Computers & Operations Research 170 (2024) 106756

Contents lists available at ScienceDirect

Computers and Operations Research

journal homepage: www.elsevier.com/locate/cor

An efficient probability-based VNS algorithm for delivery territory design

Ahmed Aly [a], Adriana F. Gabor [a,c,*], Nenad Mladenovic [b,c], Andrei Sleptchenko [b,c]

[a] Department of Mathematics, Khalifa University, Abu Dhabi, United Arab Emirates
[b] Department of Management Science & Engineering, Khalifa University, Abu Dhabi, United Arab Emirates
[c] Research Center on Digital Supply Chain and Operations Management, Khalifa University, Abu Dhabi, United Arab Emirates

ARTICLE INFO

MSC:
90B06
90B80
90C27
90C59

Keywords:
Territory design
Clustering
Variable neighborhood search (VNS)
Randomized algorithm

ABSTRACT

This paper deals with the Delivery Territory Design Problem (DTDP), a districting problem that often occurs in delivery operations. The goal of the problem is to construct clusters of nodes (territories) such that the maximum diameter of a territory is minimized, while the territories designed are balanced w.r.t. some performance measures. We propose to solve the DTDP using a Probabilistic Variable Neighborhood Search (ProbVNS) algorithm based on two local search procedures: a tailored randomized shake procedure that targets both a reduction of infeasibility and diversification, and a deterministic local search based on a linear combination of objective and constraint violation. In addition to searching in different neighborhoods, the ProbVNS also changes the search direction by exploring different penalties for violating constraints. Numerical experiments show that ProbVNS outperforms a recent GRASP with the Path-Relinking (PR) algorithm proposed in the literature in terms of feasibility and objective value. In the tested instances, ProbVNS obtained a lower infeasibility measure in 90% of the instances. For these instances, the average decrease in the objective value was 8.3%, with a maximum decrease of 51%. Finally, the running times of ProbVNS are, on average, 2.7 times lower than those of PR.

1. Introduction

The territory design problem *(TDP)*, sometimes referred to as the districting problem, is the problem of grouping small geographical units called basic units (*BU*s) into larger geographical clusters, named territories, according to relevant planning criteria. Typical applications of the TDP include sales territory design (Zoltners and Sinha, 2005), political districting (Ricca et al., 2013), school districting (Caro et al., 2004), design of pickup and delivery areas (Jarrah and Bard, 2012; Sandoval et al., 2022), design of medical emergency systems (Mayorga et al., 2013) and public service districting (Sudtachat et al., 2016).

TDP and its variants have been researched since the 1960s using a variety of models and algorithms (Ríos-Mercado, 2020; Sandoval et al., 2022). Many variants of the TDP are NP-Hard; therefore, metaheuristics are usually used in order to solve this type of problems (Ríos-Mercado and Fernández, 2009). For state-of-the-art models, algorithms, and applications to the territory design problem, we refer the reader to Ríos-Mercado (2020).

An essential criterion in the TDP is the *compactness* of the districts, which is usually achieved by minimizing a dispersion measure. Commonly used dispersion measures are the total distance to a set of centers (p-median type of objective) or the maximum distance to the cluster centers (p-center type of objective). However, the use of center-based measures has some limitations for problems where it is not clear how to define "good" centers. In such cases, a dispersion measure based on the diameters of the territories is more appropriate (Ríos-Mercado and Escalante, 2016).

A special class of TDPs is the *Commercial Territory Design Problems* (CTDPs), where compactness is minimized subject to additional constraints imposing connected and balanced districts. A district is *connected* if, between any two points in the districts, there exists a path contained in the district. A set of districts is *balanced* if certain attributes, such as demand, number of customers, and workload, are evenly distributed among districts.

This paper addresses a relaxed version of CTDP called the *Delivery Territory Design Problem* (DTDP). It aims to design a set of p disjoint territories such that the maximum diameter of a territory is minimized while balancing a set of attributes. Unlike in the Commercial Territory Design Problem, the connectivity criterion is relaxed, allowing two nodes to be assigned to the same territory as long as they are connected through a path, not necessarily fully contained in the district. For delivery applications, this models the situation where a driver may serve two locations in the city, without serving all the customers on the connecting path.

The DTDP is motivated by real world applications from delivery and logistics software development companies in the UAE. These companies

* Corresponding author at: Department of Mathematics, Khalifa University, Abu Dhabi, United Arab Emirates.
E-mail addresses: ku100045073@alumni.ku.ac.ae (A. Aly), adriana.gabor@ku.ac.ae (A.F. Gabor), andrei.sleptchenko@ku.ac.ae (A. Sleptchenko).

https://doi.org/10.1016/j.cor.2024.106756
Received 29 August 2023; Received in revised form 7 April 2024; Accepted 25 June 2024
Available online 2 July 2024

are primarily interested in designing delivery areas such that workload is fairly distributed among drivers. Connectivity of the districts was not seen as being an essential aspect, as long as there exists a road between the delivery points assigned to a driver. This is in line with Zhou et al. (2021), who also noted that in delivery and routing applications, imposing connectivity can strongly affect the quality of the solution. The reason for working with delivery districts rather than routes originates in the belief that assigning drivers to the same areas in the city improves the efficiency of the delivery process, as they become familiar with the traffic and the existing delivery zones.

To the best of our knowledge, Variable Neighborhood Search heuristics have not been previously used to solve CTDP. In this paper, we propose a probability-based VNS (ProbVNS) that uses as objective a linear combination of the maximum diameter of a district and constraints violation. The novelty of (ProbVNS) relies in the way infeasibility is addressed in both the Shake (RND-Balance) and Local Search (LS-NBI) procedures. The Shake algorithm tries to improve infeasibility by reassigning nodes from unbalanced districts according to a probabilistic procedure, where more unbalanced districts have a higher chance to be considered. Furthermore, the VNS alternates between directions that improve the objective or improve infeasibility by changing the weights of constraint violations in the local search procedure. The resulting algorithm outperforms significantly the path-relinking algorithm proposed in Ríos-Mercado and Escalante (2016) in both objective and feasibility, on the majority of the tested instances. On the instances where infeasibility is improved, the average decrease in objective is 8.3%, with a maximum decrease of 51%. In our experiments, the VNS is on average 2.7 times faster than the algorithm proposed by Ríos-Mercado and Escalante (2016).

The remainder of this paper is structured as follows. Section 2 discusses the literature on related territory design problems and our contribution. In Section 3, we describe the problem and provide the mathematical model of the DTDP, while in Section 4, we outline the Probability-based VNS heuristic and its components. In Section 5, we first describe the two sets of benchmark instances used; the first was previously proposed in Ríos-Mercado and Escalante (2016), while the second set is new, modeling different city topologies. Subsequently, we present the comparison between our algorithm and the algorithm proposed in Ríos-Mercado and Escalante (2016) on both sets of instances. Conclusions and further research are presented in Section 6.

2. Related work

The Delivery Territory Design Problem (DTDP) is related to the commercial territory design problem (CTDP) introduced by Ríos-Mercado and Fernández (2009). In CTDP, the objective is to minimize a p-center dispersion function, subject to planning criteria such as disjoint districts, multiple attribute balancing, and district connectivity. The authors propose a reactive GRASP algorithm enhanced with self-adjustment of the restricted candidate list (RCL list) and filtering, which avoids executing the local search phase for unpromising solutions. Recently, Ríos-Mercado et al. (2021) extended the classical location–allocation method used for one-criterion balancing to solve CTDP with p-median dispersion objective and multi-criteria balancing.

The performance of several MIP formulations of CTDP, with the p-center and p-median related objectives, are discussed in Salazar-Aguilar et al. (2011a). The authors present a solution framework based on branch and bound and a cut generation strategy and conclude that models with an p-median objective function can be solved faster. Improved exact methods for the p-center dispersion measure are presented in Sandoval et al. (2020).

A variant of CTDP with a budget constraint on routing decisions was studied by Ríos-Mercado and Salazar-Acosta (2011). This problem is solved by a GRASP procedure that incorporates adaptive memory and strategic oscillation. Multi-objective variants of the CTDP with p-center objectives have been discussed in Salazar-Aguilar et al. (2011b, 2012, 2013). Salazar-Aguilar et al. (2011b) propose an exact optimization method to obtain Pareto fronts for the problem where both the compactness of the territory and the attribute balance are simultaneously optimized. Salazar-Aguilar et al. (2012, 2013) solve larger instances using heuristic methods based on scatter search and GRASP, respectively.

Ríos-Mercado and Escalante (2016) propose a new variant of CTDP, where a diameter-based dispersion measure is minimized instead of a center-based one, subject to balancing a set of attributes. They show that a GRASP metaheuristic enhanced with path-relinking **(PR)** is able to find good results in terms of the dispersion measure. The DTDP problem we study is related to this variant of CTDP. However, we relax the connectivity constraints by allowing two nodes to be contained in the same district even though the connecting path is not fully contained in it.

In this paper, we propose to solve the DTDP using a variant of the Variable Neighborhood Search *(VNS)* heuristic. This heuristic was introduced in Brimberg (1996) and later formalized in Mladenović and Hansen (1997). The core paradigm behind VNS is to systematically change neighborhood structures to prevent plateaus at local optima (Mladenović and Hansen, 1997). Over the years, VNS has been extensively researched and now boasts a wide array of extensions (Hansen et al., 2017); General VNS, Variable Neighborhood Descent, and Reduced VNS, to name a few. A comprehensive overview of different versions of VNS can be found in Brimberg et al. (2023).

VNS has been successfully used to find good quality solutions to several variants of clustering problems, such as p-center and p-median problems. Mladenović et al. (2003) proposed a Basic VNS meta-heuristic with vertex substitution local search to solve the p-center problem. Their results show that VNS, on average, outperforms Tabu Search, while Tabu Search was still better for small values of p. Fleszar and Hindi (2008) showed that a VNS meta-heuristic based on the generalized assignment problem can efficiently solve the capacitated p-median problem. Mladenović N. Alkandari et al. (2019) successfully used Basic VNS to solve the obnoxious p-median problem in the Less-is-more approach (LIMA). Brimberg et al. (2019) showed that Skewed General VNS also performs well for the capacitated clustering problem. They observed that evaluating moves prior to accepting inferior solutions is preferable to random shaking procedures. The DTDP is in essence a clustering problem with balancing constraints. Note that while capacitated clustering problems impose only an upperbound on the load in a district, balancing constraints also impose a lowerbound, increasing the difficulty of the problem.

Contribution: In this work, we propose a Probability-based VNS heuristic (ProbVNS) to solve DTDP, a relaxed version of the CTDP problem introduced by Ríos-Mercado and Escalante (2016). To the best of our knowledge, the VNS framework has not previously been tested on CTDP-related problems, although GRASP and local search procedures have been proven quite successful (Ríos-Mercado and Escalante, 2016). The ProbVNS uses a merit function defined as a linear combination of the objective function and a measure of constraint violation, commonly used in local searches for TDP. However, ProbVNS systematically changes the weights of the constraint violation, allowing to alternate between solutions that improve the diameter or reduce infeasibility. Moreover, infeasibility is also reduced in the Shake procedure, where districts with higher infeasibility are more likely to be modified. To assess the performance of the proposed ProbVNS, we use a set of benchmark instances similar to the one proposed in Ríos-Mercado and Escalante (2016) and propose a new set of benchmark instances that proved harder to solve. Through numerical experiments, we show that the proposed ProbVNS procedure outperforms the path relinking algorithm (PR) of Ríos-Mercado and Escalante (2016) in terms of feasibility and objective value. In 71.5% of the 270 instances tested, ProbVNS outperforms PR in both feasibility and objective value. In cases with improved feasibility, the average decrease in the objective was 8.3%, with a maximum decrease of 51%.

3. Problem description and mathematical model

The input to the DTDP is a connected graph $G = (V, E)$, where the nodes represent the set of basic units (BU) and the edges represent the streets between BUs. For each node, we are given a set of attributes A, such as number of customers, product demand, and workload. The value of the attribute $a \in A$ of node $i \in V$ will be denoted by w_i^a.

We denote a p-partition of the set V by $X = (X_1, \dots, X_p)$ where $X_m \subset V$ is called a territory of V. We will use $X_m \in X$ to indicate that X_m is part of a partition X and call X_m a territory or district of X. We denote the collection of all p-partitions of the set V by Π.

The size of a territory X_m with respect to the attribute $a \in A$ is defined by $w^a(X_m) = \sum_{i \in X_m} w_i^a$. Ideally, the constructed districts should have the same size in each attribute, that is, $w^a(X_m) = \mu^a$ for each district X_m and attribute $a \in A$, where $\mu^a = \frac{\sum_{i \in V} w_i^a}{p}$. However, due to the discrete nature of the problem, this is often impossible to achieve; hence, relaxed criteria to ensure approximately equally sized districts will be used.

We call a partition X *balanced* w.r.t an attribute a, if each district $X_m \in X$ is *balanced*, that is, $\frac{w^a(X_m)}{\mu^a} \in [1 - \tau^a, 1 + \tau^a]$. Here, τ^a is a tolerance parameter that is pre-specified by the user.

The goal of DTDP is to find a balanced p-partition w.r.t. each attribute that minimizes the maximum diameter of the created territories. The mathematical model can be formulated as a combinatorial optimization model following (Ríos-Mercado and Escalante, 2016):

$$\min_{X \in \Pi} \max_{X_m \in X} \max_{i,j \in X_m} \{d_{ij}\} \tag{1}$$

s. t.

$$w^a(X_m) \in [(1 - \tau^a)\mu^a, (1 + \tau^a)\mu^a], \quad X_m \in X, a \in A \tag{2}$$

In this formulation, the objective function (1) minimizes the maximum diameter in a p-partition X. Constraints (2) ensure that the districts are balanced in all attributes.

Note that when all attributes have value zero, DTDP reduces to a p-maximum diameter clustering problem, which is proven to be NP-hard in Gonzalez (1985).

4. Probability-based variable neighborhood search algorithm (ProbVNS)

We propose to solve DTDP using a Probability-based VNS (ProbVNS). To evaluate the quality of a solution $X = (X_1, \dots, X_p)$, the ProbVNS procedure uses the *merit function* $\Psi(X)$, defined as a linear combination between the maximum diameter of the districts in X and a measure of constraint violation. The reason for using this type of merit function is that in many TDPs, it is hard to find feasible solutions; hence, solutions with low infeasibility may be acceptable. More precisely,

$$\Psi(X) = F(X) + \lambda G(X), \tag{3}$$

where

$$F(X) = \left(\frac{1}{d_{max}}\right) \max_{m \in \{1 \dots p\}} \max_{i,j \in X_m} \{d_{ij}\}$$

$$d_{max} = \max_{i,j \in V} \{d_{ij}\},$$

and

$$G(X) = \sum_{m=1}^{p} \sum_{a \in A} g^a(X_m), \tag{4}$$

where,

$$g^a(X_m) = \frac{1}{\mu^a} max\{w^a(X_m) - (1 + \tau^a)\mu^a, (1 - \tau^a)\mu^a - w^a(X_m), 0\}.$$

Remark that $G(X)$ is a measure of infeasibility, and for a feasible solution X, $G(X) = 0$. Note that Ríos-Mercado and Escalante (2016) uses an equivalent merit function, $\lambda_{\mathrm{PR}} F(X) + (1 - \lambda_{\mathrm{PR}}) G(X)$ with $\lambda_{\mathrm{PR}} \in [0, 1]$. We chose to present the merit function as in (3) as it conveys more directly the focus of the search on objective or feasibility improvement for different values of λ.

4.1. General procedure

A general outline of the proposed Probability-based Variable Neighborhood Search (ProbVNS) procedure is given in Algorithm 1.

Algorithm 1 Probability-based VNS($X_{in}, k_{max}, l_{max}, r_{max}$)

1: **Input:** Initial partition X_{in}
2: k_{max} : max. nr. of neighborhoods in $RND - Balance$
3: l_{max}: max. nr. of consecutive iterations without improvement
4: r_{max}: max. nr. of times $\lambda = \lambda_1$ without merit improvement
5: Λ: set of values for the weight of $G(X)$ in Ψ
6: $l \leftarrow 1$
7: $\Psi_{\lambda_1} \leftarrow \infty$
8: $r \leftarrow 1$
9: **while** $l \le l_{max}$ **do**
10: $\quad LambdaImprovement \leftarrow FALSE$
11: $\quad i_\lambda \leftarrow 1$ ⊳ index for the set of Λ
12: $\quad$ **while** $i_\lambda \le |\Lambda|$ and $r \le r_{max}$ **do**
13: $\quad\quad k \leftarrow 1$
14: $\quad\quad \Psi(X) = F(X) + \lambda_i G(X)$
15: $\quad\quad$ **while** $k \le k_{max}$ **do**
16: $\quad\quad\quad X' \leftarrow$ **RND − Balance**(X, k)
17: $\quad\quad\quad X'' \leftarrow$ **LS − NBI**(X')
18: $\quad\quad\quad$ **if** $\Psi(X'') < \Psi(X)$ **then**
19: $\quad\quad\quad\quad LambdaImprovement \leftarrow TRUE$
20: $\quad\quad\quad\quad X \leftarrow X''$
21: $\quad\quad\quad\quad k \leftarrow 1$
22: $\quad\quad\quad\quad l \leftarrow 1$
23: $\quad\quad\quad$ **else**
24: $\quad\quad\quad\quad k \leftarrow k + 1$
25: $\quad\quad\quad$ **end if**
26: $\quad\quad$ **end while**
27: $\quad\quad$ **if** $LambdaImprovement$ = TRUE **then**
28: $\quad\quad\quad$ **if** $i_\lambda > 1$ **then**
29: $\quad\quad\quad\quad i_\lambda \leftarrow 1$
30: $\quad\quad\quad$ **else**
31: $\quad\quad\quad\quad$ **if** $\Psi(X) < \Psi_{\lambda_1}$ **then**
32: $\quad\quad\quad\quad\quad \Psi_{\lambda_1} \leftarrow F(X) + \lambda_1 G(X)$
33: $\quad\quad\quad\quad$ **else**
34: $\quad\quad\quad\quad\quad r \leftarrow r + 1$
35: $\quad\quad\quad\quad$ **end if**
36: $\quad\quad\quad$ **end if**
37: $\quad\quad$ **else**
38: $\quad\quad\quad i_\lambda \leftarrow i_\lambda + 1$
39: $\quad\quad$ **end if**
40: $\quad$ **end while**
41: $\quad l \leftarrow l + 1$
42: **end while**

The ProbVNS uses a randomized Shake procedure (RND-Balance) to diversify the search and reduce infeasibility, and a deterministic local search (LS-NBI) to improve the merit function $\Psi(X)$ defined by (3). Throughout the algorithm, the weights λ of the infeasibility measure $G(X)$ used in the definition of the merit function will be changed, depending on whether the goal is to decrease the infeasibility (higher values of λ) or to encourage a decrease in the objective, even at the expense of increasing the infeasibility (lower values of λ).

ProbVNS receives as input an initial solution X_{in}, k_{max}, the maximum number of neighborhoods used in the RND-Balance procedure, the set of weights Λ used in the merit function Ψ, the maximum number of consecutive iterations without improvement, l_{max}, and r_{max}, the maximum number of iterations with $\lambda = \lambda_1$ without an improvement of the merit function (lines 1–5). We assume that the values of Λ are in increasing order.

For each value of λ, the algorithm explores the k_{max} neighborhoods and executes RND-Balance followed by the local search procedure LS-NBI (lines 15–24). If the value of the merit function Ψ is improved in one of the neighborhoods, the algorithm returns to the first neighborhood ($k \leftarrow 1$) and restarts counting the iterations without improvement ($l \leftarrow 1$) (lines 18–22). If the value of the merit function is not improved in one neighborhood, the algorithm switches to exploring the next neighborhood (line 24).

Assume there is an improvement in one of the neighborhoods for a certain value of λ, say λ^*. If $\lambda^* > \lambda_1$, the algorithm resets $\lambda = \lambda_1$ (line 29) and starts again exploring the k_{max} neighborhoods. As in this case, the weight of the infeasibility measure $G(X)$ is decreased, the algorithm has the opportunity to reduce the objective at the expense of the feasibility, with the hope that during future executions of RND-Balance and LS-NBI, the feasibility of the solution will be restored. If $\lambda^* = \lambda_1$, the algorithm checks whether a better merit has been obtained in previous iterations for λ_1. If this is not the case, the counter r of the number of times the value of λ has been reset to λ_1 without any improvement of the merit function is increased (lines 31–35). Finally, if for a certain λ, no improvement is obtained in any of the neighborhoods, the next value of λ is explored (line 38). Observe that in case the current best solution is infeasible, increasing λ leads to a change in the search direction in an attempt to reduce the infeasibility. If the current solution is feasible, i.e., $G(X) = 0$, changing λ simply leads to new iterations through shake and local search. Recall that LS-NBI may choose to increase the infeasibility and reduce the objective in order to obtain a better merit function. The procedure is repeated until no improvement is achieved for l_{max} consecutive iterations, where one iteration corresponds to lines 12–40 in Algorithm 1.

The components of ProbVNS are described in more detail in the following sections. The construction of the initial solution is given in Section 4.2, while the exact description of RND-Balance and LS-NBI will be described in Section 4.3.

4.2. Initial solution

To generate the initial solution, we use the construction phase of the GRASP algorithm outlined in Ríos-Mercado and Escalante (2016). We refer the reader to Appendix A for a more detailed description. The procedure generates n_{sol} solutions and chooses the one with the lowest merit function as the initial solution X_{in}. Each solution is generated in three stages. In the first stage, the greedy heuristic proposed in Erkut et al. (1994) is used to solve a p-dispersion problem to generate the initial set of p seeds around which districts will evolve. The algorithm chooses a random node as the center of the first territory, and the rest of the centers are chosen according to a maximum dispersion criterion. Following the selection of the centers, $\delta|V|$ unassigned nodes are assigned to their closest district without taking into account the balancing constraints.

The second stage of construction attempts to assign the remaining $(1-\delta)|V|$ unassigned nodes while trying to balance attributes. A greedy randomized adaptive algorithm is used to evaluate the balance and dispersion constraints according to Eq. (3). As the second stage does not guarantee that all nodes can be assigned to a territory due to balancing constraints, a third stage assigns each remaining unassigned node, if any, to its closest territory. Observe that due to the last phase, X_{in} may not be balanced.

4.3. The RND-Balance procedure

Before we describe the RND-Balance procedure in detail, we introduce some terminology. Consider a p-partition $X = (X_1, \dots, X_p)$ of the nodes in V. We will call two territories X_i and X_j *adjacent* if there exists an edge $(v_1, v_2) \in E$ with $v_1 \in X_i$ and $v_2 \in X_j$. We define a *neighborhood* $N_k(X)$ as the set of solutions obtained by reallocating k nodes from a territory X_i to an adjacent territory X_j. For each district X_i, let $AD(X_i)$ denote the territories adjacent to X_i and $V_{adj}(X_i, X_j)$ the set of nodes/vertices in X_i, adjacent to at least one node in X_j.

The RND-Balance procedure is described in detail in Algorithm 2. Denote by $UINF_k(X)$ the set of unbalanced districts with more than k nodes. If $UINF_k(X) \neq \emptyset$ the procedure chooses a random district X_i in $UINF_k(X)$ with probability $p_1(X_i) = \frac{G(X_i)}{\sum_{i \in UINF_k(X)} G(X_i)}$ (line 6). If all districts with more than k nodes are balanced, X_i is chosen with probability $\frac{1}{|U_k(X)|}$, where $U_k(X)$ is the set of districts with more than k nodes (line 8). After choosing a first district X_i, a second district X_j is randomly chosen from $AD(X_i)$, the adjacent districts to X_i with probability $p_2(X_j) = \frac{1}{|AD(X_i)|}$ (line 10). Note that since G is connected, $AD(X_i) \neq \emptyset$. Finally, a random node v is chosen from $V_{adj}(X_i, X_j)$, and the k closest nodes connected to v are moved from X_i to X_j (lines 11–14). Note that X_i is chosen from districts with more than k nodes; therefore, there exist k nodes to be moved to the adjacent district.

Algorithm 2 RND-Balance(X, k)

1: $U_k(X) := \{X_i \in X : |X_i| \geq k\}$
2: $UINF_k(X) := \{X_i \in X : |X_i| \geq k, G(X_i) > 0\}$
3: $AD(X_i) := \{X_j \in X : X_i$ and $X_j\}$ are adjacent , $i = 1, \dots, p$
4: $V_{adj}(X_i, X_j) := \{(v_1, v_2) : v_1 \in X_i$ and $v_2 \in V_2\}$
5: **if** $UINF_k(X) \neq \emptyset$ **then**
6:     Choose a district X_i with probability $p_1(X_i) = \frac{G(X_i)}{\sum_{i \in UINF_k(X)} G(X_i)}$
7: **else**
8:     Choose a random district X_i from $U_k(X)$ with probability $\frac{1}{|U_k(X)|}$
9: **end if**
10: Choose district X_j adjacent to X_i with probability $p_2(X_j) = \frac{1}{|AD(X_i)|}$
11: Choose a node $v \in V_{adj}(X_i, X_j)$ with probability $p_3 = \frac{1}{|V_{adj}(X_i, X_j)|}$
12: $P(v) :=$ set of k closest connected nodes to v in X_i
13: $X_i \leftarrow X_i \setminus P(v)$
14: $X_j \leftarrow X_j \cup P(v)$
15: Update $AD(X_j)$ for all $X_j \in AD(X_i)$
16: Update $V_{adj}(X_i, X_j)$ and $V_{adj}(X_j, X_i)$ for all $X_j \in AD(X_i)$
17: **Return** X

Furthermore, since $p_1(X_i)$ is proportional to $G(X_i)$, nodes are more likely to be removed from districts with higher infeasibility, if unbalanced districts exist. As a consequence, the RND-balance procedure contributes to diversifying the solution and reducing the infeasibility.

4.4. Node best improvement local search (LS-NBI)

Algorithm 3 describes the local search procedure used in ProbVNS, Node Best Improvement (LS-NBI). LS-NBI iterates over all districts of a solution X. For each territory m, it evaluates the impact of $move(m, i)$, defined as moving node $i \in V \setminus \{X_m\}$ to district m, and performs the move that leads to the best improvement in Ψ (lines 5–11). LS-NBI terminates when either no improved solution is found, or the maximum number of moves max_{moves} is reached (line 3).

5. Computational experiments

This section presents the results of the numerical experiments performed to test the proposed algorithms. All algorithms were coded in Python 3.9 and executed on Intel(R)® Core(TM) i7-4790 CPU @ 3.60 GHz with 16 GB RAM.

Algorithm 3 LS-NBI(X, max_moves)

$nmoves \leftarrow 0$
$optima \leftarrow FALSE$
while $nmoves < max\_moves$ and $optima = FALSE$ **do**
  $improvement \leftarrow FALSE$
  for all $m \in \{1, \dots, p\}$ **do**
    Find $move(m, i)$ that leads to highest decrease of Ψ over all $i \in V \setminus \{X_m\}$
    if Ψ is improved **then**
      Perform $move(m, i)$
      $improvement \leftarrow TRUE$
    end if
  end for
  if $improvement \leftarrow TRUE$ **then**
    $nmoves = nmoves + 1$
    $optima = FALSE$
  else
    $optima = TRUE$
  end if
end while

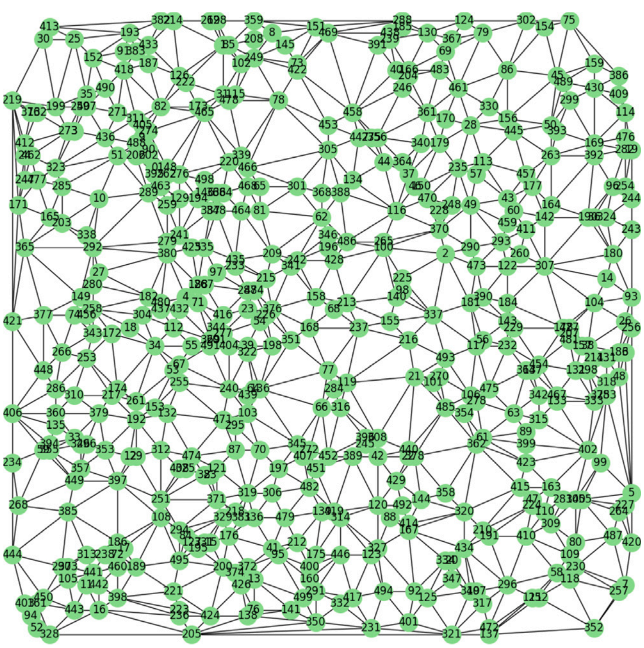

Fig. 1. Example of a planar graph obtained by triangulation.

The computational experiments were performed on two types of randomly generated planar graphs. In the first type (T-graphs), a predefined number of nodes (500, 600, or 700) were generated uniformly in an area of 100×100. The neighboring points were connected using Delaunay triangulation with Euclidean distances on each edge (see Fig. 1). This set of instances mimics the instances in Ríos-Mercado and Fernández (2009).

The second group of instances (G-graphs) was generated based on the requirements of the delivery companies we talked to, to mimic diverse topologies of delivery regions (e.g. more dense in certain areas, such as the center or corners or with a highway traversing the city).

These instances have $\{486, 600, 726\}$ nodes and are generated as follows. We started with a grid graph of $n \times n$ nodes, where $n \in \{27, 30, 33\}$, divided into 7 regions as in Fig. 2. We generated three types of graphs: Graph Type Center, Graph Type Diagonal, and Graph Type Corners (G-C, G-D, G-CN) by randomly removing one-third of the nodes from certain regions while maintaining the connectivity of the graph.

More precisely, we randomly removed $\frac{2}{9}n^2$ nodes together with their adjacent edges from the regions R3, R4, and R5 for G-C; R1, R4, and R7 for G-D; R1, R2, R6, and R7 for G-CN. Finally, we randomly removed $\frac{1}{9}n^2$ nodes and their adjacent edges from the remaining regions for each graph type. Examples of generated graphs are shown in Fig. 3. The instance generation codes and the instances used are available on GitHub (Aly et al., 2023).

Each node has three attributes: demand, workload, and number of customers. These attributes were generated using a uniform distribution on $[15, 400]$, $[15, 100]$, and $[4, 20]$, respectively, following (Ríos-Mercado and Fernández, 2009). For each edge, the distance was generated according to a uniform distribution on $[5, 20]$ km. In all instances, $p = 10$, and for each activity, $\tau^a = 0.05$.

The values of the ProbVNS parameters were chosen as follows. In the construction of the initial solution, we chose $\alpha_{RCL} = 50\%$. We construct $n_{sol} = 100$ solutions according to the construction phase (GRASP-algorithm) described in Ríos-Mercado and Escalante (2016), see also to Appendix A. The maximum number of iterations in LS-NBI $max_{moves} = 1000$, while in $RND - Balance$, we set $k_{max} = 5$, $l_{max} = 4$, and $m_{max} = 4$. We have experimented with $k_{max} \in \{5, 7, 10\}$; however, no significant improvement was observed for $k_{max} > 5$. For ProbVNS, we chose $\Lambda = \{0.1, 0.4, 0.7, 1\}$.

5.1. Performance of ProbVNS

To assess the performance of ProbVNS, we compared it with a recent algorithm proposed in Ríos-Mercado and Escalante (2016), the GRASP with static Path-Relinking (PR) algorithm. As the PR was designed for CTDP, we dropped the requirement that two nodes in a district must be connected through a path of nodes in the same district. For PR, we chose $\lambda_{\text{PR}} = 0.7$, which corresponds to $\lambda_{\text{ProbVNS}} = 0.42$ due to the scaling of the objective. For both algorithms, we ran 90 instances on T-graphs similar to Ríos-Mercado and Escalante (2016). Furthermore, we ran 270 instances on G-graphs, 30 for each combination of graph type (G-D, G-C, G-CN) and graph size (27×27, 30×30, 33×33).

Fig. 4 illustrates solutions at different stages of the algorithm for an instance of graph type G-D of size 27×27. The first solution, obtained by the construction algorithm described in Section 4.2, has a maximum diameter of 162 and an infeasibility measure $G(X) = 4.88$. The second solution, obtained after one iteration of ProbVNS in Algorithm 1 has a considerably lower infeasibility measure ($G(X) = 0.38$) and an increased maximal diameter of 175. The final solution, depicted in the third part of Fig. 4, is feasible, and has an increase in diameter to 187. Note that in order to balance the load, the algorithm is forced to give up on connectivity in the final solution.

5.1.1. Objective and infeasibility

As finding feasible solutions for TDP is hard, a common analysis of algorithms for TDP includes a comparison of both the objective value and the infeasibility of a solution. For measuring the objective, we will use the maximum diameter of a district, while for measuring the infeasibility (INF_{ProbVNS} and INF_{PR}), we will use the function $G(X)$ defined in (4).

Table 1 and Fig. 5 present a comparison of objective and infeasibility for the different types of graphs and different graph sizes. The rows with $INF_{ALG} = 0$, with $ALG \in \{ProbVNS, PR\}$ indicate the percentage of instances where the respective ALG found feasible solutions. The p-values reported in Fig. 5 correspond to the paired t-test (at significance level $\alpha = 0.05$) for the mean infeasibility ($MINF$) and mean objective ($MOBJ$) in the corresponding sets of instances: $MINF_{\text{ProbVNS}} \geq MINF_{\text{PR}}$, $MINF_{\text{ProbVNS}} = MINF_{\text{PR}}$ and $MINF_{\text{ProbVNS}} \leq MINF_{\text{PR}}$ in the upper-part of the figure and $MOBJ_{\text{ProbVNS}} \geq MOBJ_{\text{PR}}$, $MOBJ_{\text{ProbVNS}} = MOBJ_{\text{PR}}$ and $MOBJ_{\text{ProbVNS}} \leq MOBJ_{\text{PR}}$ in the lower-part. Although the p-values of the three tests are connected, we chose to report all of them for the ease of the presentation. In our analysis, for each type of graph, we

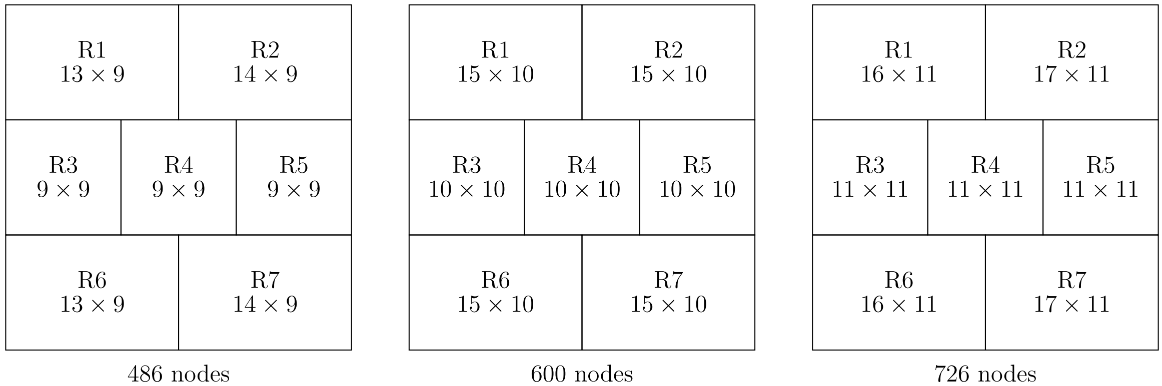

Fig. 2. Graphical representation of the regions of the graph.

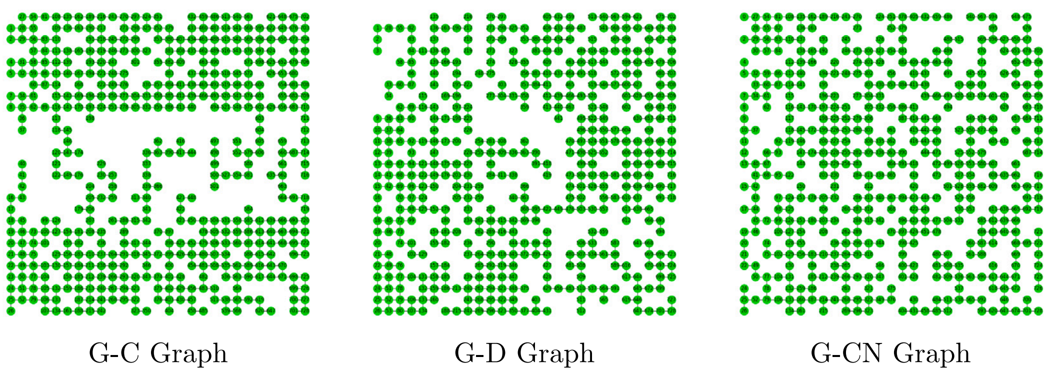

Fig. 3. Examples of the created graph of different types.

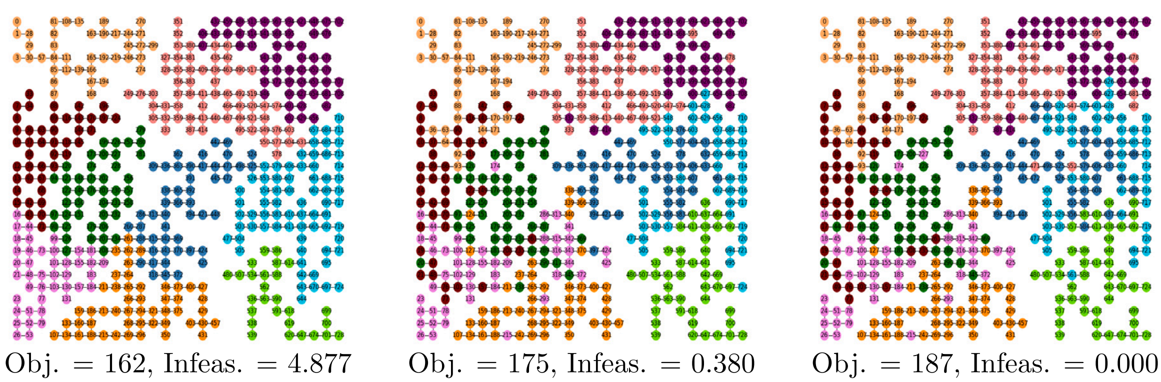

Fig. 4. Solutions for Construction, First, and Last iteration of ProbVNS for an instance of G-D 27 × 27.

Table 1
Objective and infeasibility comparison between ProbVNS and PR.

	T-Graphs	G-C graphs	G-CN graphs	G-D graphs
$INF_{\text{PR}} = 0$	41.11%	27.78%	32.22%	34.44%
$INF_{\text{ProbVNS}} = 0$	100.00%	92.22%	74.44%	75.56%
$INF_{\text{ProbVNS}} < INF_{\text{PR}}$	100.00%	94.44%	84.44%	81.11%
$MINF_{\text{PR}}$	0.004305	0.018860	0.011780	0.006345
$MINF_{\text{ProbVNS}}$	0.000000	0.000326	0.004073	0.020020
$OBJ_{\text{ProbVNS}} < OBJ_{\text{PR}}$	93.33%	68.89%	91.11%	94.44%
$INF_{\text{ProbVNS}} < INF_{\text{PR}}$ & $OBJ_{\text{ProbVNS}} < OBJ_{\text{PR}}$	93.33%	63.33%	75.56%	75.56%
$INF_{\text{ProbVNS}} > INF_{\text{PR}}$ & $OBJ_{\text{ProbVNS}} > OBJ_{\text{PR}}$	0.00%	0.00%	0.00%	0.00%

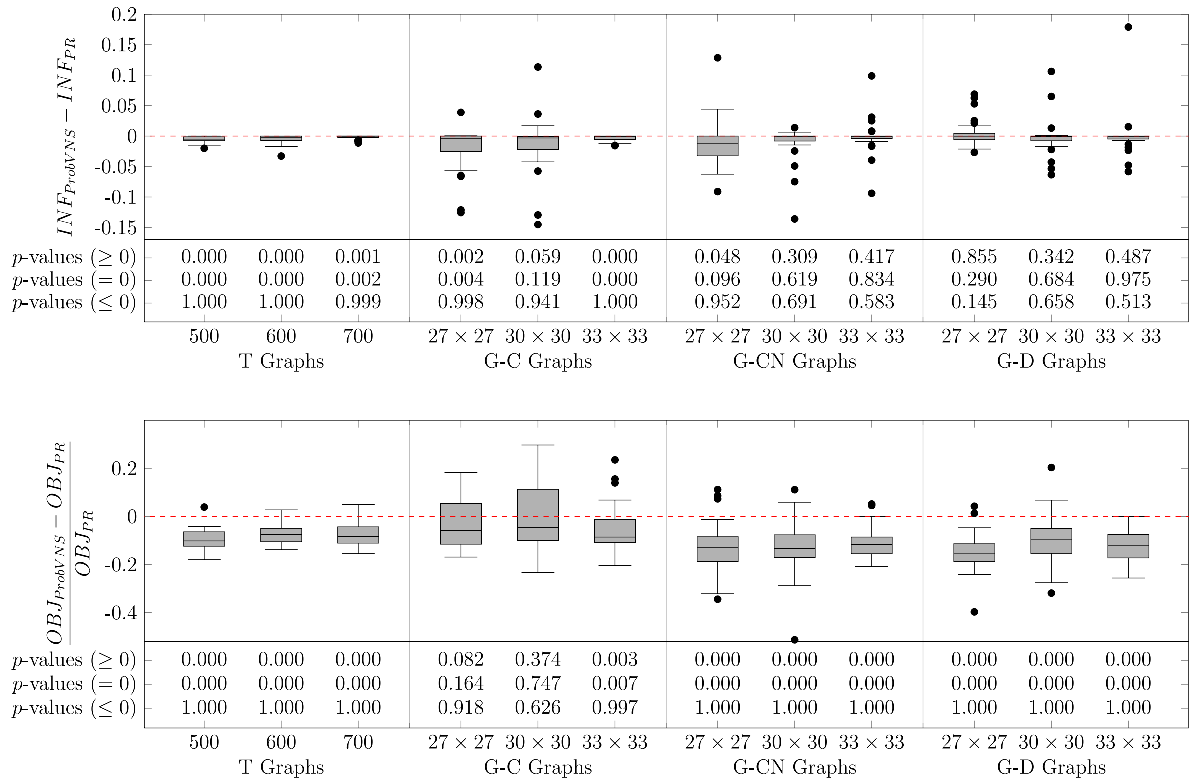

Fig. 5. Comparison of objective and infeasibility for ProbVNS and PR.

will analyze first the infeasibility measure, then the joint measures of objective and infeasibility. Looking at objective in isolation may be deceiving, as a solution may have a very low objective, while violating significantly the balancing criteria.

ProbVNS was able to find a feasible solution in all T-type graphs, while PR only in 41.1% of these instances. However, note that, although PR is not able to find feasible solutions in more than half of instances, it does produce solutions with low infeasibility ($MINF_{PR} = 0.004$). When taking the objective values into account, ProbVNS obtained solutions with lower infeasibility and lower objective values in 93.3% of the T-graphs (see Table 1). For the cases with lower infeasibility, ProbVNS had a mean reduction in objective of 8% and a maximum reduction of 17%. In none of the remaining 6.7% of the T-graphs, PR was better on both measures. The p-values reported in Fig. 5 confirm that for all the T-graphs, both $MOBJ_{\text{ProbVNS}} < MOBJ_{\text{PR}}$ and $MINF_{\text{ProbVNS}} < MINF_{\text{PR}}$.

ProbVNS found feasible solutions in 80.74% of the G-graphs, while PR found feasible solutions only in 31.48% of these graphs. A more detailed analysis of the results for type G-graphs indicates that ProbVNS was able to find feasible solutions in 92.22% of the G-C graphs, while PR only in 27.78%. This type of G-graph seemed to be the hardest type for PR and the easiest for ProbVNS in terms of feasibility. ProbVNS obtained lower infeasibility in 94.44% of G-C graphs and both lower infeasibility and lower objective in 63.33% of instances. The average reduction in objective for the cases with reduced infeasibility was 2%. The paired t-tests for comparing mean objective and mean infeasibility (see Fig. 5) indicate that for graphs of size 33 × 33, ProbVNS has significantly lower mean infeasibility and mean objective. For the cases with lower infeasibility, the objective values obtained by ProbVNS were on average 5.4% lower than the ones obtained by PR. For G-C graphs of size 27 × 27, ProbVNS has a significantly lower mean infeasibility, while the hypothesis that $MOBJ_{ProbVNS} \geq MOBJ_{PR}$ cannot be rejected (p-value is 0.082). However, the infeasibility was lower in 93.3% of these graphs and in 60% of these graphs both measures were lower. The average reduction in objective for the cases with lower infeasibility for ProbVNS was 2.3%. Finally, for size 30 × 30, both the hypotheses that $MOBJ_{ProbVNS} \geq MOBJ_{PR}$ and $MINF_{ProbVNS} \geq MINF_{PR}$ cannot be rejected. Remark that in 90% of instances the infeasibility measure was lower in the solutions obtained by $ProbVNS$ and in 53.3% of the instances with improved infeasibility the objective was lower. However, on average, the objective values obtained by ProbVNS for these graphs, were 1% higher than the objective values obtained by PR.

For ProbVNS, the hardest graph type in terms of percentage of feasible solutions was G-CN. For this graph type, ProbVNS found a feasible solution in 74.44% of the cases (PR found a feasible solution in 32.22% of these instances). In 84.44% of the G-CN graphs, $INF_{ProbVNS} < INF_{PR}$ and in 75.56%, both the infeasibility and the objective value were lower. In the cases with improved infeasibility, the average reduction in objective value obtained by ProbVNS was 12.2%. The paired t-test for the difference in means in objective and infeasibility among the two methods, reveals that for the graph size 27 × 27 the hypothesis $MINF_{ProbVNS} \geq MINF_{PR}$ can be rejected (p-value is 0.048) while for the other G-CN graphs, there is not enough evidence to reject the hypothesis $MINF_{ProbVNS} > MINF_{PR}$ at $\alpha = 0.05$ (see Fig. 5). In all the G-CN graphs, the hypothesis $MOBJ_{ProbVNS} > MOBJ_{PR}$ is rejected.

Finally, ProbVNS was able to find feasible solutions in 75.56% of the G-D graphs, while PR found feasible solutions in 34.44% of the graphs. The infeasibility measure was lower in 81.11% of the graphs, while 75.56% of graphs have both lower infeasibility and objective. For the cases with lower infeasibility, ProbVNS registered, on average, a 11.4% reduction in objective value. The statistical analysis for the mean infeasibility presented in Fig. 5 indicates that $MINF_{ProbVNS} \geq MINF_{PR}$ cannot be rejected, while $MOBJ_{ProbVNS}$ is significantly lower than $MOBJ_{PR}$. The discrepancy between the detailed instance analysis and the mean analysis can be explained by the presence of a few outliers (see Fig. 5).

In conclusion, for 90% of instances, ProbVNS had a lower infeasibility measure than PR. The average reduction in objective value over

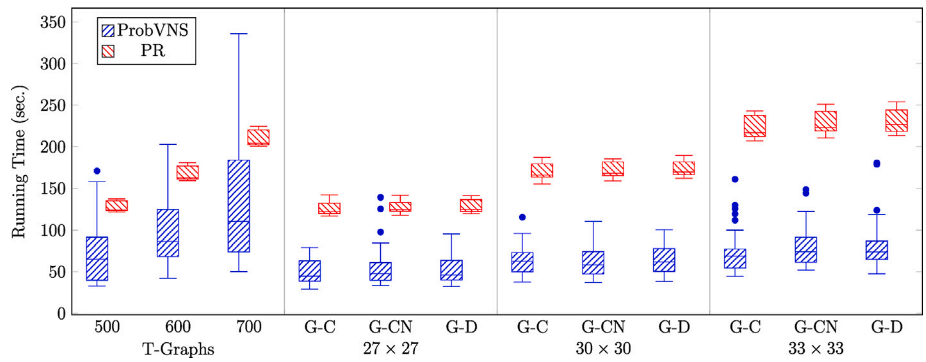

Fig. 6. Running times comparison.

these instances was 8.3%. Over all the 12 sets of instances tested, each set containing 30 instances, $MINF_{ProbVNS}$ was significantly lower than $MINF_{PR}$ in 6 sets, and $MOBJ_{ProbVNS}$ was significantly lower than $MOBJ_{PR}$ in 10 instance sets.

5.1.2. Running times and scalability

Fig. 6 illustrates a comparison of running times of ProbVNS and PR for different graph sizes and graph types. The average running time of ProbVNS over all graphs was 2.7 times lower than for PR. For T-graph types, ProbVNS is, on average, 2.12 times faster than PR. Over all G-graph types, ProbVNS is, on average, 2.90 times faster than PR. ProbVNS is, on average, 2.67 times faster than PR for 27 × 27 graphs, 2.90 times faster for 30 × 30 graphs, and 3.15 times faster for 33 × 33 graphs. As expected, the running times for G-graphs are lower than those for T-graphs because of the number of vertices and edges removed during the construction of these graphs. The type of G-graph did not seem to significantly impact the running times, indicating that the number of nodes is more decisive in the running times than the specific graph structure.

As Fig. 6 indicates, although ProbVNS is faster, it has a higher variance in running times than PR. We believe this is due to the shaking procedure that adds more randomization to the algorithm.

5.1.3. Comparison of ProbVNS with MIP solution

To further analyze the performance of ProbVNS, we compared the solutions generated with the ones obtained by implementing an MIP formulation in a commercial solver. In our experiments, we used the following MIP formulation of the DTDP:

$$
\begin{aligned}
&\min\ D \\
&\text{s.t. } D \geq d_{ij}(x_{im} + x_{jm} - 1), && \forall i, j \in V, m \in \{1, \dots, p\} \\
&\sum_{m=1}^{p} x_{im} = 1, && \forall i \in V \\
&\frac{1}{\mu^a} \sum_{i \in V} w_i^a x_{im} \geq (1 - \tau^a), && \forall a \in A, m \in \{1, \dots, p\} \\
&\frac{1}{\mu^a} \sum_{i \in V} w_i^a x_{im} \leq (1 + \tau^a), && \forall a \in A, m \in \{1, \dots, p\} \\
&x_{im} \in \{0, 1\}, && \forall i \in V, m \in \{1, \dots, p\} \\
&D \geq 0
\end{aligned}
$$

We solved this MIP with GUROBI 10.0 for the 10 instances of each type/size where $INF_{\text{ProbVNS}} = 0$. Fig. 7 presents the comparison of the diameters obtained by ProbVNS and PR with the diameters obtained by GUROBI after one hour execution time.

Within 1 h, Gurobi was able to find the optimal solution for only 1 instance out of 120. For all the other instances, the MIP gaps were greater than 45% after 1 h. At the same time, the minimal diameters obtained by ProbVNS in less than 6 min were, on average, 1.97 times smaller than the diameters obtained by GUROBI.

Table 2
Low and high ranges for node attributes.

	Low values	High values
demand	$U[15, 150]$	$U[150, 400]$
workload	$U[15, 30]$	$U[30, 100]$
nr. customers	$U[4, 8]$	$U[8, 20]$

Table 3
Sensitivity results grouped by graph type and size.

Graph	Graph	Objective		Infeasibility	
Type	Size	Mean	St.dev.	Mean	St.dev.
G-C	27 × 27	210.7	37.45	0.0015	0.0078
	30 × 30	236.7	35.81	0.0017	0.0069
	33 × 33	255.2	33.60	0.0016	0.0071
G-CN	27 × 27	201.5	23.72	0.0077	0.0188
	30 × 30	220.8	17.54	0.0029	0.0107
	33 × 33	248.9	16.54	0.0155	0.0319
G-D	27 × 27	206.0	20.30	0.0143	0.0356
	30 × 30	242.1	27.96	0.0226	0.0524
	33 × 33	285.2	22.16	0.0212	0.0570

5.2. Sensitivity analysis

In order to study the impact of the parameters on the solutions obtained by ProbVNS, we divided the values of each attribute into low and high (see Table 2) and varied the percentages of nodes with low values of the attributes (demand, workload, and number of customers). We obtained 27 instances, where for each attribute, 25%, 50%, and 90% of the nodes have low values and the remaining have high values.

The sensitivity experiments were performed on G-graphs. First, for each attribute, we randomly select the nodes that will have low values; the remaining nodes receive high values of that attribute. The values are chosen uniformly from the ranges in Table 2.

As we can see in the upper part of Fig. 8, the percentage of nodes with low/high values for attributes does not appear to impact the infeasibility measure of the solutions. Table 3 indicates that for each type of graph, the average infeasibility is below 0.023, with a maximum standard deviation of 0.057. This indicates that the performance of ProbVNS is robust in the change of the problem parameters.

The lower part of Fig. 8 indicates that the maximum diameter is not impacted by the change in parameters. If we look at the sensitivity results per graph type and size (see Fig. 9) we notice a larger average maximum diameter and a larger variance in the objective for the G-C and G-D graphs, where more nodes are removed from the central region of the graph. This structure leads to districts with larger diameters than the districts obtained for the G-CN graphs.

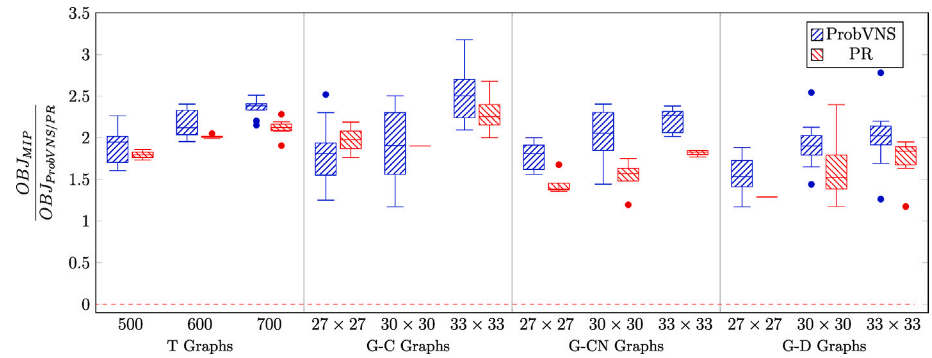

Fig. 7. Comparison of the minimal diameters obtained by PR and ProbVNS to the ones obtained by GUROBI solver in 1 h.

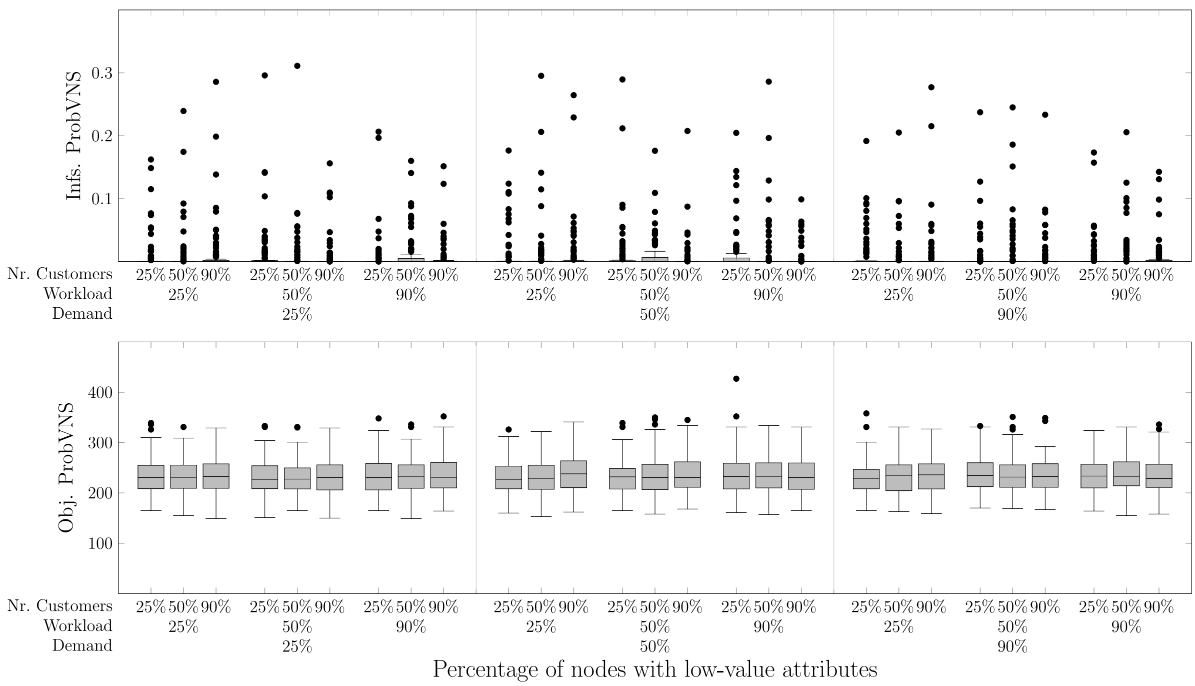

Fig. 8. Sensitivity of the ProbVNS solutions to the percentage of nodes with low/high attribute values.

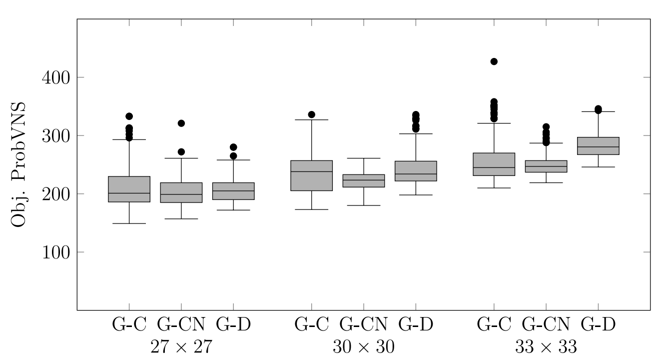

Fig. 9. Sensitivity results grouped by graph type and size.

6. Concluding remarks

In this paper, we focus on DTDP, a districting problem that often occurs in delivery operations, in which balancing constraints for several attributes (demand, workload, number of customers) are taken into account while minimizing the maximum diameter. For this problem, we propose a Probability-based VNS (ProbVNS) heuristic based on a merit function defined as a linear combination of the objective and a measure of infeasibility. The algorithm has two novel elements. First, the shaking procedure helps diversify the search region and reduces the infeasibility of the problem. Second, the direction of the search is changed by adjusting the penalty for violating the constraints, allowing the algorithm to prioritize either a decrease in objective or a decrease in infeasibility.

To assess the quality of the proposed algorithm, we compared the performance of ProbVNS to a GRASP with the Path-Relinking algorithm (PR) recently proposed by Ríos-Mercado and Escalante (2016). The comparison was performed on two sets of instances. The first one contains planar graphs based on Delaunay triangulation (T-graphs), similar to those in Ríos-Mercado and Escalante (2016). The second set of instances is new and contains graphs obtained by removing randomly nodes from specific regions (G-graphs). Our experiments indicate that finding feasible solutions for G-graphs was harder for both algorithms than for T-graphs. Therefore, we believe that the G-graphs complement the existing benchmark instances for districting problems.

On 90% of the tested instances, ProbVNS registered a lower infeasibility measure than PR. The average decrease in the objective value for these cases was 8.3%, with a maximum decrease of 51%. Sensitivity experiments indicate that ProbVNS is robust for different percentages of nodes with low/high attributes. On average, ProbVNS was 2.7 times faster than PR.

There are several possible ways to continue this research. First, one could focus on improving the performance of ProbVNS, both in quality of the solution as well as the running time. In particular, we believe that there is room for further improvement in the construction of the initial solution. Currently, the algorithm generates a set of initial solutions and chooses the best one, resulting in increased running time. Second, it would be interesting to see if the running time of the algorithm can be improved by excluding solutions that dominate each other in some of the attributes. Third, we recommend focusing on extensions of DTDP, in particular extensions that incorporate uncertainty. An important issue in the field of territory design is the non-stationarity of demand, number of locations, and travel time over the day. Algorithms capable to design districts that remain relatively unchanged when workload or travel times vary slightly are valuable to practitioners. This permits drivers to be familiar with their districts, and thus increase efficiency of deliveries.

Finally, notice that in practice, territory design problems are complex multi-criteria optimization problems. An interesting research direction would be to develop a VNS procedure for multi-objective optimization, in particular, for multi-objective TDPs.

CRediT authorship contribution statement

Ahmed Aly: Investigation, Software, Validation, Visualization, Writing – original draft. **Adriana F. Gabor:** Conceptualization, Methodology, Investigation, Supervision, Writing – original draft, Writing – review & editing. **Nenad Mladenovic:** Conceptualization, Funding acquisition, Methodology, Supervision. **Andrei Sleptchenko:** Data curation, Investigation, Software, Validation, Writing – review & editing.

Data availability

We have shared the link to the data generated in the paper. We will share the code upon request.

Acknowledgments

This research is supported by Khalifa University under Grant No. FSU-2020-19 and Award No. RC2 DSO.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.cor.2024.106756.

References

Aly, A., Gabor, A.F., Sleptchenko, A., 2023. Instances for delivery territory design problem. URL https://github.com/ahmed-o-aly/TerritoryDesign.

Brimberg, J., 1996. A variable neighborhood algorithm for solving the continuous location–allocation problem. Stud. Locat. Anal. 10, 1–12, URL https://cir.nii.ac.jp/crid/1573105973876395136.

Brimberg, J., Mladenović, N., Todosijević, R., Urošević, D., 2019. Solving the capacitated clustering problem with variable neighborhood search. Ann. Oper. Res. 272, 289–321. http://dx.doi.org/10.1007/s10479-017-2601-5.

Brimberg, J., Salhi, S., Todosijević, R., Urošević, D., 2023. Variable neighborhood search: The power of change and simplicity. Comput. Oper. Res. 155, http://dx.doi.org/10.1016/j.cor.2023.106221.

Caro, F., Shirabe, T., Guignard, M., Weintraub, A., 2004. School redistricting: Embedding gis tools with integer programming. J. Oper. Res. Soc. 55, 836–849. http://dx.doi.org/10.1057/palgrave.jors.2601729.

Erkut, E., Ülküsal, Y., Yeniçerioğlu, O., 1994. A comparison of p-dispersion heuristics. Comput. Oper. Res. 21, 1103–1113. http://dx.doi.org/10.1016/0305-0548(94)90041-8, URL https://www.sciencedirect.com/science/article/pii/0305054894900418.

Fleszar, K., Hindi, K., 2008. An effective VNS for the capacitated p-median problem. European J. Oper. Res. 191, 612–622. http://dx.doi.org/10.1016/j.ejor.2006.12.055.

Gonzalez, T.F., 1985. Clustering to minimize the maximum intercluster distance. Theor. Comput. Sci. 38, 293–306. http://dx.doi.org/10.1016/0304-3975(85)90224-5.

Hansen, P., Mladenović, N., Todosijević, R., Hanafi, S., 2017. Variable neighborhood search: basics and variants. EURO J. Comput. Optim. 5, 423–454. http://dx.doi.org/10.1007/s13675-016-0075-x.

Jarrah, A.I., Bard, J.F., 2012. Large-scale pickup and delivery work area design. Comput. Oper. Res. 39, 3102–3118. http://dx.doi.org/10.1016/j.cor.2012.03.014.

Mayorga, M.E., Bandara, D., McLay, L.A., 2013. Districting and dispatching policies for emergency medical service systems to improve patient survival. IIE Trans. Healthc. Syst. Eng. 3, 39–56. http://dx.doi.org/10.1080/19488300.2012.762437.

Mladenović, N., Hansen, P., 1997. Variable neighborhood search. Comput. Oper. Res. 24, 1097–1100. http://dx.doi.org/10.1016/S0305-0548(97)00031-2.

Mladenović, N., Labbé, M., Hansen, P., 2003. Solving the p-center problem with tabu search and variable neighborhood search. Networks 42, 48–64. http://dx.doi.org/10.1002/net.10081.

Mladenović N. Alkandari, A., Pei, J., Todosijević, R., Pardalos, P.M., 2019. Less is more approach: basic variable neighborhood search for the obnoxious p-median problem. Int. Trans. Oper. Res. 27, 480–493. http://dx.doi.org/10.1111/itor.12646.

Ricca, F., Scozzari, A., Simeone, B., 2013. Political districting: From classical models to recent approaches. Ann. Oper. Res. 204, 271–299. http://dx.doi.org/10.1007/s10479-012-1267-2.

Ríos-Mercado, R.Z. (Ed.), 2020. Optimal Districting and Territory Design. Springer International Publishing, http://dx.doi.org/10.1007/978-3-030-34312-5.

Ríos-Mercado, R.Z., Álvarez Socarrás, A.M., Castrillón, A., López-Locés, M.C., 2021. A location–allocation-improvement heuristic for districting with multiple-activity balancing constraints and p-median-based dispersion minimization. Comput. Oper. Res. 126, 105106. http://dx.doi.org/10.1016/j.cor.2020.105106.

Ríos-Mercado, R.Z., Escalante, H.J., 2016. GRASP with path relinking for commercial districting. Expert Syst. Appl. 44, 102–113. http://dx.doi.org/10.1016/j.eswa.2015.09.019.

Ríos-Mercado, R.Z., Fernández, E., 2009. A reactive GRASP for a commercial territory design problem with multiple balancing requirements. Comput. Oper. Res. 36, 755–776. http://dx.doi.org/10.1016/j.cor.2007.10.024.

Ríos-Mercado, R.Z., Salazar-Acosta, J.C., 2011. A GRASP with strategic oscillation for a commercial territory design problem with a routing budget constraint. In: Batyrshin, I., Sidorov, G. (Eds.), Advances in Soft Computing. MICAI 2011. In: Lecture Notes in Computer Science, vol. 7095, Springer, Berlin, Heidelberg, pp. 307–318. http://dx.doi.org/10.1007/978-3-642-25330-0_27.

Salazar-Aguilar, M.A., Ríos-Mercado, R.Z., Cabrera-Ríos, M., 2011a. New models for commercial territory design. Netw. Spat. Econ. 11, 487–507. http://dx.doi.org/10.1007/s11067-010-9151-6.

Salazar-Aguilar, M.A., Ríos-Mercado, R.Z., González-Velarde, J.L., 2011b. A bi-objective programming model for designing compact and balanced territories in commercial districting. Transp. Res. C 19, 885–895. http://dx.doi.org/10.1016/j.trc.2010.09.011.

Salazar-Aguilar, M.A., Ríos-Mercado, R.Z., González-Velarde, J.L., 2013. GRASP strategies for a bi-objective commercial territory design problem. J. Heuristics 19, 179–200. http://dx.doi.org/10.1007/s10732-011-9160-8.

Salazar-Aguilar, M.A., Ríos-Mercado, R.Z., González-Velarde, J.L., Molina, J., 2012. Multiobjective scatter search for a commercial territory design problem. Ann. Oper. Res. 199, 343–360. http://dx.doi.org/10.1007/s10479-011-1045-6.

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. http://dx.doi.org/10.1016/j.omega.2022.102687.

Sandoval, M.G., Díaz, J.A., Ríos-Mercado, R.Z., 2020. An improved exact algorithm for a territory design problem with p-center-based dispersion minimization. Expert Syst. Appl. 146, 113150. http://dx.doi.org/10.1016/j.eswa.2019.113150, URL https://www.sciencedirect.com/science/article/pii/S095741741930867X.

Sudtachat, K., Mayorga, M.E., Mclay, L.A., 2016. A nested-compliance table policy for emergency medical service systems under relocation. Omega 58, 154–168. http://dx.doi.org/10.1016/j.omega.2015.06.001.

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. http://dx.doi.org/10.1016/j.omega.2021.102442.

Zoltners, A.A., Sinha, P., 2005. The 2004 ISMS practice prize winner —- sales territory design: Thirty years of modeling and implementation. Mark. Sci. 24, 313–331. http://dx.doi.org/10.1287/mksc.1050.0133.