the set of days of the planning horizon of $|T|$ days. Here $V' = \{0\} \cup V$, and vertex 0 represents the depot, the set $V = \{1, \ldots, |V|\}$ of customers is indexed by i. Let $P = \{1, \ldots, |P|\}$ be an index set of $|P|$ districts, where each district $p \in P$ is indexed by p. The vehicle of district p has a capacity Q_p, a maximum working time L_p, and a fixed cost F_p. The fixed cost F_p represents the cost of hiring setting district p during the planning horizon. The fleet of $|P|$ vehicles is available at the depot on each day of the planning horizon. For each arc $(i,j) \in A$, the travel cost and duration for district p is denoted by c_{ij}^p and d_{ij}^p, respectively. For ease of reading, we summarize all the indices, sets, parameters, and variables used in this paper in Appendix F.

Each customer $i \in V$ specifies

(1) a service frequency f_i, which means that customer i needs be visited on f_i days in the planning horizon;

(2) a set C_i of allowable day-combinations of f_i visit days;

(3) a quantity q_i^t of commodity that customer i must receive if visited on day t;

(4) a service time s_i^t for serving customer i on day t;

(5) a time window $[e_i^t,\ l_i^t]$ for visiting customer i on day t;

(6) a subset $P_i \subseteq P$ of allowable vehicles for visiting customer i.

Some remarks are useful at this point. The consideration of P_i is mainly based on a realistic context in which each customer may have their preferred couriers (the concept "courier" is similar to the concept of "vehicle" in this problem). The set P_i is mainly used to generate the route templates that are elaborated later, but does not influence the proposed mathematical model directly.

The visit days of a day-combination are represented by a column in a binary matrix $[a_{ts}]$, which has $|T|$ rows, and where $a_{ts} = 1$ if and only if day t is an allowable visit day in day-combination s. Hereafter, we assume that C_i is the index set of those columns of matrix $[a_{ts}]$ corresponding to the allowable day-combinations of customer $i \in V$. Let $V_t \in V$ be the subset of customers that can be visited on day $t \in T$, i.e., $V_t = \{i \in V : \sum_{s \in C_i} a_{ts} \geq 1\}$. The relationship between the binary matrix $[a_{ts}]$ and the set C_i of allowable day-combinations of f_i visit days is illustrated in Figure 1.

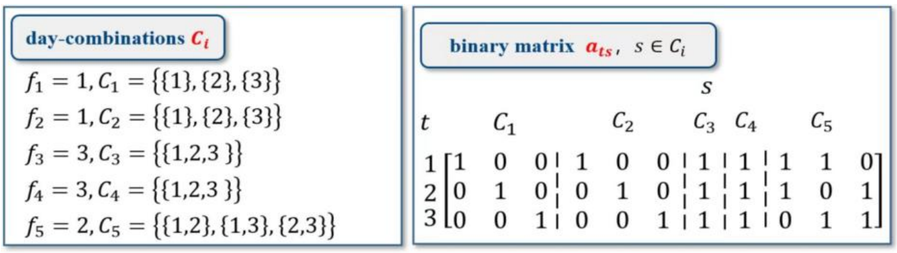


Figure 1: An example for illustrating relationship between set C_i and matrix $[a_{ts}]$

A feasible route on a day $t \in T$ for district $p \in P$ is a simple circuit in G passing through the depot and a subset of customers of $V_t \cap \{i \in V : p \in P_i\}$ such that

(1) the sum of the customer demands is less than or equal to Q_p;

(2) each customer of the route is visited within its time windows; if a vehicle arrives at i on day t before e_i^t, the start of service is delayed to time e_i^t;

(3) 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_p.

Our model uses the concept of "route template". A route template is a feasible route for a district in one day; the route template can be used in a district over several days. We define R^{pt} as the set of all route templates of day $t \in T$ for district $p \in P$. In addition, given a route, its cost is the sum of the costs c_{ij}^p of the arcs $(i, j) \in A$ traversed by the route.

The territorial design problem consists of assigning every customer $i \in V$ to exactly one district $p \in P$ and designing at most $|P|$ feasible routes for each day $t \in T$ (each district has at most one route for each day in the planning horizon), so that each customer $i \in V$ is visited f_i times, according to a feasible day-combination, by the vehicle associated to its district. In this problem, the core decision is the assignment of customers to districts, which is a long-term decision. Each vehicle serves a district with a fixed set of customers for the planning horizon.

Possible objectives of this problem can be: (1) minimizing the sum of the route costs, plus the sum of the fixed costs of the districts selected; (2) minimizing the number of districts opened, and then the sum of the route costs (i.e., fixed costs F_p for district $p \in P$ is assumed to be same).

Based on the above description, a toy example of the territorial design problem is shown in Figure 2, which depicts a territory divided into two districts. Each district has a vehicle that starts from a depot, serves a fixed group of customers, and finally returns to the depot. The planning horizon has three days. The details of the eight customers' service frequencies and day-combinations are as follows: $f_1 = 1, C_1 = \{\{1\},\{2\},\{3\}\}$, $f_2 = 1, C_2 = \{\{1\},\{2\},\{3\}\}$, $f_3 = 3, C_3 = \{\{1,2,3\}\}$, $f_4 = 3, C_4 = \{\{1,2,3\}\}$, $f_5 = 2, C_5 = \{\{1,2\},\{1,3\},\{2,3\}\}$, $f_6 = 2, C_6 = \{\{1,2\},\{1,3\},\{2,3\}\}$, $f_7 = 2, C_7 = \{\{1,2\},\{1,3\},\{2,3\}\}$, $f_8 = 3, C_8 = \{\{1,2,3\}\}$. Twelve route templates are randomly generated and shown in Figure 3. According to the territorial design presented in Figure 2, four templates among the above twelve ones are selected for two districts in three days. One district uses template (8) to serve the orange district for three days; another district uses template (5), template (2), and template (11) to serve the brown district for day 1, day 2, and day 3, respectively.

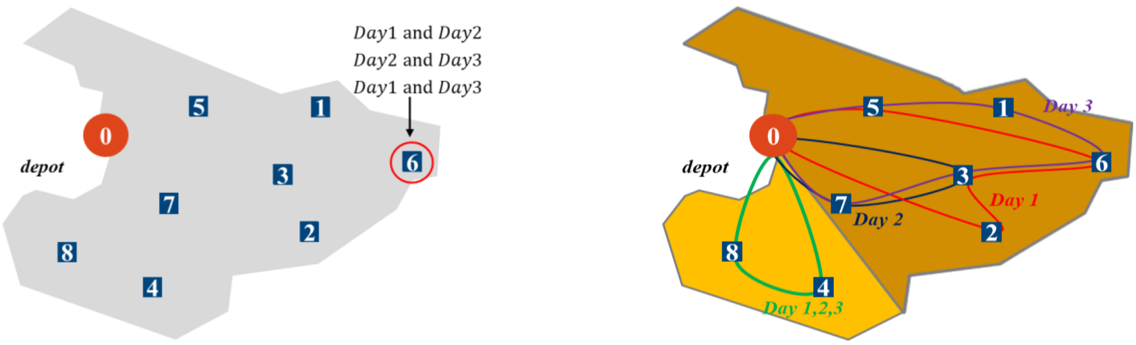


Figure 2: An example of territorial design problem

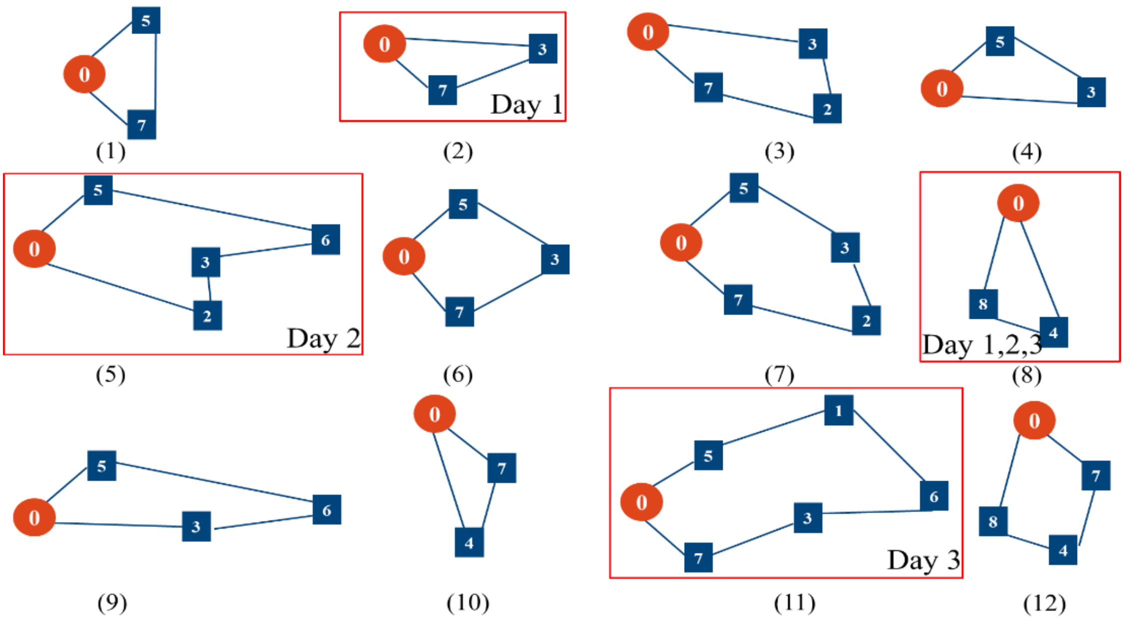


Figure 3: Example of twelve possible route templates and four selected templates

4 Set partitioning based mathematical model

To solve the above described territorial design problem efficiently, we first formulate a set partitioning based mathematical model and apply column generation to solve it. In the field of the algorithm design for the districting problems, a number of algorithms have been developed; some customized algorithms are proposed for specific problem contexts (Ríos-Mercado, 2020). Among these algorithms, this study adopts column generation as the methodology for developing an algorithm that can well fit the problem features and solve the instances efficiently. Mehrotra et al. (1998) is the first study and also one of the very few studies to use column generation for districting problems; their proposed method can yield the set of compact and contiguous districts without subjective intervention. The combination of routes in districts and days constitute the final plan for the problem, which is similar as the combination of selected columns forms the final solution for a set partitioning problem.

Recall that R^{pt} is the set of all routes of day $t \in T$ for district $p \in P$, and let R_i^{pt} be the index set of the routes of day $t \in T$ for district $p \in P$ covering customer $i \in V$. We use c_r to indicate the cost of route $r \in R^{pt}$. Here the set R^{pt} is an important input data for the set partitioning based model. A detailed procedure for generating this set is elaborated in Appendix A.

Let ξ_i^p be a binary variable equal to 1 if and only if customer $i \in V$ is assigned to district $p \in P$. This variable reflects the territorial design decision, and is mainly determined by another binary variable x_r, equal to one if and only if route $r \in R^{pt}$ on day $t \in T$ for district $p \in P$ is selected in the solution. Besides the binary variable x_r, i.e., the core decision for the territorial design problem, some other decision variables are defined as follows. Let y_{is} be a binary variable equal to one if and only if day-combination $s \in C_i$ is assigned to customer $i \in V$; let z_p be a binary variable equal to one if and only if district $p \in P$ is used. Based on the previously defined variables and parameters, the territorial design problem model (TDP) can be formulated as follows:

[**TDP**] Minimize $\sum_{p\in P}\sum_{t\in T}\sum_{r\in R^{pt}} c_r x_r + \sum_{p\in P} F_p z_p$ (1)

subject to

$\sum_{t\in T}\sum_{r\in R_i^{pt}} x_r = f_i \xi_i^p \quad i \in V, p \in P$ (2)

$\sum_{p\in P}\sum_{r\in R_i^{pt}} x_r = \sum_{s\in C_i} a_{ts} y_{is} \quad i \in V, t \in T$ (3)

$\sum_{r\in R^{pt}} x_r \le 1 \quad p \in P, t \in T$ (4)

$\sum_{p\in P} \xi_i^p = 1 \quad i \in V$ (5)

$\sum_{i\in V} \xi_i^p \le |V| z_p \quad p \in P$ (6)

$\xi_i^p \in \{0,1\} \quad i \in V, p \in P$ (7)

$z_p \in \{0,1\} \quad p \in P$ (8)

$x_r \in \{0,1\} \quad p \in P, t \in T, r \in R^{pt}$ (9)

$y_{is} \in \{0,1\} \quad s \in C_i, i \in V.$ (10)

Objective (1) minimizes the sum of the fixed costs of the district used and the operating costs related to the length of all the operated routes. It should be noted that the two parts (routes' cost and fixed cost of districts) in the objective should be comparable, which implies they should relate to the same time horizon. Constraints (2) ensure that each customer i is visited exactly f_i times. Constraints (3) guarantee that every customer i is visited exactly f_i times in the f_i days of the day-combination, which is assigned to the customer. Constraints (4) state that a solution contains at most one route on each day for each district. Constraints (5) ensure a customer is assigned to exactly one district. Constraints (6) link the variables ξ_i^p and z_p. Constraints (7)–(10) define the domains of the variables.

5 Solution methodology

This section proposes a column generation algorithm for large-scale instances of the problem. The flow of the solution method is first described. The four main components embedded in the method (i.e., the master problem, the pricing problem, the algorithm for the pricing problem, and the strategies for constructing feasible solutions) are elaborated in Sections 5.1–5.4, respectively. Some strategies for accelerating the solving process are proposed in Section 5.5. The whole solution method that integrates the above components is summarized in Section 5.6. Before elaborating on the embedded components, the algorithmic framework is outlined in Figure 4.

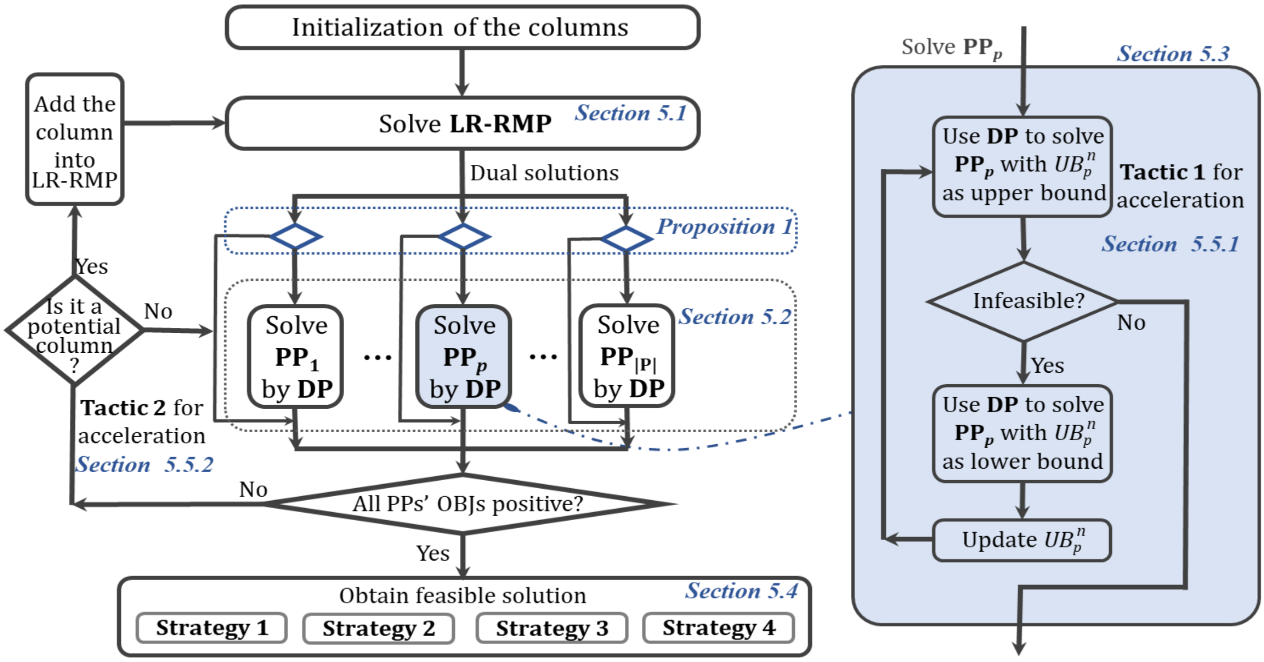


Figure 4: Flowchart of the column generation algorithm

5.1 Restricted master problem

In order to apply the Dantzig-Wolfe decomposition to reformulate the original model described in the previous section, the variable y_{is} is replaced with a newly defined binary variable y_{isp}, equal to one if and only if day-combination $s \in C_i$ is assigned to customer i in district p. In addition, Constraints (3) are replaced with Constraints (11). The variable y_{isp} is equal to zero if customer i is not assigned to district p, which is ensured by the newly added Constraints (12):

$$\sum_{r\in R_i^{pt}} x_r = \sum_{s\in C_i} a_{ts} y_{isp} \qquad i \in V, t \in T, p \in P \qquad (11)$$

$$y_{isp} \le \xi_i^p \qquad i \in V, s \in C_i, p \in P. \qquad (12)$$

Before formulating the master problem based on Dantzig-Wolfe decomposition, we define a "column" as a feasible assignment plan for one district in the whole planning horizon. More specifically, for district p the set of all the feasible assignment plans is defined as K_p, which is indexed by κ_p; a feasible assignment plan κ_p contains $|T|$ routes for district p in $|T|$ days of the planning horizon. Here the set K_p is an important input data for the column generation based solution method. A detailed procedure for generating the set is elaborated in Appendix B. For the parameters related to assignment plans, the cost of assignment plan κ_p is denoted by parameter c_{κ_p}, whose calculation will be elaborated in Section 5.3; the binary parameter $\xi_i^{\kappa_p}$ is equal to one if and only if customer i is contained in the assignment plan κ_p.

In the master problem, the core decision is the binary variable η_{κ_p}, which equals one if and only if the assignment plan κ_p is used by district p. Based on the above definitions, the master problem can be formulated as follows:

$$[\textbf{MP}] \quad \text{Minimize } \sum_{p\in P} \sum_{\kappa_p \in K_p} c_{\kappa_p} \eta_{\kappa_p} \qquad (13)$$

subject to:

$$\sum_{\kappa_p\in K_p} \eta_{\kappa_p} \le 1 \qquad p \in P \qquad (14)$$

$\sum_{p\in P}\sum_{\kappa_p\in K_p}\xi_i^{\kappa_p}\eta_{\kappa_p}=1 \qquad i\in V \qquad (15)$

$\eta_{\kappa_p}\in\{0,1\} \qquad p\in P,\kappa_p\in K_p. \qquad (16)$

In the master problem, Objective (13) minimizes the total cost of all the selected assignment plans. Constraints (14) ensure that at most one assignment plan is selected for each district. Constraints (15) guarantee that each customer is covered by exactly one selected assignment plan. Constraints (16) define the decision variables for the master problem. According to the usual practice of column generation, the above master problem is reformulated as a restricted master problem (RMP) by using restricted subsets of the assignment plans for districts rather than the complete sets of all the possible plans. In addition, the RMP is linearly relaxed by redefining the binary variable η_{κ_p} as a continuous variable. For the linear relaxed RMP (LR-RMP), the dual variables are defined as follows:

π_p dual variables for Constraints (14), $p\in P$;

θ_i dual variables for Constraints (15), $i\in V$.

The above dual variables are used to formulate the objective of pricing problems (PPs), which generates the columns to be added into the LR-RMP iteratively. Section 5.2 elaborates the formulation on the PPs.

5.2 Pricing problem

According to the feature of the original problem model, the pricing problem can be divided into $|P|$ subproblems, each of which generating columns (assignment plans) for the set K_p. This subsection elaborates the model formulation on the pricing subproblem for district p, and the pricing subproblem model is denoted by PP_p. For the sake of simplicity, the subscript or superscript "p" in the PP_p variables is omitted in the following formulation.

First, the decision variables in the PP_p, which are also the parameters of the columns used in the RMP, are explained as follows. All of these decision variables are binary. More specifically, y_{is} is equal to one if and only if day-combination $s\in C_i$ is assigned to customer i, ξ_i is equal to one if and only if customer i is assigned to the district p, x_r is equal to one if and only if route $r\in R^{pt}$ is in solution, z is equal to one if and only if the district p is used in solution. The pricing problem can be formulated as follows:

[**$\mathbf{PP_p}$**] Minimize $c_{\kappa_p}-(\pi_p+\sum_{i\in V}\theta_i\xi_i) \qquad (17)$

subject to:

$\sum_{t\in T}\sum_{r\in R_i^{pt}}x_r=f_i\xi_i \qquad i\in V \qquad (18)$

$\sum_{r\in R^{pt}}x_r\le 1 \qquad t\in T \qquad (19)$

$\sum_{r\in R_i^{pt}}x_r=\sum_{s\in C_i}a_{ts}y_{is} \qquad i\in V,t\in T \qquad (20)$

$y_{is}\le\xi_i \qquad i\in V,s\in C_i \qquad (21)$

$\sum_{i\in V}\xi_i\le|V|z \qquad (22)$

$c_{\kappa_p}=\sum_{t\in T}\sum_{r\in R^{pt}}c_rx_r+F_pz \qquad (23)$

$\xi_i\in\{0,1\} \qquad i\in V \qquad (24)$

$z\in\{0,1\} \qquad (25)$

$x_r\in\{0,1\} \qquad t\in T,r\in R^{pt} \qquad (26)$

$y_{is} \in \{0,1\}$ $\quad s \in C_i, i \in V.$ (27)

Objective (17) minimizes the reduced cost of the generated columns. Constraints (18) guarantee that customer i is visited f_i times in the assignment plan if it is assigned to district p. Constraints (19) state that the assignment plan contains at most one route on each day. Constraints (20) ensure that every customer i is visited in the f_i days specified by the selected day-combination. Constraints (21) state that if customer i is not assigned to district p, then none of day-combinations will be assigned to customer i. Constraints (22) state that if district p is not created, then no customer is assigned to it. Constraint (23) is the calculation of the cost of the column (assignment plan). Constraints (24)–(27) define the domains of the decision variables.

It should be noted that if we use CPLEX to solve the above PP_p model, we need not to solve it directly. We just need to solve the PP_p model with the z variable equalling one and then compare the solved objective value with the $-\pi_p$, i.e., the objective value when z variable equals zero.

The next subsection proposes a dynamic programming (DP) algorithm to solve the above PP_p model. DP is faster than CPLEX to solve the model. Before solving it, Proposition 1 can be used to save computation effort.

Proposition 1: When $\sum_{i \in V} \theta_i < \min_{t \in T, r \in R^{pt}} \{c_r\} + F_p$, the PP_p model does not need to be solved.

Proof: See Appendix C. ■

5.3 Solving the pricing problem

It may be time consuming to solve the above pricing problem PP_p by using CPLEX directly. In order to further improve the solving efficiency, this study proposes a DP algorithm to solve the pricing problem PP_p. The core idea of the DP algorithm is to decompose the problem into several stages, and solve them one by one. The solution of the former stage provides useful information for the solution of the latter stage. When solving any stage, several possible local solutions are listed, and those that are likely to be optimal are retained, while others are discarded. Each stage is solved in turn, and the last stage outputs a solution to the original problem.

In this study, the total number of customers is the total number of stages. Each stage (indexed by k) corresponds to one customer (indexed by i). The order of the stages is not the increasing or decreasing order of the customer indices, but the decreasing order of the value of the dual variable θ_i. Here these values are obtained from the LR-RMP, and each dual variable θ_i corresponds to one customer i. For example, if there are 10 customers indexed from 1 to 10, then the PP can be divided into 10 stages. According to the decreasing order of the dual variable values, the customers are ordered as customer 1, 3, 8, 4, 6, 7, 5, 9, 10, 2. Then the first, second, third, and last stage correspond to customer 1, 3, 8, and 2, respectively. In the remainder of this subsection, we use $i(k)$ to denote the index of a customer that corresponds to the k^{th} stage.

In each stage, the states are the day-combinations for the customer that corresponds to the stage. For example, suppose the third stage corresponds to customer 8, the service frequency of the customer is two, the set C_8 of allowable day-combinations for the customer is {(0,1,0,0,1,0), (0,0,1,0,0,1), (1,0,0,1,0,0)}. Then we set four states for this stage: state 0 denotes customer 8 is not served; state 1, 2, and 3 corresponds to the above day-combinations (0,1,0,0,1,0), (0,0,1,0,0,1), and (1,0,0,1,0,0), respectively. The number of states in a stage is equal to one, plus the number of allowable day-combinations for the customer who corresponds to the stage.

Some newly defined symbols used in the DP are listed as follows:

S_k the set of states in the k^{th} stage;

Z_{ks} the s^{th} state in the k^{th} stage, $Z_{ks} \in S_k$;

$u_k(Z_{ks})$ the decision at the s^{th} state in the k^{th} stage;

$v_k\left(u_{k-1}(Z_{k-1,s'}), u_k(Z_{ks})\right)$ the value of the reduced cost when the decision $u_{k-1}(Z_{k-1,s'})$ is taken in the $(k-1)^{\text{th}}$ stage and the decision $u_k(Z_{ks})$ is taken in the k^{th} stage.

The calculation of the above value v_k follows Objective (17), i.e., $v_k\left(u_{k-1}(Z_{k-1,s'}), u_k(Z_{ks})\right) = \tilde{c}_{kss'} + F_p - \left(\pi_p + \theta_{i(k)}\xi_{i(k)}\right)$. If the 0^{th} state is chosen in the k^{th} stage, $\xi_{i(k)} = 0$; otherwise $\xi_{i(k)} = 1$. $\tilde{c}_{kss'}$ is the cost of the assignment plan when the decision $u_{k-1}(Z_{k-1,s'})$ is taken in the $(k-1)^{\text{th}}$ stage and the decision $u_k(Z_{ks})$ is taken in the k^{th} stage. The calculation of the above value $\tilde{c}_{kss'}$ follows Constraint (23), i.e., $\tilde{c}_{kss'} = \sum_{t\in T}\sum_{r\in R^t} c_r x_r$. Here c_r is the cost of route r defined in the original problem model TDP; the value of x_r (i.e., a binary decision variable in the PP_p) is determined according to the chosen day-combinations in k stages (from the first stage to the k^{th} stage) and Constraints (18)–(21). In addition, the value of y_{is}, i.e., another decision variable in the PP_p, is determined according to the decision $u_k(Z_{ks})$ is taken in the k^{th} stage, Constraints (18) and (21). Define c'_{1s} as the cost of the assignment plan when the decision $u_1(Z_{1s})$ is taken in the first stage, i.e., $c'_{1s} = \sum_{t\in T}\sum_{r\in R^t} c_r x_r$. Here x_r is determined according to the chosen day-combinations (s^{th} state) in first stages.

The recursive equations of the DP are given as follows. For the first stage, the objective value under different states (s) is calculated as

$$f_1(Z_{1s}) = \begin{cases} 0, & s = 0 \\ c'_{1s} + F_p - \left(\pi_p + \theta_{i(1)}\xi_{i(1)}\right), & s = 1,\cdots,|C_{i(1)}| \end{cases}.$$

For the following stages (e.g., the k^{th} stage, $k = 2,\cdots,|V|$), the objective value under different states (e.g., the state s, $s = 0,1,\cdots,|C_{i(k)}|$) is calculated as

$$f_k(Z_{ks}) = \min_{\substack{x=0,1,\ldots,|C_{i(k-1)}| \\ Z_{k-1,x}\in S_{k-1}}} \left\{ v_k\left(u_{k-1}(Z_{k-1,x}), u_k(Z_{ks})\right) - \min_{s'=0,1,\ldots,|C_{i(k-2)}|}\left\{\tilde{c}_{(k-1),s,s'}\right\} + \pi_p - F_p + f_{k-1}(Z_{k-1,x}) \right\}.$$

The pseudocode of the procedure describing the detailed process of the dynamic programming of solving the pricing problem PP_p, is provided in Appendix D. ■

5.4 Strategies for the generation of feasible solutions

Since the column generation algorithm solves a linear relaxed model for the original problem, the obtained solution may be infeasible. This subsection proposes four strategies to construct feasible solutions on the basis of the results solved by the column generation procedure.

Strategy 1: Based on the set of columns obtained by solving the LR-RMP and PPs, we use CPLEX to solve the RMP rather than the LR-RMP. The integer solution for the RMP is used to construct a final plan for the territory partitioning and the routing in every district for every day of the planning horizon.

Strategy 2: This strategy is to determine some core variables' values according to the results obtained by the LR-RMP; and then solve the original model with some of the variables fixed at the above values. This strategy is made up of four steps.

Step 1: Based on the solution of the LR-RMP that may be non-integer (i.e., η_{κ_p}), we calculate the probability ϵ_{ip} that each customer i belongs to district p. For each pair of customer i and district p, the ϵ_{ip} is calculated as $\sum_{\kappa_p \in K_p} \xi_i^{\kappa_p} \eta_{\kappa_p}$. Here $\xi_i^{\kappa_p}$ is a parameter of column κ_p.

Step 2: According to the decreasing order of the values of $\{\epsilon_{ip}\}_{i \in V, p \in P}$, we assign a customer to a district one at a time until all the customers have been assigned to a district. During the assignment process, the capacity of each district must be respected. The above process outputs the assignment of customers to districts, i.e., the values of the decision variable ξ_i^p in the original model TDP. The values of variables z_p, which denote whether the district p is used, are also determined.

Step 3: We use CPLEX to solve the original model TDP given the integer values of ξ_i^p and z_p. The results on the route and day-combination of customers corresponding to each district, i.e., the values of variables x_r and y_{is}, are also computed.

Step 4: If CPLEX cannot solve the original model TDP in Step 3, we reassign a customer to a district based on the value of ϵ_{ip}. More specifically, if $\epsilon_{ip} < 1/|P|$ for all customers $i \in V$, we set $z_p = 0$; otherwise $z_p = 1$. We use CPLEX to solve original model TDP given the above determined values of z_p.

Some remarks are in order for the step 4. The threshold for the ϵ_{ip} is $1/|P|$ is explained as follows: the ϵ_{ip} denotes the probability customer i belongs to district p, the average probability a customer belongs to one of the $|P|$ districts is $1/|P|$; thus we use this average value as the threshold.

Strategy 3: The core idea of this strategy is similar to the previous strategy, which is to determine some variables' values and then solve the original model with the variables fixed. The difference from the previous strategy lies in that the fixed variable is different. This strategy is made up of five steps.

Step 1: It is the same as Step 1 in Strategy 2.

Step 2: We use CPLEX to solve a linear relaxed model of TDP with the variable ξ_i^p's value equalling the value of ϵ_{ip}, which may be non-integer.

Step 3: For each customer $i \in V$, the day-combination s with the maximum value of y_{is}, $s \in C_i$ is selected. The integer value for variable y_{is} in the original model TDP is determined.

Step 4: We use CPLEX to solve the original model TDP given the integer values of y_{is}. The results on the remainder integer variables are also solved.

Step 5: If CPLEX cannot solve the original model TDP in Step 4, we reselect the day-combination. For each customer $i \in V$, the value of y_{is} is determined in Step 2, which may be non-integer. If $y_{is} < 1/|C_i|$, we set y_{is} equal 0. We then use CPLEX to solve the original model TDP, in which some of the y_{is} values are fixed as zero. The problem on the remainder integer variables is solved.

In step 5, the threshold for the y_{is} is $1/|C_i|$ is due to: the y_{is} denotes the probability customer i selects the day-combination s, the average probability that a customer (i.e., customer i) selects one of the $|C_i|$ day-combinations is $1/|C_i|$; thus we use this average value as the threshold.

Strategy 4: The core idea of this strategy is also similar to the previous two strategies, while the differences from them are mainly as follows. The manner of fixing-and-solving is a combination of the previous two strategies' manners; here the "manner" includes the set of fixed variables, the thresholds adopted in some judgement conditions, the procedure when the solving process by CPLEX fails, and so on. This strategy is made up of seven steps.

Step 1: It is the same as Step 1 in Strategy 2.

Step 2: For each customer $i \in V$, we assign the customer to a district p with the maximum value ϵ_{ip}. Then the decision variables ξ_i^p as well as z_p in the original model TDP are determined.

Step 3: We use CPLEX to solve a linear relaxed model of TDP with the variables ξ_i^p and z_p equalling the above determined values.

Step 4: If CPLEX cannot solve the linear relaxed model of TDP in Step 3, we reassign a customer to a district based on the value of ϵ_{ip}. If $\epsilon_{ip} < 1/|P|$ for all customers $i \in V$, we set $z_p = 0$; otherwise $z_p = 1$. We use CPLEX to solve a linear relaxed model of TDP given the above determined values of z_p.

Step 5: It is the same as Step 3 in Strategy 3.

Step 6: We use CPLEX to solve the original model TDP given the integer values of ξ_i^p, z_p and y_{is}. The problem with the remaining variable x_r is also solved.

Step 7: If CPLEX cannot solve the original model TDP given the integer values of ξ_i^p, z_p and y_{is}, we change the setting for the value of y_{is}. For each customer $i \in V$, the value of y_{is} is calculated by Step 3 or Step 4. If the value of y_{is} is non-integer and less than $1/|C_i|$, we set $y_{is} = 0$. We then use CPLEX to solve the original model TDP, in which some of the y_{is} values are set equal to zero, the ξ_i^p and z_p are set equal to their previous values. The problem on the remaining variable x_r is also solved.

5.5 Strategies for algorithmic acceleration

Because the pricing problem needs be solved frequently in the column generation procedure, most of the computing time is consumed by it (Cortés et al., 2014). It is therefore important to accelerate the solution process of the pricing problem. The dynamic programming proposed in Section 5.3 is well suited for this purpose. This section further proposes some strategies for shortening the computation time of the pricing problem; in addition, some other strategies are also designed and used in key steps of the solution method so as to accelerate the whole column generation solution process.

5.5.1 Upper bounds based acceleration for solving pricing problems

This strategy is inspired by a hint from the excellent work of Václavík et al. (2018), which uses a machine learning method to tighten up the upper bound on the objective function of the pricing problem in order to accelerate the solving process of the pricing problem. This study also proposes a strategy based on the use of upper bounds to reduce the number of states that need be examined at each stage of the dynamic programming procedure for solving the PP_p.

As aforementioned, the pricing problem needs be solved several times. Let n be the index of the iteration when the pricing problem is being solved; let OBJ_p^n be the pricing problem PP_p's objective value outputted at the n^{th} iteration; suppose the input data for the PP_p at the n^{th} iteration is the dual variables (π_p^n and θ_i^n) delivered from the LR-RMP. A function $f_p^n(\pi, \boldsymbol{\theta}) = e_0\pi_p^n + \sum_{i\in V} e_i\,\theta_i^n$ is defined to approximate the objective value solved by the PP_p; here $\boldsymbol{e}$ is a vector of coefficients, which needs be trained on the basis of the accumulated data in the previous n iterations so that the gap $\sum_{\alpha=1}^{n}\left|f_p^{\alpha}(\pi,\boldsymbol{\theta})-OBJ_p^{\alpha}\right|$ is as small as possible. More specifically, the coefficients, i.e., e_0 and $\{e_i\}_{i\in V}$, are determined by the following model:

$$\min_{e_0\geq 0; e_i\geq 0, i\in V} \sum_{\alpha=1}^{n}\left|e_0\pi_p^{\alpha}+\sum_{i\in V} e_i\,\theta_i^{\alpha}-OBJ_p^{\alpha}\right|. \quad (28)$$

Based on the value of $f_p^n(\pi,\boldsymbol{\theta})$, we define an upper bound $UB_p^n=f_p(\pi,\boldsymbol{\theta})+\epsilon$, which is used at the n^{th} iteration; here ϵ is a small value of redundancy to avoid making the PP_p infeasible. More specifically, during each stage of the dynamic programming, we only examine the states whose objective values are less than UB_p^n. If the upper bound is too restrictive and no feasible solution can be obtained, the above dynamic programming algorithm is run again in a solution space of the states whose objective values are higher than UB_p^n.

5.5.2 Potentiality judgement based acceleration for solving RMP

During the CG procedure, the columns are generated by pricing problems with the objective of minimizing the reduced cost. However, the columns having a low reduced cost may not be chosen in an optimal solution of the RMP; in other words, we should choose the columns that have a large potential to improve the quality of final solution (Zhang et al., 2022). When a column is generated by a pricing problem, we first judge whether or not it is a potential column according to some criterion; if so, the column is added to the RMP. With fewer columns, the time for solving the RMP should also be reduced. This strategy is actually a tradeoff between computation time and solution quality. Therefore, the choice of a good criterion is important.

For this study, we conducted some exploratory tests and found a suitable criterion to determine what should be the range of total travel cost of the routes in an assignment plan (column), i.e., the value of $\sum_{t\in T}\sum_{r\in R^{pt}} c_r x_r$ should belongs to some range $[lb, ub]$. The rationale behind this criterion lies in the fact that assignment plans with too high or too low travel costs are less likely to be contained in an optimal solution, even though these reduced costs are negative; therefore, they are not added to the RMP in order to speed up the solving process. It should be noted that other criteria are possible such as the daily travel cost of the routes on the demand of a district. No criterion is universally better than the others, so the adoption of a suitable criterion heavily depends on the characteristics of each instance (Wolpert and Macready, 1997). The choice of a suitable range $[lb,\ ub]$ can be based on data related to feasible assignment plans in other instances.

5.6 Summary of the solution method

Based on the components elaborated in the previous subsections, the integrated algorithmic flow of the full CG-based solution method is summarized by the following pseudocode. The contributions of each component and strategies used in the solution method will be evaluated through comparative experiments in the next section.

CG based solution method with strategies of acceleration

```
1  Initialization of columns (feasible assignment plan)    // Appendix B
2  Iteration index n ← 0; Initialization on coefficients e_0 and {e_i}_{i∈V}    // Section 5.5.1
3  Do
4        n ← n + 1
5        Solve LR-RMP    // Section 5.1
```

$\{\pi_p^n, \{\theta_i^n\}_{i\in V}\}_{p\in P} \leftarrow$ dual solution of the LR-RMP
For $p = 1, \ldots, |P|$ **do** // Solve pricing subproblems in parallel
  If $\sum_{i\in V} \theta_i < \min_{t\in T, r\in R^{pt}}\{c_r\} + F_p$, **continue** // Proposition 1
  $UB_p^n \leftarrow e_0\pi_p^n + \sum_{i\in V} e_i\,\theta_i^n + \epsilon$ // Section 5.5.1
  Use dynamic programming to solve PP_p with UB_p^n as upper bound // Section 5.3
  If "PP_p with upper bound UB_p^n" is infeasible
    Solve PP_p with UB_p^n as lower bound
  End if
  $\{\boldsymbol{\xi}, \boldsymbol{x}, \boldsymbol{y}, z\} \leftarrow$ solution of the PP_p; $OBJ_p^n \leftarrow$ objective value of the solution
  If $\{\boldsymbol{\xi}, \boldsymbol{x}, \boldsymbol{y}, z\}$ is a potential column // Section 5.5.2
    Add column $\{\boldsymbol{\xi}, \boldsymbol{x}, \boldsymbol{y}, z\}$ into the LR-RMP
  End if
  Update coefficients e_0 and $\{e_i\}_{i\in V}$ by Objective (28) // Section 5.5.1
End for
While column with negative reduced cost exists
Obtain a feasible solution by one strategy // Section 5.4

6 Numerical experiments

Numerical experiments were conducted for four purposes: (i) investigating performance and contribution of the different strategies for the CG-based solution method, (ii) evaluating the quality of the solutions, (iii) delivering some managerial implications through sensitivity analyses, (iv) testing the robustness of the model. We first describe the experimental settings in the next subsection.

6.1 Experimental settings

As shown in Figure 5, the experiments consider a selected list of 80 neighborhoods in the Huangpu district of Shanghai as the locations of the customers in our problem. The Huangpu district has a population of 0.66 million, and an area of 22.52 square kilometers; it is the most central urban district in Shanghai. The depot in Figure 5 is a station for a logistic delivery service provider.

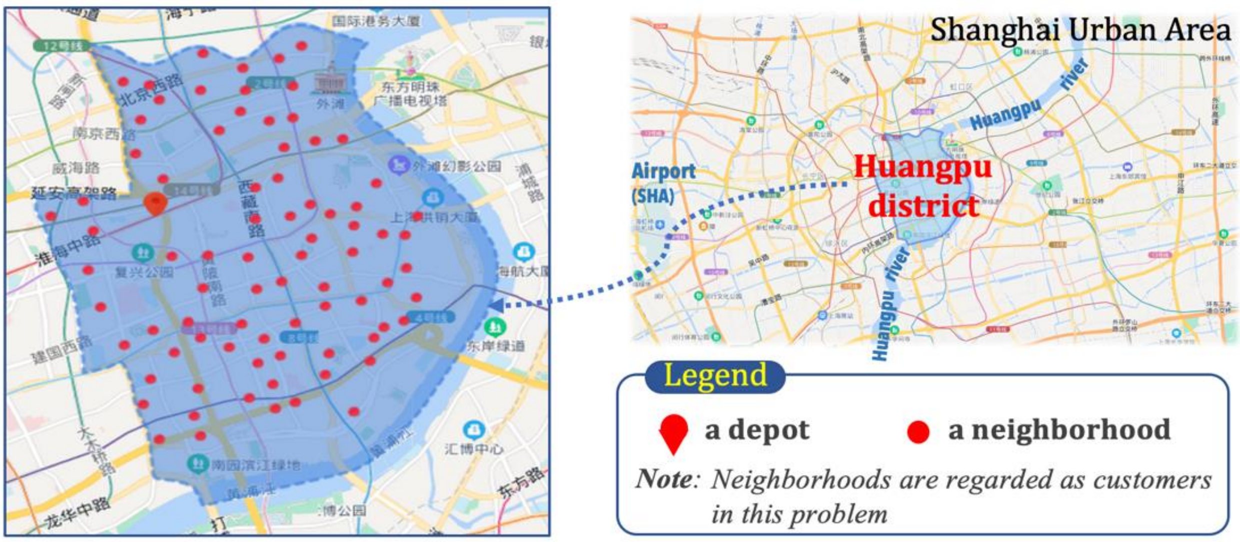


Figure 5: The neighborhoods in the Huangpu district, Shanghai

In the experiments, the value of T is set as six days. The customer service frequency f_i is randomly generated in the $[0, 6]$ interval; each customer's time window $[e_i^t,\ l_i^t]$ is also randomly generated as one of two optional intervals, i.e., [9:00,15:00], [15:00,19:00], which obeys the realistic service guarantee of most delivery service providers in China; the day-combinations of the customers are then determined by their service frequency, as shown in Appendix E. For the set P_i of "preferred" districts for each customer, it is also randomly generated such that 60 to 80% of the districts are contained. According to the data estimated by practitioners in the logistic delivery service provider, the vehicle capacity Q is set as 300 kg; the demand of customers q_i^t is in the $[40\text{ kg}, 60\text{ kg}]$ interval; the fixed cost of the district F_p the $[100\text{ CNY}, 300\text{ CNY}]$ interval. When preparing the set of candidate routes, the maximum number of routes is set at 1000. Each route cost c_r is determined according to a realistic travel distance and the service time. Here the travel distance is the length of a route between two points measured according to a realistic road map, rather than the Euclidean distance. The realistic travel distance of a route times a unit cost (CNY/distance) plus the service time is multiplied by another unit cost (CNY/time) to yield the cost of the route, i.e., c_r. To assess the algorithmic performance under more scales of instances, we generate four groups of problem instances of different sizes. Each group contains 10 instances, which differ from each other with respect to q_i^t, i.e., the quantity of commodity that customer i must receive if visited on day t. The number of districts P ranges from five to 18, and the number of customers N ranges from 20 to 80, as shown in Table 1.

Table 1: Scale of instance groups in experiments

Group ID	Number of customers (N)	Number of districts (P)
ISG1	20	5
ISG2	30	7
ISG3	40	9
ISG4	80	18

All experiments were performed on a workstation with two Xeon E5-2680 V4 CPUs (12 cores) running at 2.4 GHz with 256 GB of memory under Windows 10. The code is implemented in C# Visual Studio 2019. CPLEX with version 12.6.1 is used to solve the original model TDP, RMP, and PPs.

6.2 Evaluating performance of algorithmic components

As mentioned at the beginning of Section 5, some strategies are embedded in the solution method. Comparative experiments were conducted to validate the merits of the proposed strategies and identify which one is appropriate for the solution method.

6.2.1 Evaluating performance of strategies of constructing feasible solutions

Four strategies of constructing feasible solutions (i.e., Strategy 1, 2, 3, 4) were proposed in Section 5.4. Experiments were performed to compare the four strategies' performance with respect to the optimality gap (i.e., $\Delta_{F\#}$) and the solution time (i.e., $t_{\#}$(s)). The relative optimality gap between the solution value obtained by CPLEX and the CG-based solution value was computed. The experiments are based on small-scale ISG 1 and ISG 2. In the first row of Table 2, "CG+F1, F2, F3, and F4" denote the CG-based solution method using Strategy 1, 2, 3, and 4, respectively. It is noted that all the objective values of solutions shown in the tables are for the original model rather than for some linear relaxations or submodels.

Table 2: Comparison among four strategies of constructing feasible solutions

Instances		CPLEX					CG +F1		CG +F2		CG +F3		CG +F4	
Scale	ID	F_{CPLEX}	Num. of activated districts	Fixed cost	Route cost	t_{CPLEX}(s)	Δ_{F1}	t_1(s)	Δ_{F2}	t_2(s)	Δ_{F3}	t_3(s)	Δ_{F4}	t_4(s)
ISG1	1	671	4	436	235	253	**0.00%**	**87**	0.00%	377	0.00%	402	0.00%	151
	2	658	4	436	222	306	0.46%	69	0.00%	332	0.00%	286	**0.00%**	**132**
	3	651	4	436	215	340	0.46%	62	0.00%	416	0.00%	449	**0.00%**	**148**
	4	542	3	327	215	41	0.18%	86	0.00%	87	20.11%	90	0.00%	85
	5	656	4	436	220	310	0.46%	60	0.00%	366	0.00%	374	**0.00%**	**143**
	6	654	4	436	218	608	0.46%	72	0.46%	63	0.00%	405	**0.00%**	**142**
	7	666	4	436	230	301	0.45%	58	0.00%	323	0.00%	351	**0.00%**	**134**
	8	658	4	436	222	344	0.46%	69	0.00%	152	0.00%	188	**0.00%**	**147**
	9	664	4	436	228	253	0.00%	85	**0.00%**	**83**	0.00%	359	0.00%	172
	10	659	4	436	223	257	0.00%	108	0.00%	372	0.00%	397	**0.00%**	**180**
ISG2	1	1231	6	858	373	1656	0.97%	175	0.49%	427	0.49%	492	**0.00%**	**417**
	2	1070	5	715	355	1369	**0.00%**	**595**	0.00%	685	0.00%	1019	0.00%	600
	3	1074	5	715	359	1433	**0.00%**	**569**	0.00%	584	13.97%	621	0.00%	572
	4	1238	6	858	380	1555	**0.00%**	**111**	0.00%	416	0.00%	520	0.00%	335
	5	1079	5	715	364	1958	0.00%	560	0.00%	476	0.00%	490	**0.00%**	**467**
	6	1061	5	715	346	677	0.00%	575	**0.00%**	**571**	0.00%	609	0.00%	578
	7	1223	6	858	365	>3600	**0.00%**	**194**	0.25%	438	0.25%	534	0.00%	338
	8	1082	5	715	367	1809	0.00%	406	0.00%	424	13.12%	731	**0.00%**	**341**
	9	1226	6	858	368	>3600	0.08%	107	0.00%	411	0.00%	653	**0.00%**	**409**
	10	1215	6	858	357	2067	0.58%	170	0.82%	452	0.82%	485	**0.00%**	**430**

Notes: F_{CPLEX} and t_{CPLEX} denote the objective value and solution time of solving the original model TDP by CPLEX, respectively. In the above CG-based solution method, we use the DP to solve the PPs. The bold values in this table are the best solutions.

The results in Table 2 demonstrate that optimality gap obtained with Strategy 4 (i.e., values in the column Δ_{F4}) is always zero; and Strategy 4 performs the best in the majority of the instances. The reason why Strategy 4 performs the best among the four strategies may lie in the fact that the design of Strategy 4 is the most sophisticated and combines some features of the other strategies. For example, the set of fixed variables combines the variables ξ_i^p and z_p fixed in Strategy 2, and the variables y_{is} fixed in Strategy 3; the thresholds adopted in some judgement conditions include the threshold $1/|P|$ used in Strategy 2 and the threshold $1/|C_i|$ used in Strategy 3. In addition, the number of steps contained in this strategy is obviously more than for the other strategies.

Here the performance is evaluated according to the optimality gap $\Delta_{F\#}$ as the main criterion and to the solution time $t_{\#}$(s) as the secondary criterion. Although Strategy 4 does not perform the best for some instances, its optimality gap Δ_{F4} for these instances is still zero. In addition, for the remaining three strategies, Strategy 1 is better than Strategy 2, and Strategy 3 has the worst performance. Therefore we adopt Strategy 4 in the CG-based solution method for the remaining experiments.

6.2.2 Evaluating performance of solving PPs by DP and CPLEX

A DP was proposed in Section 5.3 to solve the PPs for improving the performance of the CG-based solution method. In the CG related literature, the PPs can also be solved by CPLEX. Hence some experiments were conducted to compare the objective value as well as solution time of the results obtained by CPLEX to solve PPs and using DP to solve PPs. The Δ_F and Δ_t columns in Table 3 denote relative gap with respect to CPLEX for all solution value and the solution time, respectively.

Table 3: Comparison of solving PPs by DP and CPLEX

Instance		**Solve PP_p by CPLEX**		**Solve PP_p by DP**		Δ_F	Δ_t
Scale	ID	F_{P_CPLEX}	t_{P_CPLEX}(s)	F_{P_DP}	t_{P_DP}(s)		
ISG1	1	671	173	671	151	0.00%	-12.72%
	2	658	141	658	132	0.00%	-6.38%
	3	651	185	651	148	0.00%	-20.00%
	4	542	85	542	85	0.00%	0.00%
	5	658	173	658	155	0.00%	-10.40%
	6	654	143	654	142	0.00%	-0.70%
	7	666	137	666	134	0.00%	-2.19%
	8	646	193	646	136	0.00%	-29.53%
	9	664	191	664	172	0.00%	-9.95%
	10	659	180	659	180	0.00%	0.00%
ISG2	1	1231	420	1231	417	0.00%	-0.71%
	2	1238	432	1238	420	0.00%	-2.78%
	3	1086	550	1086	538	0.00%	-2.18%
	4	1238	339	1238	335	0.00%	-1.18%
	5	1079	485	1079	467	0.00%	-3.71%
	6	1218	606	1218	403	0.00%	-33.50%
	7	1230	546	1230	485	0.00%	-11.17%
	8	1082	495	1082	341	0.00%	-31.11%
	9	1226	413	1226	409	0.00%	-0.97%

	10	1215	456	1215	430	0.00%	-5.70%
Average			317		284	0.00%	-9.24%

Notes: F_{P_CPLEX} and F_{P_DP} denote the objective value of the solution obtained by the CG-based method, in which the PPs are solved by using CPLEX and DP, respectively. t_{P_CPLEX} and t_{P_DP} denote the solution time of the CG-based method, in which the PPs are solved by using CPLEX and DP, respectively. $\Delta_F = (F_{P_{DP}} - F_{P_{CPLEX}})/F_{P\_CPLEX}$, $\Delta_t = (t_{P_{DP}} - t_{P_{CPLEX}})/t_{P\_CPLEX}$.

Because the DP is customized for the PPs in this study while CPLEX is a general solver, the better performance of the DP with respect to CPLEX for solving the PPs was to be expected, and is consisted with the results of related studies in the field of column generation. The results in Table 3 demonstrate that using DP to solve PPs does not worsen the solution quality, which is supported by the zero values in the column Δ_F. In addition, if we use DP to solve PPs, the solution time of the CG-based method can be shortened. As shown in the rightmost column of Table 4, the average of the Δ_t values is -9.24%, which implies that DP accelerates the solution process. According to the results of Table 3, DP takes less time to solve the PPs than CPLEX, and the difference is statistically significant (paired t-test, significance level $\alpha = 0.05$, p-value $\approx$ 0).

6.2.3 Evaluating performance of strategies for algorithmic acceleration

Some algorithmic acceleration strategies were also proposed in Section 5.5. We have conducted experiments to compare the objective value as well as the solution time of the results obtained by the CG-based method with the acceleration strategies and without them. The Δ_{F_D} and Δ_{t_D} in Table 4 denote relative gap between the two options with respect to the objective value and the solution time, respectively.

Table 4: Comparison of performance with using and without using strategies of acceleration (small scale)

Instance		**Without using the strategies**		**With using the strategies**		Δ_{F_D}	Δ_{t_D}
Scale	ID	F_{CG}	t_{CG}(s)	F_D	t_D(s)		
ISG1	1	671	151	671	107	0.00%	-29.14%
	2	658	132	658	105	0.00%	-20.45%
	3	651	148	651	137	0.00%	-7.43%
	4	542	85	542	90	0.00%	5.88%
	5	656	143	656	143	0.00%	0.00%
	6	654	142	654	124	0.00%	-12.68%
	7	666	134	666	105	0.00%	-21.64%
	8	658	147	658	115	0.00%	-21.77%
	9	664	172	664	180	0.00%	4.65%
	10	659	180	659	187	0.00%	3.89%
ISG2	1	1231	417	1231	420	0.00%	0.72%
	2	1070	600	1070	280	0.00%	-53.33%
	3	1074	572	1074	126	0.00%	-77.97%
	4	1238	335	1238	46	0.00%	-86.27%
	5	1079	467	1079	263	0.00%	-43.68%
	6	1061	578	1061	652	0.00%	12.80%
	7	1223	338	1223	281	0.00%	-16.86%
	8	1082	341	1082	103	0.00%	-69.79%

	9	1226	409	1226	441	0.00%	7.82%
	10	1215	430	1215	478	0.00%	11.16%
Average						0.00%	-20.70%

Notes: F_{CG} and F_D denote the objective value of the solution obtained by the CG-based method with using the acceleration strategies and without them, respectively. The columns t_{CG} and t_D denote the solution time of the CG-based method with using the acceleration strategies and without using the strategies, respectively. $\Delta_{F_D} = (F_D - F_{CG})/F_{CG}$, $\Delta_{t_D} = (t_D - t_{CG})/t_{CG}$.

The results in Table 4 demonstrate that using the strategies does not worsen the solution quality, which is supported by all the zero values in the column Δ_{F_D}; also the strategies can shorten the solution time of the CG-based method in a majority of the 20 instances. The average of the Δ_{t_D} values is 20.7%, which implies that the strategies proposed in Section 5.5 can sometimes accelerate the solution process significantly. The reason why our proposed strategies can accelerate the solution process mainly lies in the use of upper bounds to reduce the number of states that need be examined at each stage of the DP procedure; in addition, a criterion is also used for adding to the RMP the columns that have a large potential to improve the quality of final solution.

The results in Table 4 are based on the small-scale instances. To further confirm the above results, more experiments are conducted on the basis of the large-scale instances, i.e., ISG 3 and ISG 4. The results in Table 5 also validate the advantage brought by the proposed strategies. The results imply that the advantage of using the strategies becomes more significant on the large-scale instances. The average of the values in the column Δ_{t_D} is now 58.97%. Both Table 4 and Table 5 validate the effectiveness of the proposed acceleration strategies.

Table 5: Comparison of performance with using and without using the strategies of acceleration (large scale)

Instance		**Without using the strategies**		**With using the strategies**		Δ_{F_D}	Δ_{t_D}
Scale	ID	F_{CG}	t_{CG}(s)	F_D	t_D(s)		
ISG3	1	2641	814	2641	455	0.00%	-44.10%
	2	2646	540	2646	509	0.00%	-5.74%
	3	2632	606	2632	129	0.00%	-78.71%
	4	2640	545	2640	112	0.00%	-79.45%
	5	2632	902	2632	443	0.00%	-50.89%
	6	2620	604	2620	437	0.00%	-27.65%
	7	2648	571	2648	104	0.00%	-81.79%
	8	2638	558	2638	221	0.00%	-60.39%
	9	2358	673	2358	355	0.00%	-47.25%
	10	2347	715	2347	489	0.00%	-31.61%
ISG4	1	4898	1858	4898	177	0.00%	-90.47%
	2	4889	2112	4889	280	0.00%	-86.74%
	3	4881	2566	4881	161	0.00%	-93.73%
	4	4878	2395	4878	1304	0.00%	-45.55%
	5	4891	2743	4891	183	0.00%	-93.33%
	6	4895	3057	4895	171	0.00%	-94.41%
	7	4868	2954	4868	2532	0.00%	-14.29%
	8	4368	2384	4368	2015	0.00%	-15.48%
	9	4866	2632	4866	872	0.00%	-66.87%

	10	4892	2243	4892	653	0.00%	-70.89%
Average		-	-	-	-	0.00%	-58.97%

A hypothesis test was also performed for the above comparative result; and the difference in computation time between the methods with using and without using the strategies of acceleration is statistically significant (paired Wilcoxon signed-rank test, significance level $\alpha = 0.05$, p-value $\approx$ 0).

6.3 Evaluating performance of the tailored CG-based solution method

The experiments conducted in the previous subsection confirm the advantage of using DP to solve the PPs, as well as the contribution of some acceleration strategies. Hence our CG-based solution method incorporates these strategies. We then need to investigate the quality of solutions solved by the tailored CG-based solution method. On the small-scale instances, the results obtained by the tailored method can be compared with the optimal results obtained by CPLEX. On large-scale instances, CPLEX is probably inapplicable. Hence we need to find another metric to evaluate the performance of our method. Here we use CPLEX to solve the original model TDP while relaxing the binary variables ξ_i^p, x_r and y_{is}, and we obtain a lower bound (LB) for the TDP, which is then used on large-scale instances to measure the quality of solutions solved by the tailored CG-based solution method.

The values in column $\Delta_{F_{CG\_A}}$ of Table 6 are zero, which confirms that the CG-based solution method can obtain optimal solutions within much shorter solution time than CPLEX, by about 54.93% on average, which is reflected by the values in column $\Delta_{t_{CG\_A}}$. We note that the values in column $\Delta_{F_{LB}}$ reflect how close the LB is from optimality; the average relative gap of the LB from the optimality is about 12.26%. This value will be used in the comparative experiments of the large-scale instances. The reason for using $\Delta_{F_{LB}}$ in the small-scale results (Table 6) is due to the comparison in the large-scale results (Table 7), in which we obtain the gap between the LB and the results solved by our algorithm. Then we can evaluate the quality of our solved results by comparing the gap (in Table 7) with $\Delta_{F_{LB}}$ (Table 6). It is noted that this comparison is not strictly sound because the objects are different from each other (one is small-scale and the other is large-scale); however, this comparison could be regarded as an approximate reference for evaluating the performance of our proposed algorithm in the large-scale instances when CPLEX cannot solve them.

Table 6: Comparison of the CG-based solution method with CPLEX and LB (small-scale)

Instance		**CPLEX**		**LB**	**CG-based method**		$\Delta_{F_{LB}}$	$\Delta_{F_{CG\_A}}$	$\Delta_{t_{CG\_A}}$
Scale	ID	F_{CPLEX}	t_{CPLEX}(s)	F_{LB}	F_{CG_A}	t_{CG_A}(s)			
	1	671	253	562	671	107	19.40%	0.00%	-57.71%
	2	658	306	549	658	105	19.85%	0.00%	-65.69%
	3	651	340	541	651	137	20.33%	0.00%	-59.71%
	4	542	41	542	542	90	0.00%	0.00%	119.51%
ISG1	5	656	310	546	656	143	20.15%	0.00%	-53.87%
	6	654	608	544	654	124	20.22%	0.00%	-79.61%
	7	666	301	557	666	105	19.57%	0.00%	-65.12%
	8	658	344	549	658	115	19.85%	0.00%	-66.57%

	9	664	253	555	664	180	19.64%	0.00%	-28.85%
	10	659	257	550	659	187	19.82%	0.00%	-27.24%
ISG2	1	1231	1656	1087	1231	420	13.25%	0.00%	-74.64%
	2	1070	1369	1070	1070	280	0.00%	0.00%	-79.55%
	3	1074	1433	1074	1074	126	0.00%	0.00%	-91.21%
	4	1238	1555	1095	1238	46	13.06%	0.00%	-97.04%
	5	1079	1958	1078	1079	263	0.09%	0.00%	-86.57%
	6	1061	677	1060	1061	652	0.09%	0.00%	-3.69%
	7	1223	>3600	1080	1223	281	13.24%	0.00%	--
	8	1082	1809	1081	1082	103	0.09%	0.00%	-94.31%
	9	1226	>3600	1083	1226	441	13.20%	0.00%	--
	10	1215	2067	1072	1215	478	13.34%	0.00%	-76.87%
Average							12.26%	0.00%	-54.93%

Notes: $\Delta_{F_{LB}} = \frac{F_{CPLEX} - F_{LB}}{F_{LB}}$, $\Delta_{F_{CG\_A}} = \frac{F_{CG\_A} - F_{CPLEX}}{F_{CPLEX}}$, $\Delta_{t_D} = \frac{t_{CG\_A} - t_{CPLEX}}{t_{CPLEX}}$.

The results shown in Table 7 demonstrate that CPLEX cannot solve large-scale instances within one hour. The column F_{CPLEX} records the best objective values for the instances solved by CPLEX within one hour; the majority of these values are worse than those obtained by the CG-based method. The average value of the column $\Delta_{F_{CG\_A}}$ implies that the solution value obtained by the CG-based method is lower than that obtained by CPLEX by about 7.12%. Therefore, we use the aforementioned LB to evaluate the solution quality of the CG-based method. The average value of the column $\Delta_{F_{LB}}$ demonstrates that the solution values obtained by the CG-based method is higher than the LB by 13.35% on average, which is considerable. However, we recall that the average optimality gap of the LB is about 12.26%, validated on the small-scale instances. The comparison on these two gap values (13.35% and 12.26%) leads us to conclude that the performance of the CG-based method is satisfactory for solving large-scale instances of the territorial design problem.

Table 7: Comparison of the CG-based solution method with CPLEX and LB (large-scale)

Instance		**CPLEX**		**LB**	**CG-based method**		$\Delta_{F_{LB}}$	$\Delta_{F_{CG\_A}}$
Scale	ID	F_{CPLEX}	t_{CPLEX}(s)	F_{LB}	F_{CG_A}	t_{CG_A}(s)		
ISG3	1	2912	>3600	2371	2641	455	11.39%	-9.31%
	2	2906	>3600	2366	2646	509	11.83%	-8.95%
	3	2896	>3600	2352	2632	129	11.90%	-9.12%
	4	2895	>3600	2354	2640	112	12.15%	-8.81%
	5	2632	>3600	2362	2632	443	11.43%	0.00%
	6	2884	>3600	2343	2620	437	11.82%	-9.15%
	7	2911	>3600	2368	2648	104	11.82%	-9.03%
	8	2908	>3600	2367	2638	221	11.45%	-9.28%
	9	2358	>3600	2358	2358	355	0.00%	0.00%
	10	2347	>3600	2346	2347	489	0.04%	0.00%
ISG4	1	5381	>3600	4138	4898	177	18.37%	-8.98%
	2	5378	>3600	4126	4889	280	18.49%	-9.09%
	3	5111	>3600	4114	4881	161	18.64%	-4.50%
	4	--	>3600	4108	4878	1304	18.74%	--

5	--	>3600	4131	4891	183	18.40%	--
6	5381	>3600	4134	4895	171	18.41%	-9.03%
7	5106	>3600	4096	4868	2532	18.85%	-4.66%
8	5119	>3600	4119	4368	2015	6.05%	-14.67%
9	5103	>3600	4099	4866	872	18.71%	-4.64%
10	5372	>3600	4126	4892	653	18.57%	-8.94%
Average						13.35%	-7.12%

Notes: $\Delta_{F_{LB}} = \frac{F_{CG\_A} - F_{LB}}{F_{LB}}$, $\Delta_{F_{CG\_A}} = \frac{F_{CG\_A} - F_{CPLEX}}{F_{CPLEX}}$.

As aforementioned in the literature review, a paper by Zhou et al. (2021) is the most related to this study. We therefore implemented the genetic algorithm proposed by Zhou et al. (2021) and compared it with the column generation based method proposed in this paper. The details on the genetic algorithm implemented in the following comparative experiments are provided in Appendix G. For the comparative experiments, we generated a group of instances (i.e., ISG5) whose scale is larger than the previous groups; in each instance in ISG5, the number of districts P is 25, and the number of customers N is 100.

Table 8: Comparison of the CG-based solution method and genetic algorithm (larger-scale)

Instance		**CG-based method**		**Genetic algorithm**		$\Delta_{F_{CG\_A}}$	$\Delta_{t_{CG\_A}}$
Scale	ID	F_{CG_A}	t_{CG_A}(s)	F_{GA}	t_{GA}(s)		
ISG5	1	9430	407	9431	2317	0.01%	469.29%
	2	9400	402	9403	2561	0.03%	537.06%
	3	9391	581	9394	3298	0.03%	467.64%
	4	9392	661	9395	3256	0.03%	392.59%
	5	9437	508	9442	2613	0.05%	414.37%
	6	9382	405	9385	3232	0.03%	698.02%
	7	9382	509	9417	2887	0.37%	467.19%
	8	9416	639	9421	2466	0.05%	285.92%
	9	9402	459	9405	2723	0.03%	493.25%
	10	9325	322	9325	2717	0.00%	743.79%
Average						0.06%	496.91%

Notes: $\Delta_{F_{CG\_A}} = \frac{F_{GA} - F_{CG\_A}}{F_{GA}}$, $\Delta_{t_{CG\_A}} = \frac{t_{GA} - t_{CG\_A}}{t_{GA}}$.

The average value in column $\Delta_{F_{CG\_A}}$ of Table 8 is 0.06%, which indicates the results obtained by the two methods are very close to each other while the result solved by our CG-based method is a bit better than that of the genetic based algorithm. Moreover, another merit of our method is that the CG-based solution method can solve the problem much faster than the genetic based algorithm; this is reflected by the average of the $\Delta_{t_{CG\_A}}$ values, i.e., 496.91%, which implies the CG-based method can save the solution time significantly compared with the genetic algorithm. These experimental results validate the advantages of our proposed CG-based method with respect to both solution quality and computing time.

6.4 Deriving managerial insights from sensitivity analyses

Based on the proposed territorial design model and on the tailored solution method, sensitivity analyses were conducted for the purpose of deriving some managerial implications. Before using the proposed methodology, a decision maker needs to formulate a series of delivery route templates in advance. Thus, the number of route templates, the district cost, and the customer distribution feature should affect the final result. This subsection studies the influence of these factors.

6.4.1 Influence of route number on final performance

The number of route templates (i.e., $\left|\bigcup_{\forall p,t} R^{pt}\right|$) may have an impact on the final performance with respect to the objective value and the solution time. Figure 6 shows that generating more route templates may reduce the final cost, but not significantly; in some cases the final cost stays unchanged when the number of route templates increases. These results are consistent with those of Florio et al. (2021), who also found that decreasing the number of routes does not generally considerably impact the travel cost. In addition, the last chart in Figure 6 shows that the number of districts in the solution is not significantly influenced by the number of route templates. However, the solution time increases significantly with the number of route templates. Decision makers in delivery service companies should find a balance between the quality of a territorial plan and the amount of time consumed for obtaining it.

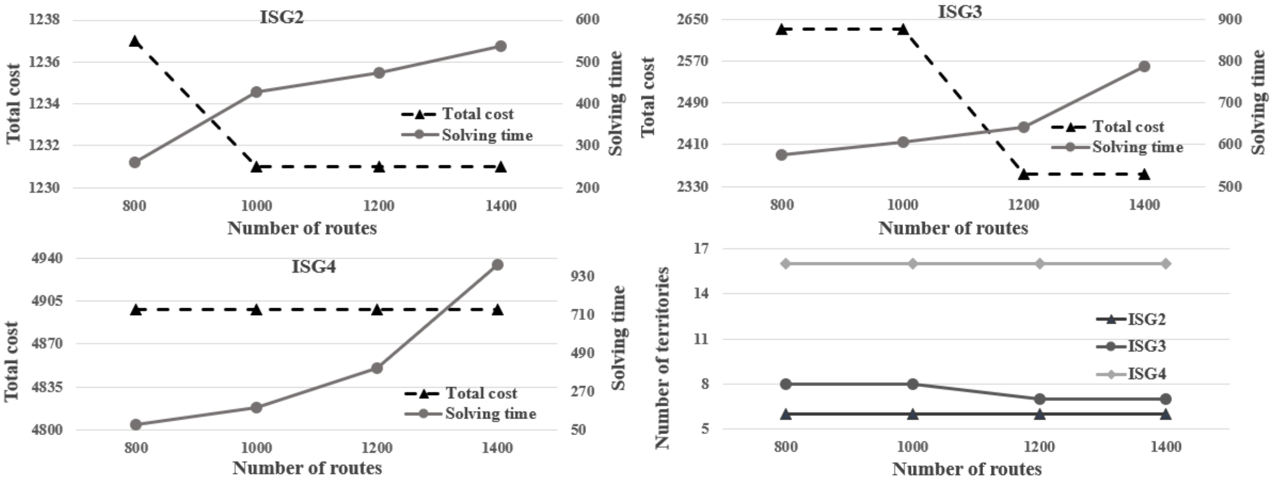


Figure 6: Influence of route number on final performance

6.4.2 Influence of district heterogeneity on final performance

In the model TDP, F_p is an important parameter. It denotes the fixed cost of setting district p during the planning horizon. If the districts are homogenous, this parameter's influence on the objective value may be limited. However, in a context with heterogeneous districts, the influence of this parameter should be investigated. Here we use the gap between maximum and minimum values of F_p, i.e., $\text{Max}_{\forall p} - \text{Min}_{\forall p} F_p$, to reflect the heterogeneous degree of districts. It should be noted the average value all F_p in each group of cases (i.e., $\text{Avg}_{\forall p} F_p$) is kept identical for a fair comparison. Figure 7 shows that given the identical average fixed cost for all cases, the higher degree for the district heterogeneity, the lower is the final cost. The reason may lie in the fact that a significant difference among districts' fixed costs yields a large probability of obtaining a better solution, so that the cost of the territorial plan could be reduced. In addition, the rightmost chart of Figure 7 illustrates that the district heterogeneity has no impact on the number of districts. Decision makers in delivery service companies should pay attention to the territory design when facing a context with significant heterogeneity with respect to the fixed cost of the districts.

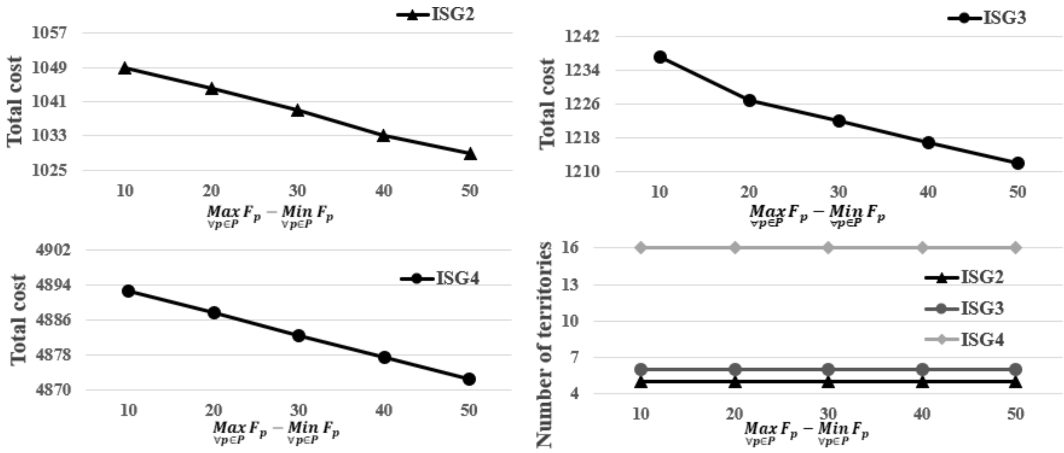


Figure 7: Influence of district heterogeneity on final performance

6.4.3 Influence of courier heterogeneity on final performance

One of the key features of our problem is the service frequency f_i of the customers, which means that customer i needs to be visited on f_i days in the planning horizon. The setting of this parameter may influence solution quality. In this sensitivity analysis, we calculate the average frequency of all customers in ISG 2 and ISG 3, and then generate a series of cases by increasing the average frequency by increments of 0.1, 0.2, 0.3, 0.4. The results shown in Figure 8 demonstrate that the higher is the average frequency, the higher is the total cost, which is consistent with the results of Zhou et al. (2021). However, the computation time decreases as the average frequency grows, the reason possibly being that when customers need to be served more frequently, it may be easier to generate good routes, which may speed up the solution process. The rightmost graph in Figure 8 shows that the changing of the average frequency has no influence on the number of districts that need to be created.

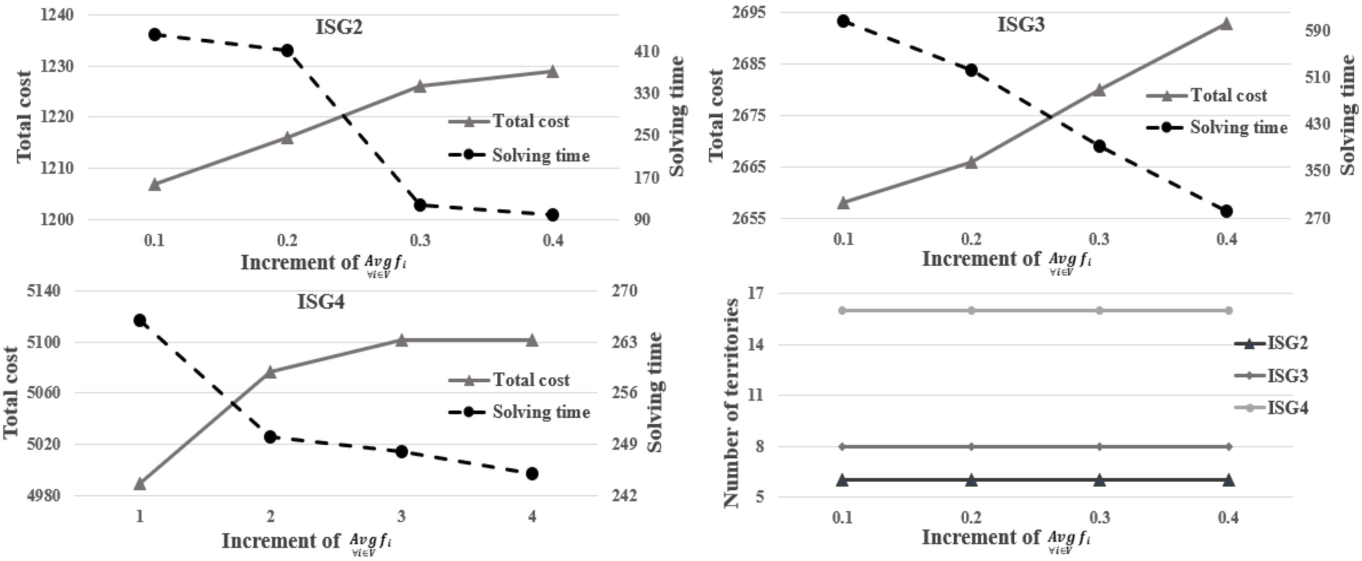


Figure 8: Influence of customer service frequency on final performance

6.4.4 Influence of customer distribution feature on final performance

Customer spatial distribution may also affect the final performance. As shown at the bottom of Figure 9, we generated two series of instances, which are different from the four instance groups listed in Table 1. These instances contain 16 and 40 customers located within a circular area with differences distributions, and the depot is at the center of the circle. From the left to right along the horizontal axis in each series of instances, the customer distribution evolves gradually from a diagonal to a uniform distribution. To ensure a fair comparison among the instances with different distributions, the average distance between customers and the depot is the same for all the instances in the same series; the average customers demand is also identical in the same series.

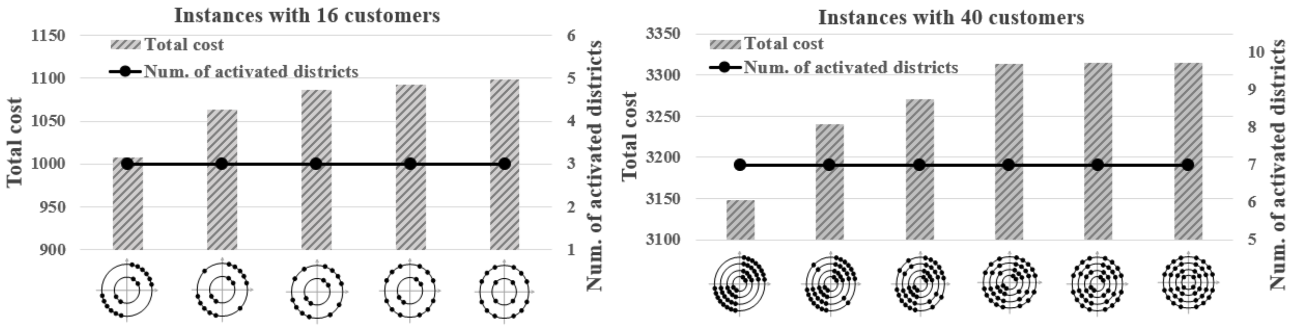
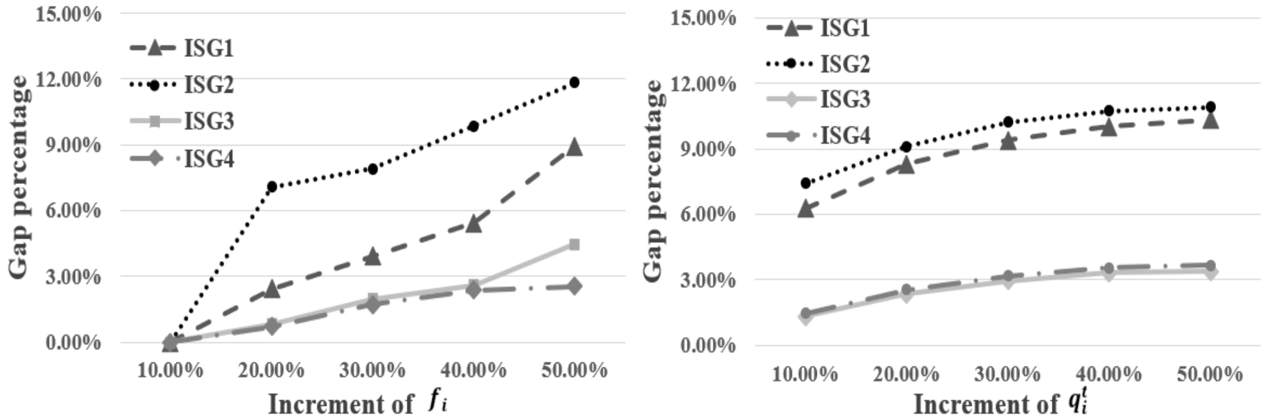


Figure 9: Influence of customer spatial distribution on cost and number of activated districts

The results (the bars) in Figure 9 demonstrate, interestingly, that the more evenly customers are dispersed, the higher is the total cost. This result is consistent with the results of Zhen et al. (2022) and of Florio et al. (2021), who found that when the demands are overdispersed, the average final cost is 3.4% higher than when they are underdispersed, the reason being that higher customer densities result in smaller transportation costs. In addition, the results (the lines) in Figure 9 indicate that the customer spatial distribution does not seem to influence the number of the activated districts.

6.5 Testing the robustness of the model

Our model is based on some estimated parameters such as f_i the number of days on which a customer has a demand during the planning horizon, q_i^t the customer demands, s_i^t the service time for a customer, c_{ij}^p the travel cost. A last stream of experiments was conducted to test the robustness of the model by analyzing the influence of these parameters' deviation on the objective's deviation. More specifically, we set the parameter value (f_i, q_i^t, s_i^t, c_{ij}^p) that deviates from a baseline value by a certain degree (e.g., 10%, 20%, 30%, 40%, 50%), solved the problem with the deviated value of the parameter, calculated the solution's objective value in the problem with the baseline value of the parameter, then compared the calculated objective value with the optimal objective value of the problem with the baseline value. The gap percentage of the comparative result is used to evaluate the robustness of the model.



(*a*) Robustness with respect to f_i (*b*) Robustness with respect to q_i^t

Figure 10: Robustness of the model with respect to parameters

The results in Figure 10 demonstrate that when the estimated value of the parameter f_i deviates from its baseline value by 50%, the obtained objective value deviates from the optimal value by at most about 12%. For the parameters q_i^t, the longest deviation percentage of the objective is about 11% when the parameters deviate from their baseline values by 50%. In the experiments, for the parameters s_i^t and c_{ij}^p, the deviation of the objective is always zero, when the parameters deviates from their baseline values by 10%, 20%, … , 50%; the reason for this phenomenon may be that the s_i^t (service time) and c_{ij}^p (travel cost) mainly influence route templates' costs, but the cost deviation does not seem to influence the selection decision of the route template. These results validate the robustness of the model with respect to the parameters f_i, q_i^t, s_i^t, and c_{ij}^p.

7 Conclusions

This study has investigated a territorial design problem for a periodical urban delivery service. A mathematical model as well as a tailored CG-based solution method were proposed. Extensive numerical experiments were conducted to validate the effectiveness of the model and the efficiency of the algorithm. The main contributions of this study can be summarized from the following three perspectives.

From the perspective of mathematical modelling, this study may be the first to formulate a mathematical model for a territorial design problem for customers with delivery demand frequency. A set partitioning based mathematical model was formulated on the basis of route templates and day-combinations periodicity. Our model can make simultaneous decisions on territory partitioning and route planning.

From an algorithmic perspective, a column algorithm based on column generation was designed to solve the proposed model. The algorithm decomposes the original problem into an RMP and a series of PPs, each of which applies to a single district. A dynamic programming based method was designed for the solution of the PPs. In addition, several tailored strategies were proposed and applied in the components of the algorithm so as to further accelerate the solution process for the PPs and the RMP. Numerical experiments were conducted to validate the contributions of the dynamic programming and of the strategies.

From the perspective of managerial insights, we have provided several potentially useful suggestions for practitioners. For example, preparing more route templates is beneficial to delivery managers to make a better territorial design plan. The managers should pay more attention to territorial design in contexts with significant heterogeneity with respect to districts fixed costs. In such a context, model based quantitative decisions may provide benefits. Another finding is that when customers are concentrated in some areas, it is beneficial for a delivery service provider to operate its delivery activities as well as plan the districts.

This study also contains limitations. For example, more advanced algorithms such as the exact solution methods could be further developed for this problem. In addition, this study can be extended to a more generic context. According to the historical data, we estimate the probability that each customer has demand at each day of the planning horizon (e.g., a customer's probability that it has demands every Monday). We could propose a stochastic programming model on territory design with the aim of minimizing the expected cost over the planning horizon.

Appendices

Appendix A: Initialization of the sets R^{pt}

First, it should be noted that the probability that each customer i belongs to one of districts is equal among all districts P; therefore, the set R^{pt} is identical for all $p \in P$ and $t \in T$ if the capacity limitation Q_p is identical for all the districts and the total travel time L_p is identical for all the districts. Thus the algorithm elaborated in this Appendix Eor routes' generation apply to a symmetric case. R^{pt} is actually the set of all the possible routes. The generation of the set R^{pt} is related to the parameters q_i^t, Q_p, and L_p. Here q_i^t is the delivery amount to customer i if visited on day t; Q_p is the limit of the total delivery amount of the customers in district p on one day; L_p is the maximum total travel time of the route used for district p on any day. The process of generating the set R^{pt} is described by the following pseudocode.

Pseudocode for generating the set R^{pt}

Input: the vehicle capacity of district p Q_p, the quantity of customer i on day t q_i^t, the number of customers $|V|$, the number of districts$|P|$, the travel cost for district p c_{ij}^p, the duration for district p d_{ij}^p, service time s_i^t for serving customer i on day t, a time window $[e_i^t,\ l_i^t]$ for visiting customer i on day t, the total working time of the route is L_p, the maximum number of routes L, $\forall p \in P, i \in V$

Output: R^{pt}, route r's cost c_r,$r \in R^{pt}$

For $t = 1, \dots, |T|$ **do** // for each day
  For $p = 1, \dots, |P|$ **do** // for each district, construct routes
    $AS \leftarrow 0$ // AS is index of route
    $CC \leftarrow 0, TT \leftarrow 0, CR \leftarrow 0$ // initialize CC as total currently used capacity for the customers that have been included in the route, TT is the total currently accumulated working time in the route, CR is the cost of the route
    For $i = 1, \dots, |V|$ **do** // for each customer as the first one in the route; each route is constructed by adding new customers with respect to the first customer i in the route
      $CC \leftarrow q_i^t,\ TT \leftarrow 2d_{0i}^p + s_i^t,\ CR \leftarrow 2c_{0i}^p$//0 denotes depot. When customer i is served, the route is $0 \dashrightarrow i \dashrightarrow 0$, so the time is $2d_{0i}^p + s_i^t$
      For $j = i+1, \dots, |V|$ **do** // for each customer that will be added into the route; the following loop is used to inserting new customers into the route
        If $CC + q_j^t \le Q_p$ && time window is met && $TT - d_{0i}^p + d_{ij}^p + d_{0j}^p + s_j^t \le L_p$
          Add customer j into the route
          $CC \leftarrow CC + q_j^t,\ TT \leftarrow TT - d_{0i}^p + d_{ij}^p + d_{0j}^p + s_j^t,\ CR \leftarrow CR + d_{ij}^p c_{ij}^p - d_{0i}^p c_{0i}^p + d_{0j}^p c_{ij}^p$

// update route AS's used capacity, accumulated working time, and route cost
End if
End for
$c_{AS} \leftarrow CR$ // Route AS's cost
$AS \leftarrow AS + 1$ // for generating next route in the next For loop
End for
Output set R^{pt} and route r's cost c_r, $r \in R^{pt}$
End for
End for

The construction of routes is based on the insertion of new customers (i.e., j) with respect to the first customer (i.e., i) in the route. We could also evaluate the insertion of customers from the last inserted customer; the performance is the same as the insertion of customers from the first one, which is adopted in the above pseudocode.

Appendix B: Initialization of the sets K_p

To start the column generation procedure, the RMP needs a set K_p of assignment plans as the initial input. The pseudocode for generating set K_p on the basis of the above generated set R^{pt} is as follows.

Pseudocode for generating the set K_p

Input: R^{pt}, F_p, Q_p, q_i^t, $|V|$, $|P|$, c_r, C_i, f_i, a_{ts}, $\forall i \in V, p \in P, t \in T$
Output: K_p, $Cost$ // $Cost$ is the cost of each assignment plan contained in the set K_p
$y_{is}, \xi_i^p, z_p, x_r \leftarrow 0$
Define elements in set R^{pt} as $B_1, B_2, \dots, B_{|R^{pt}|}$
$RR_{pt} \leftarrow \emptyset$ // set of customers served in district p at day t
For $t = 1, \dots, |T|$ **do**
  If $t = 1$
    For $p = 1, \dots, |P|$ **do**
      $CC \leftarrow 0, TT \leftarrow 0, CR \leftarrow 0$ // initialization for CC as the total used capacity by the customers that have been included in route, TT as the accumulated working time of the route, CR as the cost of the route
      For $i = 1, \dots, |V|$ **do**
        $A_i = 0$ // denote whether customer i is served, zero means not
        For $s = 1, \dots, |C_i|$ **do**
          If $a_{ts} = 1$
            $CC \leftarrow CC + q_j^t$, $TT \leftarrow TT - d_{0i}^p + d_{ij}^p + d_{0j}^p + s_j^t$, $CR \leftarrow CR + d_{ij}^p c_{ij}^p - d_{0i}^p c_{0i}^p + d_{0j}^p c_{ij}^p$
            If $CC \leq Q_p$ && time window is met && $TT \leq L_p$ && $A_i = 0$
              Add customer i into set RR_{pt}
              $A_i, y_{is}, z_p, \xi_i^p \leftarrow 1$
            End if
          End if
        End for
      End for
      For $r = 1, \dots, |R^{pt}|$ **do**
        If $B_r == RR_{pt}$ // if B_r is the same as RR_{pt}
          $x_r \leftarrow 1$
        End if
      End for

End for
Else
For $p = 1, \ldots, |P|$ **do**
$CC, TT, CR \leftarrow 0$
For $i = 1, \ldots, |V|$ **do**
For $s = 1, \ldots, |C_i|$ **do**
If $y_{is} = 1$ and $a_{ts} = 1$
$CC \leftarrow CC + q_j^t,\ TT \leftarrow TT - d_{0i}^p + d_{ij}^p + d_{0j}^p + s_j^t,\ CR \leftarrow CR + d_{ij}^p c_{ij}^p - d_{0i}^p c_{0i}^p + d_{0j}^p c_{ij}^p$
If $CC \leq Q_p$ && time window is met && $TT \leq L_p$
Add customer i into set RR_{pt}
End if
End if
End for
End for
For $r = 1, \ldots, |R^{pt}|$ **do**
If $B_r == RR_{pt}$
$x_r \leftarrow 1$
End if
End for
End for
End if
End for
For $r = 1, \ldots, |R^{pt}|$ **do**
$Cost \leftarrow \sum_{p\in P}\sum_{t\in T}\sum_{r\in R^{pt}} c_r x_r + \sum_{p\in P} F_p z_p$
End for
Return set K_p and the cost of each assignment plan contained in the set K_p

Appendix C: Proof of Proposition 1

Proposition 1: When $\sum_{i\in V}\theta_i < \min_{t\in T, r\in R^{pt}}\{c_r\} + F_p$, the PP_p model does not need to be solved.

Proof: In the pricing problem PP_p, z is a binary variable and denotes whether or not the district p is selected. If z equals zero, the binary variable ξ_i equals zero because of the inequality $\sum_{i\in V}\xi_i \leq |V|z$; then we have: (*i*) variable x_r equals zero because of the equation $\sum_{t\in T}\sum_{r\in R_i^{pt}} x_r = f_i\xi_i,\ \forall i \in V$; (*ii*) the variable y_{is} equals zero because of the inequality $y_{is} \leq \xi_i$. In this case, the objective of the PP_p model is $-\pi_p$, which is also the maximum objective value for the solution of the pricing problem (PP_p).

When z equals one, the objective of the PP_p model is $\sum_{t\in T}\sum_{r\in R^{pt}} c_r x_r + F_p - (\pi_p + \sum_{i\in V}\theta_i\xi_i)$, which should be less than $-\pi_p$, otherwise the binary variable z surely equals zero because the pricing problem (PP_p) is a minimization problem. In other words, the condition $\sum_{i\in V}\theta_i\xi_i > \sum_{t\in T}\sum_{r\in R^{pt}} c_r x_r + F_p$ must hold, otherwise the PP_p model needs not be solved and z equals zero directly.

Because of $\sum_{i\in V}\theta_i\xi_i < \sum_{i\in V}\theta_i$, the above condition turns to: $\sum_{i\in V}\theta_i < \sum_{t\in T}\sum_{r\in R^{pt}} c_r x_r + F_p$. In addition, due to $\sum_{t\in T}\sum_{r\in R^{pt}} c_r x_r > \min_{t\in T, r\in R^{pt}}\{c_r\}$, the above condition turns to $\sum_{i\in V}\theta_i < \min_{t\in T, r\in R^{pt}}\{c_r\} + F_p$.

In all, when $\sum_{i\in V}\theta_i < \min_{t\in T, r\in R^{pt}}\{c_r\} + F_p$, the PP_p model needs not be solved. ■

Appendix D: Pseudocode of the dynamic programing for solving the pricing problem

For describing the detailed process of the dynamic programming of solving the pricing problem PP_p, the pseudocode of the procedure is elaborated as follows.

Dynamic programing for solving PP_p

Input: Set of customers V, service frequency f_i, set of day-combinations C_i, set of route templates R^{pt}

Output: The column (assignment plan for district p) with the lowest reduced cost (denoted by re)

$f_1(Z_{10}) \leftarrow 0$
For $s = 1, \ldots, |C_{i(1)}|$ **do**
    $f_1(Z_1) \leftarrow c'_{1s} + F_p - (\pi_p + \theta_{i(1)}\xi_{i(1)})$
End for
For $k = 2,3, \ldots, |V|$ **do**
    For $s = 0,1,2, \ldots, |C_{i(k)}|$ **do**
        $f_k(Z_{ks}) \leftarrow \infty$
    End for
End for
For $k = 2, \ldots, |V|$ **do**
    For $s = 0,1,2, \ldots, |C_{i(k)}|$ **do**
        For $s_1 = 0,1, \ldots, |C_{i(k-1)}|$ **do** // day-combinations for the $(k-1)^{th}$ stage
            Determine routes according to the set of route templates, the chosen day-combinations in the previous stages; calculate the cost of all the selected routes.
            $v_k\left(u_{k-1}(Z_{k-1,s_1}), u_k(Z_{ks})\right) \leftarrow \tilde{c}_{kss_1} + F_p - (\pi_p + \theta_{i(k)}\xi_{i(k)})$
        End for
        $r\_cost \leftarrow 0$ // r_cost is defined as the objective value in the k^{th} stage
        $r\_cost \leftarrow \min_{\substack{x=0,1,\ldots,|C_{i(k-1)}| \\ Z_{k-1,x} \in S_{k-1}}} \left\{ v_k(u_{k-1}(Z_{k-1,x}), u_k(Z_{ks})) - \min_{s'=0,1,\ldots,|C_{i(k-2)}|} \{\tilde{c}_{(k-1),s,s'}\} + \pi_p - F_p + f_{k-1}(Z_{k-1,x}) \right\}$
        If $r\_cost < f_k(Z_{ks})$
            $f_k(Z_{ks}) \leftarrow r\_cost$
        End if
    End for
End for
$re \leftarrow \infty$ // re is defined as the lowest reduced cost
For $k = 1,2, \ldots, |V|$ **do**
    For $s = 0,1,2, \ldots, |C_{i(k)}|$ **do**
        If $re > f_k(Z_{ks})$
            $re \leftarrow f_k(Z_{ks})$ and record updated the chosen day-combinations in stages
        End if
    End for
End for
Output the chosen day-combinations in stages, the chosen routes in each day, re, i.e., the value of the lowest reduced cost

Appendix E: The day-combination types of customers

According to Zhou et al. (2021), the day-combination types of customers are shown in Table F.

Table F: The day-combination types of customers

Freq./Day	Mon.	Tue.	Wed.	Thu.	Fri.	Sat.
1	1	0	0	0	0	0
	0	1	0	0	0	0
	0	0	1	0	0	0
	0	0	0	1	0	0
	0	0	0	0	1	0
	0	0	0	0	0	1
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
6	1	1	1	1	1	1

Appendix F: The notations of indices, sets, parameters and variables in this paper

Table G: The list of notations for indices, sets, parameters and variables

Notations	Types	Meaning
T	set	set of days of the planning horizon
t	index	a day
V	set	set of customers
V'	set	$V' = \{0\} \cup V$
V_t	set	subset of customers that can be visited on day $t \in T, V_t \in V$
i, j	index	a customer
P	set	set of districts
P_i	set	a subset $P_i \subseteq P$ of allowable vehicles for visiting customer i
p	index	a district
A	set	set of arcs
C_i	set	set of allowable day-combinations of f_i visit days
s	index	a day-combination
R^{pt}	set	set of all route templates of day $t \in T$ for district $p \in P$
R_i^{pt}	set	index set of the routes of day $t \in T$ for district $p \in P$ covering customer $i \in V$
r	index	a route
K_p	set	set of all the feasible assignment plans for district p
κ_p	index	a feasible assignment plan for district p
Q_p	parameter	delivery capacity of the courier/vehicle in district p
L_p	parameter	maximum working time of the courier/vehicle in district p
F_p	parameter	fixed cost of activating district p

c_{ij}^{p}	parameter	travel cost of arc $(i,j) \in A$ for district p
d_{ij}^{p}	parameter	duration of arc $(i,j) \in A$ for district p
f_i	parameter	number of days for customer i needs be visited in the planning horizon
q_i^t	parameter	commodity quantity of customer i that must receive if visited on day t
s_i^t	parameter	service time for serving customer i on day t
$[e_i^t,\ l_i^t]$	parameter	time window for visiting customer i on day t
$[a_{ts}]$	parameter matrix	binary matrix, equal to one if and only if day t is an allowable visit day in day-combination s
c_r	parameter	cost of route $r \in R^{pt}$
c_{κ_p}	parameter	cost of the assignment plan κ_p
$\xi_i^{\kappa_p}$	parameter	binary parameter, equal to one if and only if customer i is contained in the assignment plan κ_p
π_p	parameter	dual variables for Constraints (14), $p \in P$
θ_i	parameter	dual variables for Constraints (15), $i \in V$
ξ_i^p	variable	binary variable, equal one if and only if customer $i \in V$ is assigned to district $p \in P$
x_r	variable	binary variable, equal one if and only if route $r \in R^{pt}$ on day $t \in T$ for district $p \in P$ is selected in the solution
y_{is}	variable	binary variable, equal one if and only if day-combination $s \in C_i$ is assigned to customer $i \in V$
z_p	variable	binary variable, equal one if and only if district $p \in P$ is used
y_{isp}	variable	binary variable, equal one if and only if day-combination $s \in C_i$ is assigned to customer i in district p
η_{κ_p}	variable	binary variable, equal one if and only if assignment plan κ_p is used by district p

Appendix G: Details on genetic algorithm implemented in Table 8's comparative experiments

(1) The main differences between our implemented genetic algorithm (GA) and the GA used in Zhou, Zhen, Baldacci, et al. (2021) are as follows:

- In the stage of initializing population, our GA is based the process of generating initial solutions for the column generation (elaborated in Appendix B), while the GA of Zhou et al. is based on a two-stage greedy heuristic.
- For the fitness function, our GA adopts the formulated model's objective; while the GA of Zhou et al. uses a customized objective and there exists a possibility that the initial solutions are infeasible for their problem.
- The chromosome design in our GA is simpler than the design in the GA of Zhou et al. because the main decision in the set partitioning based problem context is the selection of route templates while the Zhou et al.'s GA need make districting and routing decisions in their chromosomes.

(2) Tuning the algorithmic parameters of our GA

According to the usual practice of the GA, the rates of the crossover and mutation are two of the important algorithmic parameters. Thus we conduct a series of experiments under different combinations of crossover rate and mutation rate. The set of possible mutation rates is: {0.01, 0.02, 0.03, 0.04, 005}; and the set of possible crossover rates is: {0.5, 0.6, 0.7, 0.8, 0.9}. The experiments are conducted on the basis of ten instances, the average value of the solved objectives and the solution time among the ten instance are recorded in each entry of Table G-1 and Table G-2, respectively. According to Table G-1, the setting of crossover rate as 0.8 and mutation rate as 0.03 is the best one among the 25 combinations. According to Table G-2, the solution time does not vary significantly among the 25 combinations. Therefore we choose the setting of crossover rate as 0.8 and mutation rate as 0.03 for the comparative experiments in Table 8.

Table G-1: Objective values under different combinations of crossover rate and mutation rate

crossover rate \ mutation rate	0.01	0.02	0.03	0.04	0.05
0.5	2267	2265	2267	2266	2267
0.6	2266	2265	2272	2264	2270
0.7	2267	2272	2272	2268	2265
0.8	2269	2274	**2258**	2270	2262
0.9	2272	2274	2267	2267	2269

Table G-2: Solution time (s) under different combinations of crossover rate and mutation rate

crossover rate \ mutation rate	0.01	0.02	0.03	0.04	0.05
0.5	40	38	37	38	38
0.6	37	37	39	36	38
0.7	40	38	38	38	39
0.8	38	39	40	40	38
0.9	40	39	39	39	38

The size of the population, i.e., the number of chromosomes, may also have the influence on the performance of the algorithm. Four different sizes of population are tested with setting the size as 50, 100, 150, and 200. We further conduct experiments on the basis of five instances, the solved objective values and the solution time are recorded in Table G-3. We adopt the solved objective value as the primary criterion and the solution time as the secondary criterion in the algorithmic parameter selection; the setting of population size as 100 is chosen for our GA. Therefore, the GA with setting population size as 100, crossover rate as 0.8 and mutation rate as 0.03 is implemented in the comparative experiments shown in Table 8.

Table G-3: Objective values and solution time under different sizes of population

Instance	Population size = 50		Population size = 100		Population size = 150		Population size = 200	
ID	OBJ_1	t_1(s)	OBJ_2	t_2(s)	OBJ_3	t_3(s)	OBJ_4	t_4(s)
1	2274	17	**2266**	**34**	2266	55	2266	75
2	2259	17	**2251**	**35**	2251	55	2251	93
3	2242	18	**2240**	**36**	2240	54	2240	99
4	2884	45	**2810**	**91**	2810	147	2810	257
5	2851	45	**2776**	**91**	2776	150	2776	253
Average	2502	28	2469	57	2469	92	2469	155

Acknowledgement

The authors thank the editor (Professor J.F. Oliveira) and three anonymous reviewers for their constructive suggestions on improving this paper. This research was supported by the National Natural Science Foundation of China (Grant numbers 72025103 and 71831008). Thanks are due to the referees for their valuable comments.

References

Alinaghian M., Ghazanfari M., Hamedani S.G. (2018) A new bi-objective periodic vehicle routing problem with maximization market share in an uncertain competitive environment. *Computational & Applied Mathematics* 37(2): 1680–1702.

Bard J.F., Jarrah A. (2009) Large-scale constrained clustering for rationalizing pickup and delivery operations. *Transportation Research Part B: Methodological* 43(5): 542–561.

Beltrami E. J., Bodin, L. D. (1974) Networks and vehicle routing for municipal waste collection. *Networks* 4(1): 65–94.

Bender M., Kalcsics, J. (2020) Multi-period service territory design. In *Optimal Districting and Territory Design*, Volume 284 of International Series in Operations Research and Management Science, Chapter 7: 129–152. Springer, Cham, Switzerland.

Bender M., Kalcsics J., Meyer A. (2020) Districting for parcel delivery services – a two stage solution approach and a real-world case study. *Omega*, 96:102283.

Bender M., Kalcsics J., Nickel S., Pouls M. (2018) A branch-and-price algorithm for the scheduling of customer visits in the context of multi-period service territory design. *European Journal of Operational Research* 269(1): 382–396.

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

Carlsson J. G. (2012) Dividing a territory among several vehicles. *INFORMS J Computing* 24(4): 565–577.

Carlsson J. G, Delage E. (2013) Robust partitioning for stochastic multivehicle routing. *Operations Research* 61(3): 727–744.

Cordeau J.-F., Gendreau, M., Laporte G. (1997) A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks* 30, 105–119.

Cortés C. E., Gendreau M., Rousseau L.-M., Souyris S., Weintraub A. (2014) Branch-and-price and constraint programming for solving a real-life technician dispatching problem. *European Journal of Operational Research* 238(1): 300–312.

Daganzo C. F. (1984) The distance traveled to visit n points with a maximum of C stops per vehicle: an analytic model and an application. *Transportation Science* 18(4): 331–350.

Darmian S.M., Fattahi M., Keyvanshokooh E. (2021) An optimization-based approach for the healthcare districting under uncertainty. *Computers & Operations Research* 135: 105425.

Dawande M., Johar M., Kumar S., Mookerjee V. S. (2008) A comparison of pair versus solo programming under different objectives: An analytical approach. *Information Systems Research* 19(1): 71–92.

Elloumi S., Labbé M., Pochet Y. (2004) A new formulation and resolution method for the *p*-center problem. *INFORMS Journal on Computing* 16(1): 84–94.

Florio A. M., Hartl R. F., Minner S., Salazar-González J. J. (2021) A Branch-and-price algorithm for the vehicle routing problem with stochastic demands and probabilistic duration constraints. *Transportation Science* 55(1):122–138.

Francis P.M., Smilowitz K.R., Tzur M. (2006) The period vehicle routing problem with service choice. *Transportation Science* 40(4): 439–454.

Garey M. R., Johnson D. S. (1979) *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, New York.

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

Haugland D., Ho S. C., Laporte G. (2007) Designing delivery districts for the vehicle routing problem with stochastic demands. *European Journal of Operational Research* 180(3): 997–1010.

Kalcsics J., Nickel S., Schröder M. (2005) Towards a unified territorial design approach – applications, algorithms, and GIS integration. *TOP* 13(1): 1–56.

Kalcsics J., Ríos-Mercado R.Z. (2019) Districting problem. In Laporte G., Nickel S., and Saldanha da Gama F. (editors), *Location Science*, 2nd edition, Chapter 25: 705–743. Springer, Cham, Switzerland.

Kovacs A.A., Parragh S.N., Hartl R.F. (2014) A template-based adaptive large neighborhood search for the consistent vehicle routing problem. *Networks* 63(1): 60–81.

Kovacs A.A., Parragh S.N., Hartl R.F. (2015) The multi-objective generalized consistent vehicle routing problem. *European Journal of Operational Research* 247(2): 441–458.

Laporte G., Nickel S., Saldanha da Gama F. (editors). (2019) *Location Science*. Springer, Cham.

Lei H., Laporte G., Guo B. (2012) Districting for routing with stochastic customers. *EURO Journal on Transportation and Logistics* 1(1-2): 67–85.

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

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

Luo Z., Qin H. Che C. H., Lim A. (2015) On service consistency in multi-period vehicle routing. *European Journal of Operational Research* 243(3): 731–744.

Mancini S., Gansterer M., Hartl R. F. (2021) The collaborative consistent vehicle routing problem with workload balance. *European Journal of Operational Research* 293 (3): 955–965.

Mehrotra A., Johnson E.L., Nemhauser G.L. (1998) An optimization based heuristic for political districting. *Management Science* 44(8): 1100–1114.

Mladenović N., Brimberg J., Hansen P., Moreno-Pérez J. A. (2007) The *p*-median problem: A survey of metaheuristic approaches. *European Journal of Operational Research* 179(3): 927–939.

Nikzad E., Bashiri M., Abbasi B. (2021) A matheuristic algorithm for stochastic home health care planning. *European Journal of Operational Research* 288(3): 753–774.

Ouyang Y., Daganzo C. F. (2006) Discretization and validation of the continuum approximation scheme for terminal system design. *Transportation Science* 40(1): 89–98.

Ríos-Mercado R.Z. (editor). (2020) *Optimal Districting and Territory Design*. International Series in Operations Research and Management Science, Vol. 284. Springer, Cham, Switzerland.

Rodríguez-Martín I., Salazar-González J. J., Yaman H. (2019) The periodic vehicle routing problem with driver consistency. *European Journal of Operational Research* 273(2): 575–584.

Salamatbakhsh-Varjovi A., Tavakkoli-Moghaddam R., Alinaghian M., Najafi E. (2018) Robust periodic vehicle routing problem with time windows under uncertainty: an efficient algorithm. *KSCE Journal of Civil Engineering* 22(11): 4626–4634.

Sandoval M.G., Álvarez-Miranda E., Pereira J., Ríos-Mercado R.Z., Díaz J.A. (2022) A novel districting design approach for on-time last-mile delivery: An application on an express postal company. *Omega* 113: 102687.

Schneider M., Stenger A., Schwahn F., Vigo D. (2014) Territory-based vehicle routing in the presence of time-window constraints. *Transportation Science* 49(4): 732–751.

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

Sungur I., Ren Y., Ordóñez F., Dessouky M., Zhong H. (2010) A model and algorithm for the courier delivery problem with uncertainty. *Transportation Science* 44(2): 193–205.

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

Václavík R., Novák A., Šůcha P., Hanzálek Z. (2018) Accelerating the branch-and-price algorithm using machine learning. *European Journal of Operational Research* 271(3): 1055–1069.

Wang K., Zhen L., Xia J., Baldacci R., Wang S. (2022) Routing Optimization with Generalized Consistency Requirements. *Transportation Science* 56(1): 223–244.

Wen M., Cordeau J.-F., Laporte G., Larsen J. (2010) The dynamic multi-period vehicle routing problem. *Computers & Operations Research* 37(9): 1615–1623.

Wolpert D. H., Macready W. G. (1997) No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation* 1(1): 67–82.

Zhang X., Chen L., Gendreau M., Langevin A. (2022) Learning-based branch-and-price algorithms for the vehicle routing problem with time windows and two-dimensional loading constraints. *INFORMS Journal on Computing*. In Press.

Zhen L., Lv W., Wang K., Ma C., Xu Z. (2020) Consistent vehicle routing problem with simultaneous distribution and collection. *Journal of the Operational Research Society* 71(5): 813–830.

Zhen L., Gao J., Tan Z., Wang S., Baldacci R. (2023) Branch-price-and-cut for trucks and drones cooperative delivery. *IISE Transactions* 55(3): 271–287.

Zhong H., Hall R.W., Dessouky M. (2007) Territory planning and vehicle dispatching with driver learning. *Transportation Science* 41(1): 74–89.

Zhou L., Zhen L., Baldacci R., Boschetti M., Dai Y., Lim A. (2021) A heuristic algorithm for solving a large-scale real-world territory design problem. *Omega* 103: 102442.