Contents lists available at ScienceDirect

Omega

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

Solutions for districting problems with chance-constrained balancing requirements☆

Antonio Diglio[a,*], Juanjo Peiró[b], Carmela Piccolo[a], Francisco Saldanha-da-Gama[c,d]

[a] Università degli Studi di Napoli Federico II, Department of Industrial Engineering (DII), Piazzale Tecchio, 80 - 80125 Naples, Italy
[b] Departament d'Estadística i Investigació Operativa, Facultat de Ciències Matemàtiques, Universitat de València, Spain
[c] Departamento de Estatística e Investigação Operacional, Faculdade de Ciências da Universidade de Lisboa, Lisboa 1749-016, Portugal
[d] Centro de Matemática, Aplicações Fundamentais e Investigação Operacional, Faculdade de Ciências da Universidade de Lisboa, Lisboa 1749-016, Portugal

ARTICLE INFO

Article history:
Received 1 August 2020
Accepted 16 February 2021
Available online 18 February 2021

Keywords:
Stochastic programming
Districting
Contiguity
Stochastic demand
Heuristics

ABSTRACT

In this paper, a districting problem with stochastic demands is investigated. The goal is to divide a geographic area into p contiguous districts such that, with some given probability, the districts are balanced with respect to some given lower and upper thresholds. The problem is cast as a p-median problem with contiguity constraints that is further enhanced with chance-constrained balancing requirements. The total assignment cost of the territorial units to the representatives of the corresponding districts is used as a surrogate compactness measure to be optimized. Due to the tantalizing purpose of deriving a deterministic equivalent for the problem, a two-phase heuristic is developed. In the first phase, the chance-constraints are ignored and a feasible solution is constructed for the relaxed problem; in the second phase, the solution is corrected if it does not meet the chance-constraints. In this case, a simulation procedure is proposed for estimating the probability of a given solution to yield a balanced districting. That procedure also provides information for guiding the changes to make in the solution. The results of a series of computational tests performed are discussed based upon a set of testbed instances randomly generated. Different families of probability distributions for the demands are also investigated, namely: Uniform, Log-normal, Exponential, and Poisson.

1. Introduction

A districting problem consists of grouping a set of territorial units (TUs) into a set of districts accounting for some aspects namely, balancing, contiguity, and compactness. Districts are said to be balanced when they are of equitable size in terms of dimension, which is usually measured by the total demand or service request within a district. Contiguity is a topological property that holds when it is not necessary to cross other districts for traveling between TUs in the same district. Compactness is also a topological property and indicates that the TUs in a district are somehow close to the center of the district.

The role of OR in districting problems keeps attracting much attention from the scientific c[1]ommunity due to the broad scope of potential applications, which include sales territory design, political districting, and waste collection, to mention a few. The reader can refer to Kalcsics and Ríos-Mercado [16] and to the references therein for a recent overview on these and other aspects in the topic. A recent volume highlighting current trends and applications in districting is due to Ríos-Mercado [30]. Some recent works that are not quoted in the above references include Bender et al. [2], Dugošija et al. [8], Faughi et al. [11], Kong et al. [20], Liu et al. [23], Restrepo et al. [28], Sandoval et al. [35], and Yaník et al. [39]. In fact, we currently observe new directions being pursued such as the need to capture time-dependent parameters and decisions. We also see existing research directions being deepened such as the need to consider several objective functions in some districting applications. It is also worth noticing that some applications are still challenging and call for more research to be done despite all the developments already achieved. This is the case, for instance, with political districting problems.

☆ Area: Production Management, Scheduling and Logistics. This manuscript was processed by Associate Editor Yagiura. Research Paper: Production Management, Scheduling and Logistics.

* Corresponding author.
E-mail addresses: antonio.diglio@unina.it (A. Diglio), juanjo.peiro@uv.es (J. Peiró), carmela.piccolo@unina.it (C. Piccolo), fsgama@ciencias.ulisboa.pt (F. Saldanha-da-Gama).

[1] $\mathrm{RSD}[d] = \frac{\sqrt{\mathrm{Var}[d]}}{\mathrm{E}[d]}$

https://doi.org/10.1016/j.omega.2021.102430
0305-0483/

In the current paper, we investigate another recent trend in districting problems which has to do with the inclusion of uncertainty in models and methods supporting decision-making processes. We investigate the use of chance-constraints for dealing with balancing requirements when demand is stochastic. We assume that such demand can be described by a joint cumulative distribution function known beforehand (e.g., using historical data). The goal is to devise a districting plan ensuring that the districts are balanced with some given probability.

Like in many other situations, the motivation for considering chance-constraints in districting problems stems from the fact that, under uncertainty, the extreme scenarios (e.g., too high or too low demand) may significantly influence the solution made, which should not be the case if such scenarios occur with a very small probability. Chance-constraints are a way to overcome this drawback since they impose a minimum acceptable service level: a solution is sought such that, with some (high) probability, it turns out to be feasible when uncertainty is disclosed.

Another aspect of relevance in our work has to do with contiguity, which we assume as a hard constraint. This has not been always the case as we observe in the review paper by Ricca et al. [29] and more recently in Diglio et al. [7] and Yaník et al. [39]. Nevertheless, hard contiguity constraints have been considered explicitly in optimization models by several authors such as Salazar-Aguilar et al. [34], Shirabe [36], 37], and Xie and Ouyang [38]. As pointed out by Kalcsics and Ríos-Mercado [16], contiguity is easier to ensure if heuristics are adopted to solve the problem and we can see this in some literature such as Butsch et al. [5], Elizondo-Amaya et al. [9], Ríos-Mercado and Fernández [32], Ríos-Mercado and López-Pérez [33], and Ríos-Mercado et al. [31]. Our work also explores this fact.

Stochastic districting is a more recent research trend as we conclude in the overview provided by Kalcsics and Ríos-Mercado [16]. Most of the existing work has been developed in the context of vehicle routing with stochastic demand. In this case, the customers are grouped (districts are created) before demand is known. When demand becomes known, the routes are defined and the service is provided. For this research direction the reader can refer to the seminal work by Haugland et al. [13] as well as to the works by Carlsson and Delage [6], Lei et al. [21], Lei et al. [22], and Mayorga et al. [25].

Elizondo-Amaya et al. [9] investigate a stochastic disticting problem. Uncertainty is associated with the demand level in each TU.The authors develop a Tabu Search heuristic for the problem that consists of partitioning the TUs into a set of p territories so that the expected value of the maximum demand observed across the different territories is maximized. Balancing and connectivity constraints are also considered. More recently, Diglio et al. [7] have cast a districting problem under uncertain demand as a two-stage stochastic programming problem: in the first stage the districts are defined; in the second stage (after the demand is disclosed) several recourse actions are investigated for ensuring that the balancing constraints are satisfied. A neutral attitude of the decision-maker towards risk is assumed. It should be noted that in that work the balancing constraints are looked at as hard constraints in the sense that, independently from the initial districting defined, the balancing constraints must be satisfied in the second stage.

The contribution of the current paper to the literature is the following:

i) We propose a chance-constrained optimization model for districting.
ii) A heuristic procedure is developed for the problem, which combines a local search algorithm with simulation. The latter is needed for estimating the probability of satisfying the probabilistic constraints.
iii) Different families of demand distributions are empirically assessed, namely: Uniform, Log-normal, Poisson, and Exponential.

The remainder of the work is organized as follows: in Section 2 we discuss the modeling aspects related to the problem we are investigating. In Sections 3 and 4 we present the heuristic we developed for finding feasible solutions to the problem. In Section 5 we report on the results of the computational tests performed to assess our methodological contributions. The paper ends with an overview of the work done and some directions for further research.

2. Optimization models

We consider a finite set I of TUs that we want to partition into p districts. Each district will have a TU representing it. Hence, the inclusion of a TU in a district is established by its assignment to the selected representative of the district. A given connected graph is assumed for representing the underlying geographical region, i.e., there is at least one chain connecting every pair of TUs. Every TU $i \in I$ has a demand d_i, which is assumed to be a random variable. The joint CDF of the random vector $[d_i]_{i\in I}$ is assumed to be known (e.g., estimated using historical data).

As customary in districting problems, districts are required to be balanced. This is ensured by considering a reference value and a maximum deviation allowed w.r.t. it. In a deterministic setting, the reference value commonly adopted is $(1/p)\sum_{i\in I} d_i$, which corresponds to a uniform distribution of the overall demand across the p districts. In our case, we keep assuming the above reference value although it becomes itself a random variable since expressed as a linear combination of the random variables we are working with. Furthermore, due to uncertainty we cannot ensure in advance that a districting solution renders the districts balanced. Therefore we cast this feature using a set of probabilistic constraints, i.e., we assume an exogenous value, say γ, for the probability that a solution turns out to be balanced.

As usually done in the literature, compactness—a desirable feature for the districts—is captured using a surrogate compactness measure, which is given by the total assignment cost of the TUs to the districts. Since every district has a TU representing it, a "natural" assignment cost of a TU to a district is a function of the distance between the TU and the representative. Hence, by minimizing the total assignment cost we seek to obtain districts such that the TUs are close to the corresponding representative, i.e., in line with an intuitive notion of compactness ([16]).

The parameters defining our problem can be summarized as follows:

d_i,	demand of TU $i \in I$; $D = \sum_{i\in I} d_i$, is the total demand;
c_{ij},	cost for assigning TU i to TU j ($i, j \in I$);
p,	number of districts to consider;
α,	maximum desirable deviation of the demand within a district w.r.t. the reference value adopted.
γ,	probability that a districting solution turns out to be balanced.

The complete problem can be formulated mathematically using the following decision variables introduced in the seminal paper by Hess et al. [14]:

$$x_{ij} = \begin{cases} 1, & \text{if TU } i \text{ is assigned to TU } j; \\ 0, & \text{otherwise.} \end{cases} \qquad (i, j \in I)$$

$x_{jj} = 1$ indicates that TU j is assigned to itself, i.e., it is selected as the representative of its district.

Using these variables we can formulate the problem as follows:

$$\text{minimize} \quad \sum_{i\in I}\sum_{j\in I} c_{ij}x_{ij}, \tag{1}$$

$$\text{subject to} \quad \sum_{j\in I} x_{ij} = 1, \qquad i \in I, \tag{2}$$

$$\sum_{j\in I} x_{jj} = p, \tag{3}$$

$$x_{ij} \le x_{jj}, i, j, \in I, \tag{4}$$

$$\mathbb{P}\left[(1-\alpha)\frac{D}{p}x_{jj} \le \sum_{i\in I} d_i x_{ij} \le (1+\alpha)\frac{D}{p},\ j \in I \right] \ge \gamma, \tag{5}$$

$$x_{ij} \in \{0,1\}, i, j \in I. \tag{6}$$

The objective function (1) quantifies the total assignment cost to be minimized. Constraints (2) ensure that each TU is assigned to exactly one district whereas constraints (3) guarantee that exactly p districts will be designed. Inequalities (4) state that we can only assign each TU to a representative of a district. Inequalities (5) are the chance-constraints for the balancing requirements imposing that solutions must be balanced with probability at least equal to γ. Finally (6) define the domain of the decision variables. Note that problem (1)–(6) is $\mathcal{NP}$-hard since it contains the discrete p-median problem as a particular case (which is well-known to be $\mathcal{NP}$-hard).

In districting problems, the costs c_{ij} are typically related with the distances ([16]). Denoting by ℓ_{ij} the distance between i and j $(i, j \in I)$, a common cost to consider is $c_{ij} = \ell_{ij}$ or $c_{ij} = \ell_{ij}^2$. This turns the above objective function into a so-called compactness measure known as moment of inertia (14). The reader may refer to Kalcsics and Ríos-Mercado [16] for variants of distance-based compactness measures. In particular, in that book chapter, we also observe cost structures that consider the demands as weighting factors. Finally, we note that euclidean distances are often considered (3; 1).

Assuming that demands are independent, then we can go farther in terms of the above model writing (5) as

$$\prod_{j\in I}\mathbb{P}\left[(1-\alpha)\frac{D}{p}x_{jj} \le \sum_{i\in I} d_i x_{ij} \le (1+\alpha)\frac{D}{p} \right] \ge \gamma. \tag{7}$$

Unfortunately, such expression does not help when it comes to deriving a deterministic equivalent for the chance-constraints. This limitation motivates the procedure we are proposing in the following sections for deriving a feasible solution to the problem.

2.1. Embedding contiguity constraints

The above model does not ensure one desirable feature of a solution: contiguity (i.e., no enclaves). One way to ensure this property consists of using a set of network flow constraints (see 37). The idea is to look at each TU as generating one unit of supply and the districts' representatives acting as sinks. In this case, a district is contiguous if we can find a feasible flow from its TUs to its representative using only TUs in the district. Let us assume that the problem has an underlying graph $G = (I, A)$ with vertices representing the TUs and A containing all the direct links between TUs (a link exists for every pair of adjacent TUs). Contiguity constraints can be included in the above model by considering one additional set of decision variables:

y_{mij} = flow of district j sent from m to i, with $i, j, m \in I$, and $(m, i) \in A$.

The following constraints that are inspired by those proposed by [37] can now be added to model (1)–(6):

$$\sum_{m\in I\,|\,(m,i)\in A} y_{mij} \le (|I|-p)x_{ij}, i, j \in I, \tag{8}$$

$$\sum_{m\in I\,|\,(i,m)\in A} y_{imj} - \sum_{m\in I\,|\,(m,i)\in A} y_{mij} = x_{ij}, \quad i, j, \in I,\ i \neq j. \tag{9}$$

Constraints (8) ensure that flow corresponding to district j can be sent to a node i only if that node is assigned to that district. The right-hand side results from the fact that in each district we have at most $|I| - p$ TUs that are not representatives. Constraints (9) guarantee that if a node is assigned to some district but is not the representative of the district, then it generates exactly one unit of flow in that district.

It should be pointed out that other ways of imposing contiguity have been proposed in the literature, as in [34]. In this case, connectivity constraints are modeled in line with the sub-tour elimination constraints for the Traveling Salesman Problem. These constraints are exponential in number but in [34] they are embedded in a cut generation approach (i.e., only those needed are generated). One advantage of such representation of contiguity is that no additional decision variables are necessary (like the y-variables above considered). Nevertheless, in our work, we adopt (8) and (9) since at some point we consider the resolution of a deterministic approximation model using a general-purpose solver. Therefore, it is convenient to use the polynomial number of contiguity constraints conveyed by the y-representation. Next, we provide the details of this approximate model.

2.2. An approximate deterministic model

When considering a Stochastic Programming model one should investigate whether a simplified model can provide a reasonable approximate solution to the problem thus avoiding solving a more complicated model. A typical approximation consists of reducing the future to one scenario. Often, one simple way for accomplishing this consists of replacing each random variable by its expectation. This can also be done in our case.

For every TU $i \in I$, demand d_i is a random variable; hence, the total demand generated in the system, D, is also a random variable. Let $\mu_i = \mathbb{E}[d_i]$, $i \in I$. We can reduce the future to the single scenario in which TU i has a demand exactly equal to μ_i and the expected total demand is $E[D] = \sum_{i\in I}\mu_i$. In this case (single scenario) it does not make sense to consider chance-constraints and we can re-write the balancing constraints as traditionally done in a deterministic setting. Considering that the reference value for the demand in each district is now given by $E[D]/p$, the balancing constraints can be re-written as:

$$(1-\alpha)\frac{E[D]}{p}x_{jj} \le \sum_{i\in I} d_i x_{ij} \le (1+\alpha)\frac{E[D]}{p}x_{jj}, \quad j \in I. \tag{10}$$

The approximate model we are proposing consists of minimizing (1), subject to (2), (3), (6), (8), (9), and (10). Now we do not need to include (4) since these constraints are implied by (10). In Section 5 we will discuss the results of the computational tests performed using this model to understand the relevance of the probabilistic model we are proposing. Of course, there might be more refined ways to derive better deterministic approximations to our model (see, for instance, 17,18). However, these are out of the scope of the present work since our aim, here, is just to identify a simple reference point to assess the performance of the proposed heuristic, as we discuss later.

2.3. In search for an approximate algorithm to the problem

In the next section, we introduce an algorithm for the stochastic districting problem we are studying. This is a procedure that integrates simulation and a metaheuristic as a way to obtain approximate solutions to a stochastic optimization problem. In the literature, such type of algorithm has been called a *simheuristic* (see [15]). Our procedure has two phases: (i) Building an initial districting solution ignoring the chance-constraints (Section 3); (ii) Estimating the probability stated in constraints (5) and changing the districts if the solution violates these constraints (Section 4).

3. Designing an initial districting

In this section, we focus on finding a feasible solution to the districting problem based solely upon the costs. Thus, for the moment we ignore the Chance-constraints (5). To help ensuring contiguity we assume that after collecting the data for a particular instance we have built a symmetric adjacency matrix for the TUs: for each pair of TUs the corresponding entry indicates whether (=1) or not (=0) the TUs are adjacent.

Ignoring Constraints (5), our problem reduces to a p-median problem with contiguity constraints. Therefore, we can think of taking advantage of existing knowledge on the p-median problem to efficiently find a feasible solution when contiguity constraints are involved. This motivated the algorithm we propose, which has two components: (i) a feasible solution to the underlying uncapacitated p-median problem is found; (ii) that solution is checked in terms contiguity and if the latter does not hold then the solution is corrected.

Regarding the above component (i), we note that the literature is prone to approximate algorithms for the p-median problem (see, e.g., [24]). Every such algorithm can be used here. Nevertheless, other possibilities could be considered. An alternative could consist of straightforward improvements over a purely random solution. This can be motivated by the fact that for the moment we are ignoring contiguity constraints and thus it may not compensate to put too much effort into the p-median solution. Finally, the third option could consist of selecting the p representatives by solving a p-dispersion problem and then assigning the remaining nodes to those centroids based upon distances to them.

In the scope of this paper, we have tried the three alternatives just described to obtain a starting p-median solution. We considered the heuristic method proposed by Resende and Werneck [27] in the first alternative, and two constructive heuristics presented by Erkut et al. [10] for the third alternative. A series of preliminary computational tests were performed to compare the three alternatives in the context of our work and the results showed that the first alternative clearly outperforms the other. For this reason, we adopted it.

After obtaining a feasible p-median solution we need to check the contiguity constraints and correct the solution if necessary. With this purpose, we make use of the adjacency matrix for the TUs and compute for every district the connected component that includes its representative (a procedure of $\mathcal{O}(|I|^2)$ for the worst case). Then we observe the nodes that are not in the connected component for the district they belong to. These are nodes not connected to their representative thus contributing to disrupting the contiguity of the solution.

Let $\overline{C}$ denote the set of TUs (if some exist) that are not contiguous to the corresponding representative, i.e., every path in the underlying graph between the TU and its representative crosses other district(s). Furthermore, for $i \in \overline{C}$, let C_i denote the set containing every TU adjacent to i that is either a representative or is contiguous to its own representative. With this notation, a result follows.

Lemma 1. *If the underlying graph is connected and $\overline{C} \neq \emptyset$, then there exists at least one $i \in \overline{C}$ such that $C_i \neq \emptyset$.*

Proof. Suppose that $C_i = \emptyset$, $\forall i \in \overline{C}$. Consider an arbitrary $i' \in \overline{C}$ and an arbitrary representative j'. Since our graph is connected, there is at least one chain connecting i' and j'. Let $i', \imath_1, \ldots, \imath_L, j'$ be a path we can use to go from i' to j'. This path has of course a finite number of nodes. Consider node $\imath_L$. This node is contiguous to its representative (j' or other), otherwise we would have $\imath_L \in \overline{C}$ and $C_{\imath_L} \neq \emptyset$ ($\imath_L$ has an adjacent node (j') that is contiguous to its representative—itself). Consequently, $\imath_{L-1}$ is also contiguous to its representative, otherwise we would have $\imath_{L-1} \in \overline{C}$ and $C_{\imath_{L-1}} \neq \emptyset$ ($\imath_{L-1}$ is connected to $\imath_L$ which is contiguous to its representative). Proceeding this reasoning backward in the path we would conclude that $\imath_1$ is contiguous to its representative, otherwise we would have $\imath_1 \in \overline{C}$ and $C_{\imath_1} \neq \emptyset$ ($\imath_1$ is connected to $\imath_2$). Finally, we conclude that $C_{i'} \neq \emptyset$ since i' is connected to a node ($\imath_1$) that is contiguous to its representative. This is an absurd, which results from the assumption that $C_i = \emptyset$ $\forall i \in \overline{C}$ that, it turn, implies that all nodes $\imath_1, \ldots, \imath_L$ are contiguous to the corresponding representatives. □

Based upon the previous Lemma, in case $\overline{C} \neq \emptyset$ after a feasible solution is obtained for the underlying uncapacitated p-median, then contiguity can be ensured by applying Algorithm 1. In this algorithm, for a TU i, the representative of its district is denoted by j_i.

Algorithm 1 Ensure contiguity ($\overline{C} \neq \emptyset$).

```
1:  while $\overline{C} \neq \emptyset$ do
2:      for $i \in \overline{C}$ do
3:          Build $C_i$   // TUs adjacent to $i$ that are contiguous to their representative.
4:          if $C_i \neq \emptyset$ then
5:              $\ell^* \leftarrow \arg\min_{\ell \in C_i}\{c_{ij_\ell}\}$;
6:              $x_{ij_i} \leftarrow 0$;   $x_{ij_{\ell^*}} \leftarrow 1$;   // $i$ is reassigned to district (representative) $j_{\ell^*}$
7:              $\overline{C} \leftarrow \overline{C} \setminus \{i\}$;
8:          end if
9:      end for
10: end while
```

The reasoning underlying Algorithm 1 is twofold:

- √ Take a TU $i \in \overline{C}$ and build the list, say C_i, with all TUs ℓ adjacent to it, that are either a representative of a district or that are connected to the representative of the district they belong to.
- √ If $C_i \neq \emptyset$, then assign i to the district represented in that set which induces the resulting solution with the lowest cost. This makes the solution "more contiguous" since node i is being assigned to a district adjacent to it.

Remark 1. It is worth noticing the following two observations:

a) In Algorithm 1, when considering a node i in line 2, it may happen that the corresponding set C_i is empty. This means that currently there is no TU adjacent to i that is a representative or that is connected to its representative. In such a situation we choose another TU in $\overline{C}$. By Lemma 1, there is at least one node in $\overline{C}$ whose contiguity to a representative is ensured. Hence, sooner or later all $i \in \overline{C}$ will be reassigned (in a contiguous way) and, thus, the algorithm terminates in a finite number of steps with a contiguous solution.

b) Algorithm 1 always terminates due to Lemma 1 and the fact that we are assuming a connected underlying graph.

After the execution of the heuristic by [27], and possibly making use of Algorithm 1, a feasible solution satisfying constraints (2), (3), (4), (6), (8) and (9) is available. Despite it may violate the probabilistic constraints, its cost can then be easily computed according to objective function (1) since for such computation we do not use stochastic data.

4. Ensuring the satisfaction of the probabilistic constraints

Unfortunately, we are not aware of any mechanism that allows us to write a compact set of constraints defining a deterministic equivalent for (5). Note that we are assuming a general CDF for the demand vector. For this reason, we develop a procedure making use of a simulation algorithm for estimating the probability stated in the chance constraints. If our estimate for such probability indicates that they are violated, then we change the solution.

Algorithm 2 formalizes this procedure. Given an incumbent solution, in line 2 we simulate the probability. If the obtained estimate is above desired threshold (γ), then we stop since the solution is feasible (lines 3 and 4). Otherwise, we change the solution (line 6). This change is guided by information gathered during the simulation procedure, namely by identifying the districts that more often contribute to the violation of the chance constraints, either by surplus or by shortage.

Algorithm 2 Ensuring feasibility w.r.t. the probabilistic constraints.

```
1: Input: $\hat{x}$ and the time limit   // $\hat{x}$ is the solution resulting from executing the heuristic by [27].
2: SIMULATE($\hat{x}, \hat{\pi}, \beta^-, \beta^+$,PRECISION,$N_{\min}$)   // Estimate $\hat{\pi}$, the probability stated in Const. (5).
3: if $\hat{\pi} \geq \gamma$ or the time limit has been attained then
4:     Stop.
5: else
6:     CORRECTSOLUTION($\hat{x}, \beta^-, \beta^+$)   // Change the solution.
7:     Return to 2:
8: end if
```

By repeating the loop SIMULATE( )-CORRECTSOLUTION( ) we seek to getting a feasible solution to our problem. However, we do not know beforehand whether Algorithm 2 converges. In fact, the value of γ may be too binding without us knowing that. For this reason, we also consider a time limit as a termination criterion.

4.1. Estimating the probability stated in the chance constraints

Algorithm 3 allows us to obtain an estimate of the probability stated in (5) for the current solution $\hat{x}$. The idea is to generate a vector of demands according to the underlying probability function and then check whether $\hat{x}$ is balanced (i.e., the demand assigned to each district is within the lower and upper thresholds). If so, we count this trial as a success. By performing a certain number, say N, of trials we count a resulting number, say T, of successes. The probability stated in Constraints 5 is estimated by the ratio $\hat{\pi} = T/N$ (line 28).

Other important details about Algorithm 3 are the following:

- There is a minimum ($N_{\min}$) and a maximum ($N_{\max}$) number of iterations (trials) that are performed, thus the complexity of the algorithm for its worst case is $\mathcal{O}(N_{\max}\ p|I|)$.
- In every trial we update the ratio T/N and compare it with the ratio after the previous trial. If the difference is below some given precision then we accept that the (absolute) ratio has stabilized and provides a good estimate of the probability stated in (5).
- For every TU j that is a representative in solution $\hat{x}$, we compute the proportion of trials in which the demand of the corresponding district failed to satisfy the balancing constraints by shortage (β_j^-) and by surplus (β_j^+). A large proportion means that very often the district contributes to a non-balanced solution. This is an indication that often the district has either too less or too much demand assigned to it, which is a very useful hint for changing the solution, as we see next.

Algorithm 3 Simulate($\hat{x}, \hat{\pi}, \beta^-, \beta^+$,precision,$N_{\min}$)—Estimating π.

```
1: $N \leftarrow 0$; $T \leftarrow 0$; AVG$\leftarrow 0.001$
2: for $j \in I$ such that $x_{jj} = 1$ do
3:     $b_j^- \leftarrow 0$; $b_j^+ \leftarrow 0$
4: end for
5: repeat
6:     $N \leftarrow N + 1$   // Number of simulations so far.
7:     AVG_OLD $\leftarrow$ AVG
8:     GENERATE($\tilde{d}$) // Vector of demands generated according to the CDF of $d$.
9:     $\tilde{\mu} \leftarrow \frac{1}{p} \sum_{i=1}^{|I|} \tilde{d}_i$
10:    $\ell \leftarrow 0$
11:    for $j \in I$ such that $x_{jj} = 1$ do
12:        $\tilde{D}_j \leftarrow \sum_{i \in I} \tilde{d}_i x_{ij}$
13:        if $\tilde{D}_j < (1 - \alpha)\tilde{\mu}$ then
14:            $b_j^- \leftarrow b_j^- + 1$
15:        else
16:            if $\tilde{D}_j > (1 + \alpha)\tilde{\mu}$ then
17:                $b_j^+ \leftarrow b_j^+ + 1$
18:            else
19:                $\ell \leftarrow \ell + 1$
20:            end if
21:        end if
22:    end for
23:    if $\ell = p$ (All districts are balanced) then
24:        $T \leftarrow T + 1$
25:    end if
26:    AVG= $\frac{T}{N}$
27: until $\left(\frac{|\text{AVG} - \text{AVG\_OLD}|}{\text{AVG\_OLD}} < \text{PRECISION}\right.$ and $\left.(N \geq N_{\min})\right)$ or $(N \geq N_{\max})$
28: $\hat{\pi} \leftarrow \frac{T}{N}$
29: for $j \in I$ such that $x_{jj} = 1$ do
30:     $\beta_j^- \leftarrow \frac{b_j^-}{N}$; $\beta_j^+ \leftarrow \frac{b_j^+}{N}$ // Proportions of under(over)-balancing.
31: end for
```

4.2. Changing the solution

After applying Algorithm 3 we may conclude that the current solution $\hat{x}$ does not satisfy the probabilistic constraints. This may be the case because either (i) there is no feasible solution to the problem because the probability γ is too binding, or (ii) because some districts are often under- or over-balanced. The second case can be overcome by correcting the solution in such a way that a district that is often under-balanced receives some TUs and a district that is often over-balanced gives away some TUs.

A way to check whether a district is "too" often under- or over-balanced consists of looking at the values β_j^- and β_j^+ (for $j \in I$: $\hat{x}_{jj} = 1$). The former indicates under-balancing whereas the latter indicates over-balancing. These values are obtained "for free" when applying Algorithm 3. In Algorithm 4 we formalize the procedure we propose for correcting a solution. This algorithm receives the current solution as an input and returns a different one in which the under- or over-balancing of one district is improved.

In lines 1 and 2 of Algorithm 4 we sort the districts non-increasingly according to the observed frequency in which they

Algorithm 4 correctSolution($\hat{x}, \beta^-, \beta^+$).

```
1:  Sort the indices $j \in I : \hat{x}_{jj} = 1$ non-increasingly according to the values of $\beta_j^+$ to obtain
        $[j_1^+, j_2^+, \dots, j_p^+]$
2:  Sort the indices $j \in I : \hat{x}_{jj} = 1$ non-increasingly according to the values of $\beta_j^-$ to obtain
        $[j_1^-, j_2^-, \dots, j_p^-]$
3:  if $\beta^-_{j_1^-} > \beta^+_{j_1^+}$ then
4:      $j' \leftarrow j_1^-$
5:      $k \leftarrow 0$
6:      while $k < k_{\max}$ do
7:          if FINDTOINSERT($\hat{x}, i^*, j', j'', \min\Delta$) = 1 then
8:              $\hat{x}_{i^*j'} \leftarrow 1$;   $\hat{x}_{i^*j''} \leftarrow 0$
9:              obj($\hat{x}$) ← obj($\hat{x}$) + $\min\Delta$
10:             $k \leftarrow k+1$
11:         else
12:             $k \leftarrow k_{\max}$
13:         end if
14:     end while
15: else
16:     $j' \leftarrow j_1^+$
17:     $k \leftarrow 0$
18:     while $k < k_{\max}$ do
19:         if FINDTOREMOVE($\hat{x}, i^*, j', j'', \min\Delta$) = 1 then
20:             $\hat{x}_{i^*j''} \leftarrow 1$;   $\hat{x}_{i^*j'} \leftarrow 0$
21:             obj($\hat{x}$) ← obj($\hat{x}$) + $\min\Delta$
22:             $k \leftarrow k+1$
23:         else
24:             $k \leftarrow k_{\max}$
25:         end if
26:     end while
27: end if
```

are over-balanced (line 1) or under-balanced (line 2). We then compare the highest value of both sequences. The “winner” district is denoted by j'. If the “winner” is an under-balanced district, then it should receive TUs from other districts. This is done in lines 3–14. We set a maximum number, $k_{\max}$, of TUs to be inserted into the district (this is a constant parameter considered for all calls of this algorithm). Furthermore, we use a routine—FINDTOINSERT($\hat{x}, i^*, j', j'', \min\Delta$)—to find a TU i^* and a district j'' such that TU i^* will be removed from district j'' and inserted into district j' (line 7). This routine is specified in Algorithm 5. The variation in the solution cost is represented by $\min\Delta$ (line 9). This is due to the fact that the selection of a TU to be moved across districts is based upon the variation in the solution cost. A minimum variation is desired (keeping feasibility).

On the other hand, if the “winner” districting, in terms of violation of the balancing requirements, is an over-balanced district, then it should give away TUs to other districts. This is done in lines 15–26. Again, we consider the maximum number $k_{\max}$ of TUs to be removed from the district (this is the same constant above mentioned). Furthermore, we use another routine—FINDTOREMOVE($\hat{x}, i^*, j', j'', \min\Delta$)—to find a TU i^* and a district j'' such that TU i^* will be removed from district j' and inserted into district j'' (line 19). This routine is detailed in Algorithm 6. The variation in the solution cost is again represented by $\min\Delta$ (line 21) for the reasons already mentioned.

The routine FINDTOINSERT($\hat{x}, i^*, j', j'', \min\Delta$), detailed in Algorithm 5, is called whenever we want to move extra TUs into a district that is frequently under-balanced. In this case, the best is to search for TUs in districts that are frequently over-balanced. Hence, we search the over-balanced districts according to the sequence $[j_1^+, j_2^+, \dots, j_p^+]$, i.e., we start with the districts that are over-loaded more frequently. By selecting TUs in this way, we have the possibility of “correcting” the solution in two ways: inserting TUs in an under-balanced district using TUs removed from over-balanced districts. In all this procedure, disrupting contiguity is not allowed, i.e., we only accept moves that do not disrupt contiguity in the involved pairs of districts.

Algorithm 5 FINDTOINSERT($\hat{x}, i^*, j', j'', \min\Delta$).

```
$\ell \leftarrow 1$
while $\ell \le p$ do
    $\min\Delta \leftarrow \infty$
    for $i \in I : \hat{x}_{ij_\ell^+} = 1$ and $j_\ell^+ \neq j'$ do
        if setting $\hat{x}_{ij'} = 1$ does not break contiguity then
            $\delta \leftarrow c_{ij'} - c_{ij_\ell^+}$
            if $\delta < \min\Delta$ then
                $\min\Delta \leftarrow \delta$
                $i^* \leftarrow i$
                $j'' \leftarrow j_\ell^+$
            end if
        end if
    end for
    if $\min\Delta = \infty$ then
        $\ell \leftarrow \ell + 1$
    else
        break;
    end if
end while
if $\min\Delta = \infty$ then
    return 0;
else
    return 1;
end if
```

Algorithm 6 FINDTOREMOVE($\hat{x}, i^*, j', j'', \min\Delta$).

```
$\ell \leftarrow 1$
while $\ell \le p$ do
    $\min\Delta \leftarrow \infty$
    for $i \in I : \hat{x}_{ij_\ell^-} = 1$ and $j_\ell^- \neq j'$ do
        if setting $\hat{x}_{ij_\ell^-} = 1$ does not break contiguity then
            $\delta \leftarrow c_{ij_\ell^+} - c_{ij'}$
            if $\delta < \min\Delta$ then
                $\min\Delta \leftarrow \delta$
                $i^* \leftarrow i$
                $j'' \leftarrow j_\ell^-$
            end if
        end if
    end for
    if $\min\Delta = \infty$ then
        $\ell \leftarrow \ell + 1$
    else
        break;
    end if
end while
if $\min\Delta = \infty$ then
    return 0;
else
    return 1;
end if
```

Remark 2.

a) Algorithm 5 allows disrupting a balanced district as an intermediate step to improve the balancing status in other districts.

This is the case because we do not allow disrupting contiguity. Suppose that we have an under-balanced district—in which we need to insert TUs. Suppose also that no overbalanced district can give away to our district a TU without disrupting contiguity. In this case, the way for performing the correction may include having a balanced district giving away a TU to the under-balanced district and, if necessary, in a later iteration receiving a TU from an over-balanced one.

b) We emphasize that there may not be a feasible solution to the chance constraints so, again, we stress that the execution of these algorithms is an attempt to render a solution feasibile assuming that this is possible. Unfortunately, we do not know beforehand whether γ is too binding, as our preliminary results suggest that, in fact, this is sometimes the case.

Algorithm 6 works in a similar way as Algorithm 5 but this time the focus is on over-balanced districts to find TUs that can be removed from them. Each of these two algorithms has a complexity of $\mathcal{O}(p(|I|-p)^3)$ for the worst case. Since they are embedded in Algorithm 4, this makes the time complexity of the latter to be of $\mathcal{O}(k_{\max}\ p(|I|-p)^3)$ for its worst case.

Lastly, we consider a final step in our overall procedure that consists of solving a 1-median problem within each district (if a feasible solution was found). This is motivated by the fact that after moving TUs across districts, updating the representatives may reduce the objective function value.

5. Computational tests

In this section we report on the computational tests performed to evaluate the relevant aspects of our work, namely:

(i) the relevance of considering a probabilistic approach for the problem, and

(ii) the effectiveness of the heuristic proposed in Sections 3 and 4.

We start by detailing the test bed instances. Afterwards, we discuss the use of the deterministic problem presented in Section 2.2 as a means to obtain (approximate) solutions to our districting problem. Finally we discuss the computational results for the heuristic procedure proposed.

5.1. Instances generation

The generation of the instances relies on the specification of two major aspects: those related with the base instances data of the problem which regards I, c_{ij} $(i, j \in I)$, p, α, and γ, and those related with the probability distributions investigated for the demand.

5.1.1. Base instances data

Each base instance contains the following characteristics: (i) a number of TUs, $|I|$, (ii) an adjacency matrix, and (iii) an assignment costs matrix. These data was generated in the following way:

- We have considered $|I| \in \{100, 150\}$.
- To create an adjacency matrix, for any two distinct TUs i and j, we have randomly selected a value in $\{0, 1\}$ following a discrete uniform distribution, indicating with 1 that these two TUs are adjacent, and with 0 otherwise. We then check whether the underlying graph is connected. If this is not the case, this adjacency matrix is discarded and the process is repeated until a connected graph is obtained.
- The assignment cost c_{ij} between two distinct TUs i and j has been randomly generated following a uniform distribution in the interval $[1000, 10000]$. We set to zero the distance from a TU to itself. Notice that this cost generation process does not ensure that the triangle inequality holds. For this reason, after having the cost matrix, we computed the shortest path on the underlying graph between every pair of TUs updating the corresponding entry in the cost matrix with the value of that path. However, it should be noted that our algorithms do not require the validity of the triangle inequality. Nevertheless, since we are working with a problem that has a strong geographical context, we believe that adequate assignment costs should have this feature namely because in practice they are often related to the distances.

Table 1
Probability distributions tested.

Random variable (d)	E[d]	Var[d]	RSD[d]
$U(a,b)$	$\frac{a+b}{2}$	$\frac{(b-a)^2}{12}$	$\frac{1}{\sqrt{3}}\frac{b-a}{b+a}$
$LogN(\mu,\sigma)$	$e^{\mu+\frac{\sigma^2}{2}}$	$e^{2\mu+\sigma^2}(e^{\sigma^2}-1)$	$\sqrt{e^{\sigma^2}-1}$
$P(\lambda)$	λ	λ	$\frac{1}{\sqrt{\lambda}}$
$Exp(\lambda)$	$\frac{1}{\lambda}$	$\frac{1}{\lambda^2}$	1

Table 2
Setting the parameters of the distributions as functions of the expected value and coefficient of variation.

Random variable (d)	Parameters
$U(a,b)$	$a = E(d)(1-\sqrt{3}*RSD)$ $b = E(d)(1+\sqrt{3}*RSD)$
$LogN(\mu,\sigma)$	$\mu = \ln(E(d)) - \ln\sqrt{RSD^2+1}$ $\sigma = \sqrt{ln(RSD^2+1)}$
$P(\lambda)$	$\lambda = \frac{1}{RSD^2}$ or $\lambda = E(d)$
$Exp(\lambda)$	$\lambda = \frac{1}{E(d)}$

- The values of the parameters that are still needed to be considered for specifying an instance $(p, \alpha$ and $\gamma)$, are detailed in Sections 5.3 and 5.4. In the latter section we also provide the details about other implementation details such as the values set for $N_{\min}$, $N_{\max}$, and PRECISION.

5.1.2. Demand generation

One central aspect of our work is the stochasticity of the demand. For performing the computational tests we considered four different probability distributions describing it: Uniform, Log-normal, Poisson, and Exponential. For each of these distributions we recall their expected value, variance, and relative standard deviation—see Table 1. We consider that the TUs have independent demands thus generating them in accordance to that.

For the Uniform and Log-normal distributions, we fixed the expected value equal to 50 and we varied the RSD[1], to obtain different values for the variance. In particular, we set RSD=0.125, 0.250, and 0.500, thus considering a standard deviation equal to 1/8, 1/4, and 1/2 (for the fixed expected value). The distributions obtained for each pair (E[d],Var[d]) are depicted in Fig. 1. The corresponding parameters are computed according to standard formulas that we detail in Table 2.

In the case of Poisson distribution, it is not possible to vary RSD without modifying the magnitude of the expected value. Hence, we keep setting RSD=0.125, 0.250, 0.500 as before but now we consider E[d]=64, 16, and 4, respectively, according to the relation reported in Table 2. Finally, for the Exponential distribution we note that it is characterized by an RSD=1; in this case, we calibrated the parameter of the distribution—λ—with the aim of obtaining the same expected values above chosen for the Poisson distribution. The distributions are reported in Fig. 2.

We also note that in the particular case of the Exponential distribution, we only need to test one value of λ since the probability stated in Constraints (5) is independent of λ. To see this, for every j such that $x_{jj} = 1$ let us define the set of TUs in this district, i.e., $I_j = \{i \in I \mid x_{ij} = 1\}$. Since $d_i \sim Exp(\lambda)$ for all $i \in I$, and given the independent demands, we

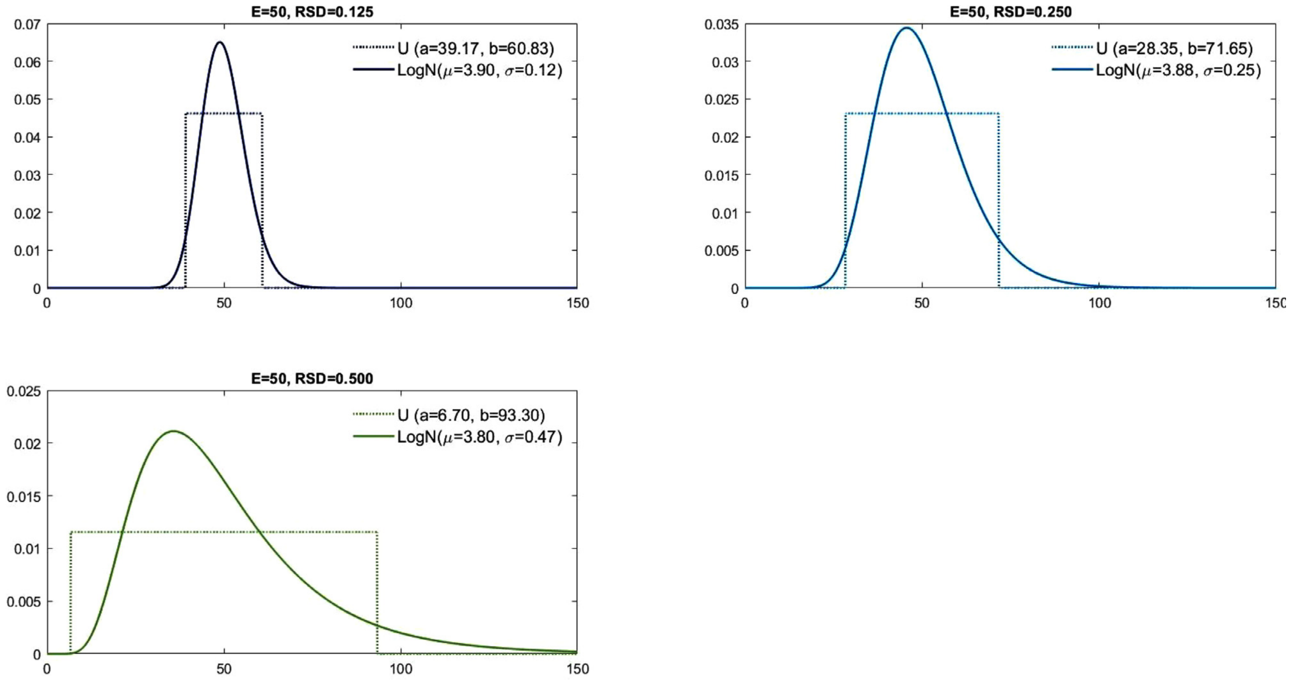

Fig. 1. Uniform and Log-normal distributions.

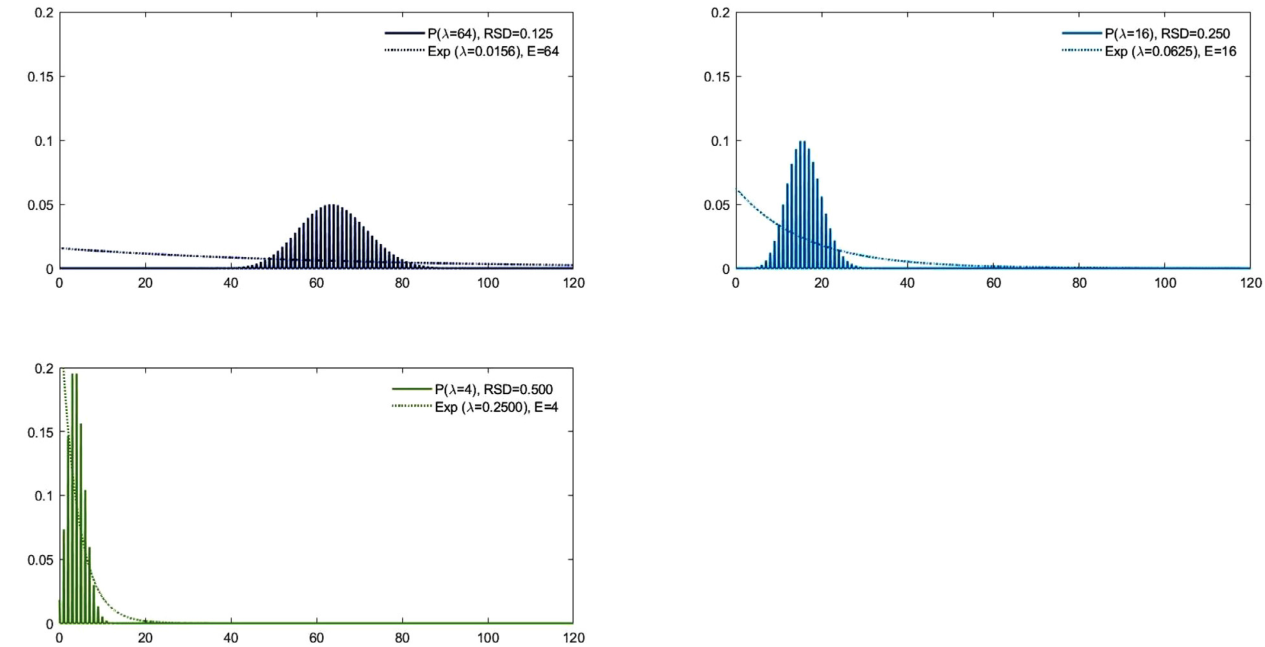

Fig. 2. Poisson and Exponential distributions.

know that $\sum_{i\in I} d_i \sim \text{Gamma}(|I|, \lambda)$ and $\sum_{i\in I} d_i x_{ij} \sim \text{Gamma}(|I_j|, \lambda)$. Hence, $W = (\sum_{i\in I} d_i x_{ij})/(\sum_{i\in I} d_i) \sim \text{Beta}(|I_j|, |I| - |I_j|)$, which is independent from λ. Hence, we have

$$\begin{aligned}\mathbb{P}\left[(1-\alpha)\frac{D}{p}x_{jj} \le \sum_{i\in I} d_i x_{ij} \le (1+\alpha)\frac{D}{p}x_{jj}\right] \\ = \mathbb{P}\left[\frac{1-\alpha}{p} \le \frac{\sum_{i\in I} d_i x_{ij}}{\sum_{i\in I} d_i} \le \frac{1+\alpha}{p}\right] \\ = I_{\frac{1+\alpha}{p}}(|I_j|, |I| - |I_j|) - I_{\frac{1-\alpha}{p}}(|I_j|, |I| - |I_j|),\end{aligned} \tag{11}$$

where $I_\rho(|I_j|, |I| - |I_j|)$ stands for the regularized incomplete beta function, i.e., the CDF of W in the designated points, which is constant w.r.t. λ. Note that the use of x_{jj} in the upper inequality of (11) is convenient for the presented argumentation despite being mathematically redundant.

5.2. Implementation details

All the results reported have been obtained with an Intel(R) Core(TM) i7–8750H CPU at 2.20 GHz, with 16 GiB of RAM running Windows 10 Pro–64 bits operating system. In particular, the heuristic procedure has been implemented in C using the MinGW

Table 3
Results of the single-scenario model.

					$\hat{\pi}$, estimate for the probability (5)					
					$\vert I\vert = 100$			$\vert I\vert = 150$		
Distribution	E[d]	RSD[d]	α		$p = 4$	$p = 6$	$p = 8$	$p = 4$	$p = 6$	$p = 8$
$U(39.17; 60.83)$	50.00	0.125	0.20		1.00	0.21	0.04	0.88	0.27	0.33
$U(28.35; 71.65)$	50.00	0.250	0.20		0.94	0.19	0.04	0.62	0.24	0.21
$U(6.70; 93.30)$	50.00	0.500	0.20		0.64	0.13	0.02	0.46	0.15	0.10
$U(39.17; 60.83)$	50.00	0.125	0.10		0.53	0.31	0.76	0.45	0.34	0.14
$U(28.35; 71.65)$	50.00	0.250	0.10		0.30	0.17	0.20	0.35	0.11	0.05
$U(6.70; 93.30)$	50.00	0.500	0.10		0.17	0.04	0.01	0.23	0.05	0.01
$LogN(3.90; 0.12)$	50.00	0.125	0.20		1.00	0.22	0.04	0.91	0.27	0.35
$LogN(3.88; 0.25)$	50.00	0.250	0.20		0.94	0.19	0.04	0.63	0.25	0.20
$LogN(3.80; 0.47)$	50.00	0.500	0.20		0.65	0.12	0.03	0.46	0.17	0.11
$LogN(3.90; 0.12)$	50.00	0.125	0.10		0.51	0.32	0.80	0.49	0.34	0.14
$LogN(3.88; 0.25)$	50.00	0.250	0.10		0.26	0.18	0.19	0.36	0.12	0.05
$LogN(3.80; 0.47)$	50.00	0.500	0.10		0.17	0.05	0.01	0.26	0.05	0.01
$P(64)$	50.00	0.125	0.20		1.00	0.21	0.04	0.88	0.28	0.37
$P(16)$	50.00	0.250	0.20		0.93	0.18	0.04	0.62	0.27	0.20
$P(4)$	50.00	0.500	0.20		0.63	0.13	0.02	0.50	0.19	0.12
$P(64)$	50.00	0.125	0.10		0.50	0.32	0.77	0.47	0.34	0.14
$P(16)$	50.00	0.250	0.10		0.27	0.17	0.18	0.35	0.12	0.05
$P(4)$	50.00	0.500	0.10		0.16	0.04	0.01	0.26	0.05	0.02
$E(1/64)$	50.00	1.000	0.20		0.27	0.04	0.01	0.28	0.07	0.02
$E(1/64)$	50.00	1.000	0.10		0.07	0.00	0.00	0.10	0.01	0.00
CPU (sec.)				Min	18.27	22.76	49.90	219.95	236.05	305.24
				Max	27.87	49.98	67.57	330.71	467.64	431.11
				Avg	22.80	34.82	57.01	275.37	352.63	369.43

compiler. A time limit for Algorithm 2 equal to 1000 seconds was imposed. The approximate deterministic models were coded in Python 3.6 and solved using IBM ILOG CPLEX 12.10.

5.3. Results for the deterministic problem

We start by presenting the results obtained using the single-scenario model proposed in Section 2.2, that is, the model that consists of minimizing (1), subject to (2)–(4), (6), (8)–(10).

We aim at investigating the quality of the districting solutions provided by that model with respect to the probabilistic constraints. To do so, once we obtain the solution for the model, we use a simulation algorithm to estimate the actual probability of that solution to satisfy the probabilistic constraints. This is accomplished using an algorithm in line with Algorithm 3 (considering only the elements used for estimating the probability). By doing so, we are also able to assess the added-value of using our approach to detriment of using the deterministic approximate model.

For our analysis, we considered the instances with $|I| = 100$ and 150. We solved the model for $p = 4, 6, 8,$ and $\alpha = 0.20, 0.10$. The results are presented in Table 3.

Observing this table we can draw several conclusions:

- For each distribution, apart from the Exponential, by increasing RSD the probability of satisfying the balancing requirement decreases.
- The Exponential seems to be the distribution that makes it harder to meet the balancing requirement.
- The number of nodes $|I|$ tends to affect the probability of having a balanced solution. For $p = 6, 8,$ it tends to increase and this is possibly due to the fact that by increasing the number of TUs the expected value of the total demand in the study region also increases, thus widening the range set for having a balanced solution. (While for $p = 4$, the probabilities are comparable or even higher for $|I| = 100$.)
- Conversely, the tolerance α has a double effect on the estimated probabilities. On the one hand, by decreasing it, we have tighter bounds for balancing constraints and this could make harder to satisfy them; on the other hand, we target a solution with more similar districts (in terms of the expected demand assigned to them). The overall effect results from a trade-off between these two conflicting aspects. This “double effect” is more evident for $p = 6$. For $p = 4$ we observe the probabilities increasing with α (both for $|I| = 100, 150$). Interestingly, for $p = 8$ and $|I| = 100$ the probabilities increase as α decreases (with the exception of the Exponential and of a few other cases). For $p = 8$ and $|I| = 150$, again the probabilities increase with α.
- Also p seems to have a double effect. On the one hand, the increase of p reduces the probabilities (we have indeed tighter bounds, which is consistent). However, we can also note that in some cases the probabilities increase: this happens for $|I| = 100, \alpha = 0.10, p = 8$ for all the distributions (Exponential excluded), in the cases with the lowest RSD (i.e., RSD = 0.125). This reveals that these parameters strongly interact, thus impacting the results.
- In any case, the estimated probabilities observed in this table are very low with the exception of the values corresponding to $p = 4$, which clearly indicates that the optimal solution provided by the deterministic approximate model will turn out to be a balanced solution with small probabilities. This calls for the design of a more robust/reliable districting map. As we will see, using our stochastic approach we are able to increase significantly the probability of satisfying the balancing requirements, under different scenarios, and thus

we are able to provide solutions that better hedge against uncertainty. This is what we discuss next.

- Finally, we note that the deterministic model can be solved to proven optimality within a very acceptable CPU time. We provide the information at the bottom of Table 3. It is worth pointing out that the small CPU times required to solve the deterministic approximation model also validate our choice in terms of the constraints adopted for ensuring contiguity.

5.4. Results for the heuristic procedure

An instance to our problem is fully specified by setting values for all parameters including γ—the given probability for the chance-constraints. This probability is a triggering factor for the need to invoke Algorithm 2. In other words, a solution resulting from the heuristic by [27] is to be "corrected" by Algorithm 2 if we observe a violation in the chance-constraints.

In our experiments, we run the heuristic for $\gamma \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$. The probability associated with each solution is estimated by means of the SIMULATE routine, detailed in Algorithm 3. To run it, we set $N_{\text{min}} = 100$, $N_{\text{max}} = 1000$ and PRECISION $= 0.001$. In other words, this routine stops either when the difference of the estimated probabilities in two consecutive runs is below 0.001 after a minimum of 100 trials, or when 1000 trials are performed.

In Tables 4 and 5 we present the results of the tests performed. The first column in these tables specify the distribution probability assumed for the demand. In the second column, we present the value of α—the tolerance defining the thresholds that render a solution balanced. The third column states the value of γ. Columns 4–12 contain the results of our heuristic for the three values of p tested. In particular, in columns 4, 7, and 10 we present the probability of satisfying the chance-constraints (i.e., $\hat{\pi}$) for the feasible solution found. The columns headed with "Δobj (%)" contain the increase in the cost w.r.t the optimal solution to the single-scenario model. In practice, it informs the reader on the values of the objective function of the solutions found by our algorithm. Columns 6, 9, and 12 contain the CPU time (in seconds) required by our heuristic. In these columns, "t.l." indicates that our heuristic was not able to find a feasible solution to the problem within the time limit (1000 seconds for Algorithm 2). The cells corresponding to such cases are highlighted in gray. Also in these cases, the value reported in the columns headed with "$\hat{\pi}$" is the maximum value obtained for the estimate of probability (5) found during the execution of the heuristic. Hence, it gives an indication of how far we can go in terms of satisfying the balancing constraints. The corresponding objective function value is used to calculate the associated value of "Δobj (%)". When one such case occurs, the instances for higher values of γ (other parameters unchanged) are not reported since variations in "$\hat{\pi}$" and "Δobj (%)" (as expected) were not detected.

The first relevant aspect emerging when comparing Tables 4 and 5 with Table 3 is that the optimal solution to the single scenario model is never feasible to our problem for $p = 6, 8$; only for $p = 4$ we have some feasible cases. This is an interesting observation since it highlights two things: first, the deterministic model is a poor simplification of our problem. Second, a lower bound on the optimal solution to the problem becomes available by solving the deterministic approximation. Such lower bound can be used to evaluate the quality of the feasible solutions obtained by our heuristic. Thus, the percentage values observed in the columns "Δobj (%)" can, in fact, be looked at as upper bounds on the optimal gap of our heuristic. Recall that "Δobj (%)" measures the gap between the solution value obtained by our heuristic (when a feasible solution is obtained) and the lower bound provided by the single-scenario model.

Several other conclusions can be drawn for the values presented in the tables:

- For the instances with 100 TUs, when the variance of the underlying distribution increases, there is a value of γ above which the instances seem to be infeasible, i.e., it is not possible to find a feasible solution satisfying the chance constraints. We save this information in the cells highlighted in gray. For instance, considering the Uniform distribution with $a = 6.70$ and $b = 93.30$ the highest value for the probability for which a feasible solution could be found for $p = 6$ and $\alpha = 0.2$ is 0.71.
  Of course, it is possible to claim that in some cases our heuristic was not able to find a feasible solution to the problem although it exists. However, we strongly believe that this is not the case since the balancing requirements impose a trade-off solution across the districts which may be difficult to ensure with a high probability if the demand is stochastic. When the variance of the underlying distribution increases, it is expected that no feasible solution satisfying the chance constraints for a probability above some threshold exists. Nonetheless, this effect is smoothed when we increase the number of TUs, as we can see when we compare the results in Tables 4 and 5. This change in terms of the success in finding feasible solutions is not surprising because more TUs for the same number of districts ($p = 6$) means that we can more easily ensure a balanced solution. We also have evidence that the value of p influences this outcome as well. Indeed, we can see that an increase in p corresponds to a reduction in the number of instances for which a feasible solution could be found. Interestingly, especially for $p = 4$ we are able to obtain solutions with a value equal to the optimal value of the deterministic problem (i.e., Δobj(%) $= 0.00$%), indicating that for a small number of districts the deterministic may provide a good approximation to our problem.
- When the demand follows an exponential distribution, it is clearly difficult to satisfy the chance constraints for intermediate to large values of γ. Using this distribution we only succeed in a few cases in finding a feasible solution to the problem ($p = 4$ and $|I| = 150$).
  Considering the single-scenario model we observe that its optimal solution satisfies the balancing constraints with probabilities that no one wishes to consider in the real-world—the values are too small. We recall that in the case of the Exponential distribution only one value for the parameter needs to be tested, so it does not help to deepen the testing for this distribution.
  The above observations give strength to the claim that if the demand follows an Exponential distribution, then the use of chance constraints should possibly be avoided, indicating that other modeling paradigms should be considered for capturing uncertainty, e.g., stochastic programming with recourse.
- As explained above, the percentages presented in the column "Δobj (%)" turn out to be an upper bound on the gap of the solution provided by the heuristic. Considering that the real gap can be much smaller (we cannot quantify how tight/loose the solution to the single-scenario model is as a lower bound) we conclude that our heuristic provides extremely good solutions to the problem. In fact, looking into the non-shadowed values presented in the tables we observe extremely small gaps—in most of the cases in between 2% and 3%. Nevertheless, since we are using a heuristic, we know that this may depend also on the way instances are generated.
- To complement the previous observation we note that in most of the cases our heuristic requires a very small CPU time. Given the quality of the solutions presumably obtained, this is another indication of the high performance of our developments.

Table 4
Computational results: $|I| = 100$. $\hat{\pi}$: estimate for the probability (5); t.l.: time limit exceeded.

			$p = 4$			$p = 6$			$p = 8$		
Distribution	α	γ	$\hat{\pi}$	Δobj(%)	CPU(sec.)	$\hat{\pi}$	Δobj(%)	CPU(sec.)	$\hat{\pi}$	Δobj(%)	CPU(sec.)
$U(39.17; 60.83)$	0.20	0.50	1.00	0.00%	1	0.97	0.71%	<1	0.95	0.77%	<1
		0.60	1.00	0.00%	<1	0.97	0.71%	1	0.95	0.77%	<1
		0.70	1.00	0.00%	<1	0.97	0.71%	<1	0.95	0.77%	1
		0.80	1.00	0.00%	<1	0.97	0.71%	<1	0.95	0.77%	<1
		0.90	1.00	0.00%	<1	0.97	0.71%	<1	0.95	0.77%	<1
$U(28.35; 71.65)$	0.20	0.50	0.97	0.00%	<1	0.66	1.05%	<1	0.63	0.81%	<1
		0.60	0.97	0.00%	<1	0.66	1.05%	1	0.63	0.81%	<1
		0.70	0.97	0.00%	<1	0.76	1.19%	<1	0.75	1.08%	<1
		0.80	0.97	0.00%	<1	0.94	1.27%	<1	0.96	1.67%	<1
		0.90	0.97	0.00%	<1	0.94	1.27%	<1	0.96	1.67%	<1
$U(6.70; 93.30)$	0.20	0.50	0.61	0.00%	<1	0.52	1.21%	<1	0.35	3.79%	t.l.
		0.60	0.61	0.00%	<1	0.61	1.47%	1			
		0.70	0.79	0.06%	<1	0.71	2.25%	69			
		0.80	0.93	0.16%	<1	0.71	2.25%	t.l.			
		0.90	0.93	0.16%	<1						
$U(39.17; 60.83)$	0.10	0.50	0.57	0.01%	<1	0.58	0.51%	<1	0.83	1.44%	<1
		0.60	0.70	0.02%	<1	0.73	0.56%	<1	0.83	1.44%	<1
		0.70	0.97	0.12%	<1	0.73	0.56%	1	0.83	1.44%	<1
		0.80	0.97	0.12%	<1	0.98	0.69%	<1	0.83	1.44%	<1
		0.90	0.97	0.12%	<1	0.98	0.69%	<1	0.91	4.27%	<1
$U(28.35; 71.65)$	0.10	0.50	0.79	0.11%	<1	0.51	0.70%	0	0.25	2.44%	t.l.
		0.60	0.79	0.11%	<1	0.60	1.21%	7			
		0.70	0.79	0.11%	<1	0.11	0.99%	t.l.			
		0.80	0.90	0.26%	<1						
		0.90	0.90	0.26%	<1						
$U(6.70; 93.30)$	0.10	0.50	0.45	0.33%	t.l.	0.11	0.99%	t.l.	0.03	3.86%	t.l.
$LogN(3.90; 0.12)$	0.20	0.50	1.00	0.00%	<1	0.50	0.57%	<1	0.99	0.77%	1
		0.60	1.00	0.00%	<1	0.96	0.71%	<1	0.99	0.77%	<1
		0.70	1.00	0.00%	<1	0.96	0.71%	<1	0.99	0.77%	<1
		0.80	1.00	0.00%	<1	0.96	0.71%	<1	0.99	0.77%	<1
		0.90	1.00	0.00%	<1	0.96	0.71%	<1	0.99	0.77%	<1
$LogN(3.88; 0.25)$	0.20	0.50	0.95	0.00%	<1	0.68	0.96%	<1	0.62	1.36%	1
		0.60	0.95	0.00%	<1	0.68	0.96%	<1	0.62	1.36%	<1
		0.70	0.95	0.00%	<1	0.83	1.41%	<1	0.77	1.79%	<1
		0.80	0.95	0.00%	<1	0.83	1.41%	<1	0.95	2.17%	<1
		0.90	0.95	0.00%	<1	0.92	1.67%	1	0.95	2.17%	<1
$LogN(3.80; 0.47)$	0.20	0.50	0.64	0.00%	<1	0.54	1.27%	<1	0.38	4.92%	t.l.
		0.60	0.64	0.00%	1	0.63	1.55%	<1			
		0.70	0.74	0.05%	<1	0.70	2.34%	5			
		0.80	0.83	0.11%	1	0.76	**1.76%**	t.l.			
		0.90	0.94	0.25%	<1						
$LogN(3.90; 0.12)$	0.10	0.50	0.55	0.01%	<1	0.56	0.54%	<1	0.78	0.67%	<1
		0.60	0.74	0.11%	<1	0.76	0.70%	1	0.78	0.67%	<1
		0.70	0.74	0.11%	<1	0.76	0.70%	<1	0.78	0.67%	<1
		0.80	0.99	0.12%	<1	0.97	0.86%	<1	0.81	1.94%	<1
		0.90	0.99	0.12%	<1	0.97	0.86%	<1	0.91	3.87%	2
$LogN(3.88; 0.25)$	0.10	0.50	0.74	0.12%	<1	0.55	0.70%	<1	0.24	4.14%	t.l.
		0.60	0.74	0.12%	<1	0.59	1.66%	t.l.			
		0.70	0.74	0.12%	<1						
		0.80	0.93	0.25%	<1						
		0.90	0.93	0.25%	<1						
$LogN(3.80; 0.47)$	0.10	0.50	0.46	0.26%	t.l.	0.12	1.31%	t.l.	0.03	4.47%	t.l.
$P(64)$	0.20	0.50	1.00	0.00%	<1	0.91	1.05%	1	0.97	0.77%	1
		0.60	1.00	0.00%	<1	0.91	1.05%	2	0.97	0.77%	1
		0.70	1.00	0.00%	<1	0.91	1.05%	1	0.97	0.77%	1
		0.80	1.00	0.00%	<1	0.91	1.05%	1	0.97	0.77%	1
		0.90	1.00	0.00%	<1	0.91	1.05%	1	0.97	0.77%	1
$P(16)$	0.20	0.50	0.93	0.00%	<1	0.68	0.73%	<1	0.51	1.33%	<1
		0.60	0.93	0.00%	<1	0.68	0.73%	1	0.61	1.92%	1
		0.70	0.93	0.00%	<1	0.81	0.91%	<1	0.75	2.44%	1
		0.80	0.93	0.00%	<1	0.81	0.91%	1	0.95	3.33%	<1
		0.90	0.93	0.00%	<1	0.90	1.02%	0	0.95	3.33%	<1
$P(4)$	0.20	0.50	0.63	0.00%	<1	0.53	1.27%	1	0.37	5.22%	t.l.
		0.60	0.63	0.00%	1	0.66	1.61%	<1			
		0.70	0.75	0.05%	<1	0.70	2.46%	44			
		0.80	0.94	0.16%	1	0.70	2.46%	t.l.			
		0.90	0.94	0.16%	1						
$P(64)$	0.10	0.50	0.51	0.01%	1	0.53	0.51%	1	0.71	1.34%	1
		0.60	0.72	0.11%	1	0.73	0.56%	1	0.71	1.34%	1
		0.70	0.72	0.11%	1	0.73	0.56%	1	0.71	1.34%	1
		0.80	0.97	0.12%	1	0.99	0.88%	2	0.81	3.06%	2
		0.90	0.97	0.12%	1	0.99	0.88%	1	0.92	3.45%	805
$P(16)$	0.10	0.50	0.71	0.11%	<1	0.52	0.81%	<1	0.24	5.41%	t.l.
		0.60	0.71	0.11%	<1	0.59	1.56%	t.l.			
		0.70	0.71	0.11%	<1						
		0.80	0.85	0.18%	<1						
		0.90	0.90	0.28%	<1						
$P(4)$	0.10	0.50	0.45	0.26%	t.l.	0.11	1.56%	t.l.			t.l.
$E(1/64)$	0.20	0.50	0.46	0.26%	t.l.	0.13	2.00%	t.l.	0.03	4.30%	t.l.
$E(1/64)$	0.10	0.50	0.11	0.25%	t.l.	0.02	1.22%	t.l.	0.00	5.96%	t.l.

Table 5
Computational results: $|I| = 150$. $\hat{\pi}$: estimate for the probability (5); t.l.: time limit exceeded.

			$p = 4$			$p = 6$			$p = 8$		
Distribution	α	γ	$\hat{\pi}$	Δobj(%)	CPU(sec.)	$\hat{\pi}$	Δobj(%)	CPU(sec.)	$\hat{\pi}$	Δobj(%)	CPU(sec.)
$U(39.17; 60.83)$	0.20	0.50	0.91	0.00%	<1	0.95	0.58%	<1	0.80	0.29%	<1
		0.60	0.91	0.00%	<1	0.95	0.58%	<1	0.80	0.29%	<1
		0.70	0.91	0.00%	<1	0.95	0.58%	<1	0.80	0.29%	<1
		0.80	0.91	0.00%	<1	0.95	0.58%	<1	0.80	0.29%	<1
		0.90	0.91	0.00%	<1	0.95	0.58%	<1	0.97	0.66%	<1
$U(28.35; 71.65)$	0.20	0.50	0.59	0.00%	<1	0.76	0.58%	<1	0.75	0.34%	<1
		0.60	0.89	0.02%	<1	0.76	0.58%	<1	0.75	0.34%	<1
		0.70	0.89	0.02%	<1	0.76	0.58%	<1	0.75	0.34%	<1
		0.80	0.89	0.02%	<1	0.94	0.86%	<1	0.88	0.72%	<1
		0.90	0.96	0.04%	<1	0.94	0.86%	<1	0.96	0.79%	<1
$U(6.70; 93.30)$	0.20	0.50	0.57	0.02%	<1	0.52	0.66%	<1	0.52	0.91%	<1
		0.60	0.79	0.04%	<1	0.67	0.93%	<1	0.63	1.66%	<1
		0.70	0.79	0.04%	<1	0.76	1.28%	<1	0.65	2.19%	t.l.
		0.80	0.86	0.07%	<1	0.82	1.64%	<1			
		0.90	0.91	0.24%	<1	0.90	1.26%	<1			
$U(39.17; 60.83)$	0.10	0.50	0.93	0.06%	<1	0.65	0.82%	<1	0.90	0.71%	<1
		0.60	0.93	0.06%	<1	0.65	0.82%	<1	0.90	0.71%	<1
		0.70	0.93	0.06%	<1	1.00	0.88%	<1	0.90	0.71%	1
		0.80	0.93	0.06%	<1	1.00	0.88%	<1	0.90	0.71%	<1
		0.90	0.93	0.06%	<1	1.00	0.88%	<1	0.90	0.71%	<1
$U(28.35; 71.65)$	0.10	0.50	0.64	0.06%	<1	0.57	0.70%	<1	0.51	0.80%	<1
		0.60	0.64	0.06%	<1	0.62	0.98%	<1	0.58	2.74%	t.l.
		0.70	0.82	0.17%	<1	0.75	1.04%	<1			
		0.80	0.82	0.17%	<1	0.81	1.15%	<1			
		0.90	0.94	0.29%	<1	0.94	0.86%	1			
$U(6.70; 93.30)$	0.10	0.50	0.54	0.20%	<1	0.22	0.85%	t.l.	0.06	2.43%	t.l.
		0.60	0.63	0.39%	12						
		0.70	0.63	0.39%	t.l.						
$LogN(3.90; 0.12)$	0.20	0.50	0.91	0.00%	<1	0.96	0.58%	<1	0.80	0.29%	<1
		0.60	0.91	0.00%	<1	0.96	0.58%	<1	0.80	0.29%	<1
		0.70	0.91	0.00%	<1	0.96	0.58%	<1	0.80	0.29%	<1
		0.80	0.91	0.00%	<1	0.96	0.58%	<1	0.99	0.34%	<1
		0.90	0.91	0.00%	<1	0.96	0.58%	<1	0.99	0.34%	<1
$LogN(3.88; 0.25)$	0.20	0.50	0.62	0.00%	<1	0.75	0.58%	<1	0.74	0.66%	1
		0.60	0.62	0.00%	1	0.75	0.58%	1	0.74	0.66%	<1
		0.70	0.86	0.02%	1	0.75	0.58%	1	0.74	0.66%	<1
		0.80	0.86	0.02%	<1	0.91	0.86%	<1	0.84	0.72%	<1
		0.90	0.96	0.04%	1	0.91	0.86%	<1	0.99	0.84%	1
$LogN(3.80; 0.47)$	0.20	0.50	0.61	0.02%	<1	0.52	0.86%	<1	0.59	0.95%	<1
		0.60	0.61	0.02%	<1	0.66	0.93%	<1	0.63	1.08%	1
		0.70	0.79	0.04%	<1	0.71	1.08%	<1	0.69	1.18%	t.l.
		0.80	0.92	0.07%	<1	0.91	1.17%	1			
		0.90	0.92	0.07%	<1	0.91	1.17%	1			
$LogN(3.90; 0.12)$	0.10	0.50	0.93	0.06%	1	0.69	0.82%	<1	0.92	1.03%	0
		0.60	0.93	0.06%	<1	0.69	0.82%	<1	0.92	1.03%	0
		0.70	0.93	0.06%	<1	0.99	0.88%	<1	0.92	1.03%	0
		0.80	0.93	0.06%	<1	0.99	0.88%	<1	0.92	1.03%	0
		0.90	0.93	0.06%	<1	0.99	0.88%	<1	0.92	1.03%	0
$LogN(3.88; 0.25)$	0.10	0.50	0.71	0.06%	<1	0.57	0.70%	1	0.53	1.19%	1
		0.60	0.71	0.06%	<1	0.63	0.76%	1	0.55	2.59%	t.l.
		0.70	0.71	0.06%	1	0.71	0.92%	1			
		0.80	0.86	0.17%	1	0.81	0.98%	1			
		0.90	0.96	0.20%	1	0.92	**0.76%**	2			
$LogN(3.80; 0.47)$	0.10	0.50	0.57	0.32%	1	0.25	0.85%	t.l.	0.07	2.00%	t.l.
		0.60	0.62	0.27%	1						
		0.70	0.64	**0.19%**	t.l.						
$P(64)$	0.20	0.50	0.88	0.00%	<1	0.94	0.58%	1	0.79	0.29%	<1
		0.60	0.88	0.00%	1	0.94	0.58%	1	0.79	0.29%	1
		0.70	0.88	0.00%	1	0.94	0.58%	1	0.79	0.29%	1
		0.80	0.88	0.00%	1	0.94	0.58%	1	0.97	0.34%	1
		0.90	0.99	0.02%	1	0.94	0.58%	1	0.97	0.34%	1
$P(16)$	0.20	0.50	0.66	0.00%	<1	0.74	0.58%	1	0.74	0.66%	<1
		0.60	0.66	0.00%	<1	0.74	0.58%	1	0.74	0.66%	<1
		0.70	0.92	0.02%	<1	0.74	0.58%	1	0.74	0.66%	<1
		0.80	0.92	0.02%	<1	0.92	0.86%	1	0.92	0.78%	<1
		0.90	0.92	0.02%	<1	0.92	0.86%	1	0.92	0.78%	<1
$P(4)$	0.20	0.50	0.62	0.02%	<1	0.52	0.86%	<1	0.56	1.27%	1
		0.60	0.62	0.02%	1	0.63	0.93%	<1	0.62	1.82%	1
		0.70	0.75	0.04%	1	0.71	0.96%	1	0.70	2.46%	950
		0.80	0.85	0.07%	1	0.90	1.10%	1	0.70	2.46%	t.l.
		0.90	0.91	0.12%	1	0.91	1.37%	1			
$P(64)$	0.10	0.50	0.96	0.06%	1	0.68	0.67%	1	0.86	0.71%	1
		0.60	0.96	0.06%	2	0.68	0.67%	1	0.86	0.71%	1
		0.70	0.96	0.06%	1	1.00	0.70%	2	0.86	0.71%	1
		0.80	0.96	0.06%	1	1.00	0.70%	2	0.86	0.71%	1
		0.90	0.96	0.06%	1	1.00	0.70%	2	0.97	0.84%	1
$P(16)$	0.10	0.50	0.64	0.06%	<1	0.58	0.88%	1	0.52	0.90%	1
		0.60	0.64	0.06%	1	0.65	1.02%	1	0.57	3.04%	t.l.
		0.70	0.83	0.17%	1	0.74	1.13%	1			t.l.
		0.80	0.83	0.17%	1	0.86	1.34%	1			t.l.
		0.90	0.94	0.20%	1	0.91	1.18%	2			t.l.
$P(4)$	0.10	0.50	0.56	0.20%	<1	0.23	0.85%	t.l.	0.07	1.75%	t.l.
		0.60	0.60	0.24%	36						
		0.70	0.62	0.32%	t.l.						
$E(1/64)$	0.20	0.50	0.53	0.24%	<1	0.23	1.45%	t.l.	0.07	1.54%	t.l.
		0.60	0.60	0.49%	2						
		0.70	0.64	0.42%	t.l.						
$E(1/64)$	0.10	0.50	0.17	0.33%	t.l.	0.03	1.39%	t.l.	0.01	2.22%	t.l.

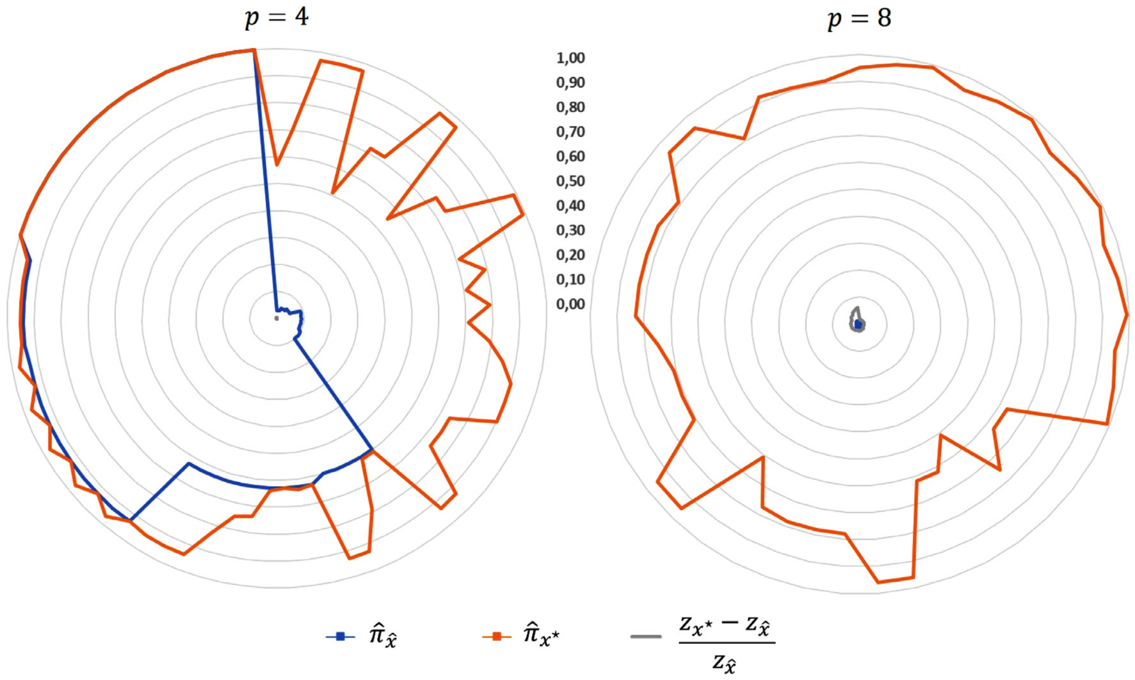

Fig. 3. Evaluation of the phases of the heuristic.

Finally, we point out one situation that may emerge from the application of our procedure: the best feasible solution found for some value of γ (other parameters unchanged) may have a lower objective function value than that found for a lower value of γ. In this case, the former solution dominates the latter for the lower value of γ. This situation (although scarce) can occur and it is highlighted in the results (Table 4, $LogN(3.80; 0.47)$, $|I| = 100, p = 6, \alpha = 0.20, \gamma = 0.80$; Table 5, $LogN(3.88; 0.25)$, $|I| = 150, p = 6, \alpha = 0.10, \gamma = 0.90$; Table 5: $LogN(3.80; 0.47)$, $|I| = 150, p = 4, \alpha = 0.20, \gamma = 0.70$.). This behavior may of course occur due to the randomized nature of the demand scenarios we generate.

5.5. Additional insights

After having analyzed in-depth the capability of our heuristic to deal with the chance-constrained problem we are investigating, there are still some unanswered queries: What is the contribution of each component to the success of the overall procedure? What behavior can we expect for large-scale instances? In this section, we provide some additional insights related to these aspects.

5.5.1. The different phases of the heuristic

Recall the two components of our heuristic algorithm: construction of a feasible solution to the underlying p-median problem, and correction of that solution ensuring that the chance constraints are satisfied. Let us denote by $\hat{x}$ the solution obtained after the construction phase (not necessarily a feasible solution to the chance-constrained problem) and by $x^\star$ the final solution produced by our heuristic. Additionally, let $\hat{\pi}_{\hat{x}}$ and $\hat{\pi}_{x^\star}$ be the estimates for probability (5) associated respectively to these solutions. Finally, let $z_{\hat{x}}$ and $z_{x^\star}$ be the corresponding objective function values. The relevance of this information is twofold:

(i) The value of $\hat{\pi}_{\hat{x}}$ indicates the degree of (in)feasibility after the construction phase, and $\hat{\pi}_{x^\star}$ the improvement in terms of attaining probability (5), i.e., the benefit from using the balancing procedure;

(ii) The relative gap $(z_{x^\star} - z_{x^\star})/z_{x^\star}$ indicates the extra-cost paid to ensure feasibility if it does not hold after the construction phase.

For this analysis, we focus on the instances with $|I| = 100$. Furthermore, we present results for $p = 4, 8$ (i.e., 75 and 45 instances—see Table 4). We are omitting the results for $p = 6$ since they are similar to those for $p = 8$. We summarize the results in Fig. 3. From this figure, and focusing on the left-hand side image, we see that for $p = 4$ in almost half of the instances (36 out of 75) the solution provided by the constructive phase is already feasible to our problem (the orange and blue lines are overlapping). In the others, further effort is required. Nevertheless, the worsening of the objective function is very low (we cannot even devise the gray line in the image). As already we noted, the problem is relatively easy to solve for $p = 4$. Regarding the right-hand side image, i.e. for $p = 8$, the solution provided by the constructive phase is never feasible. In fact, $\hat{\pi}_{\hat{x}}$ is always equal to 0.00 (the blue line cannot be devised in the image). The extra-costs paid for rendering a solution feasible are more noticeable, although quite low: they range from a minimum of 1.56%, up to a maximum of 6.16%, with an average value equal to 2.64%.

Overall, we conclude that at the expense of a small increase in the objective function value (compactness of the districts), we can do much better in terms of the probability of having a balanced solution.

5.5.2. Results on large-scale instances

To investigate the performance of our heuristic on large-scale instances, we generated two additional sets of instances: with 500 and 1000 TUs. In the first case we considered $p = 10, 20$; in the second we assumed $p = 20, 40$. Two values of α were analyzed: $\alpha = 0.10, 0.20$. The entire set of experiments detailed in Section 5.4 is repeated for these large-scale instances, whose results are summarized in Tables 6 and 7. Since the purpose at this stage is to devise how far we can go using our heuristic, for each probability distribution and for the sake of brevity we report only the maximum value of γ for which we were able to find a feasible solution and the corresponding CPU time (in seconds). The cases

Table 6
Computational results: $|I| = 500$ N/A: not available; t.l.: time limit exceeded.

	$p = 10$				$p = 20$			
	$\alpha = 0.10$		$\alpha = 0.20$		$\alpha = 0.10$		$\alpha = 0.20$	
Distribution	γ	CPU (sec.)	γ	CPU (sec.)	γ	CPU (sec.)	γ	CPU (sec.)
$U(39.17; 60.83)$	0.90	2	0.90	2	0.90	2	0.90	2
$U(28.35; 71.65)$	0.90	2	0.90	3	0.50	200	0.90	2
$U(6.70; 93.30)$	N/A	t.l.	0.90	3	N/A	t.l.	N/A	t.l.
$LogN(3.90; 0.12)$	0.90	3	0.90	2	0.90	2	0.90	2
$LogN(3.88; 0.25)$	0.90	4	0.90	3	N/A	t.l.	0.90	2
$LogN(3.80; 0.47)$	N/A	t.l.	0.90	4	N/A	t.l.	0.50	13
$P(64)$	0.90	10	0.90	7	0.90	15	0.90	4
$P(16)$	0.90	7	0.90	6	0.50	930	0.90	3
$P(4)$	N/A	t.l.	0.90	4	N/A	t.l.	0.50	105
$E(1/64)$	N/A	t.l.	N/A	t.l.	N/A	t.l.	N/A	t.l.

Table 7
Computational results: $|I| = 1000$ N/A: not available; t.l.: time limit exceeded.

	$p = 20$				$p = 40$			
	$\alpha = 0.10$		$\alpha = 0.20$		$\alpha = 0.10$		$\alpha = 0.20$	
Distribution	γ	CPU (sec.)	γ	CPU (sec.)	γ	CPU (sec.)	γ	CPU (sec.)
$U(39.17; 60.83)$	0.90	10	0.90	9	0.90	11	0.90	11
$U(28.35; 71.65)$	0.90	11	0.90	9	N/A	t.l.	0.90	12
$U(6.70; 93.30)$	N/A	t.l.	0.90	11	N/A	t.l.	N/A	t.l.
$LogN(3.90; 0.12)$	0.90	11	0.90	9	0.90	13	0.90	13
$LogN(3.88; 0.25)$	0.90	14	0.90	10	N/A	t.l.	0.90	15
$LogN(3.80; 0.47)$	N/A	t.l.	0.90	13	N/A	t.l.	N/A	t.l.
$P(64)$	0.90	36	0.90	10	0.90	59	0.90	27
$P(16)$	0.90	27	0.90	11	0.50	t.l.	0.90	24
$P(4)$	N/A	t.l.	0.90	14	N/A	t.l.	N/A	t.l.
$E(1/64)$	N/A	t.l.	N/A	t.l.	N/A	t.l.	N/A	t.l.

for which no feasible solution was found within the time limit (t.l.) imposed are indicated by "N/A" under columns headed with γ.

The first aspect emerging from these tables is that when a feasible solution is found a small CPU time is spent in that. As before, a smaller value of α leads to more difficult instances: in this case, the CPU time is not always negligible and seems to increase when α decreases. Additionally, the chance-constraints seem more difficult to satisfy (γ is possibly binding for smaller values).

6. Conclusions

In this paper, we have introduced a chance-constrained model for districting problems with stochastic demands. The chance constraints are associated with balancing requirements. Due to the difficulty in deriving a compact deterministic equivalent for the problem, we proposed a heuristic for finding a feasible solution. This is a two-phase procedure: in the first phase, a solution is obtained ignoring the chance constraints; in the second stage, that solution is checked in terms of those constraints and corrected if necessary. Each solution is assessed by estimating the probability of turning out to be a balanced solution when demand is disclosed. Such an estimate is obtained by simulation.

Computational tests were performed using a set of instances generated considering different combinations of parameters and different probability distributions for the demands. The results show that by considering a procedure that takes uncertainty into account we can find high-quality solutions for the chance-constrained model. Moreover, those solutions seem to hedge very well against uncertainty. This is confirmed when looking at the results provided by a traditional single-scenario solution, which leads to solutions that meet the balancing requirements with a very small (often unacceptable) probability.

We were also able to identify a distribution—the Exponential—that works invariably badly with chance-constraints, indicating that if this is the underlying distribution, then possibly a chance-constrained model is not the best paradigm to hedge against uncertainty. On the other hand, we found out distributions of practical relevance (e.g., Log-normal) for which we were able to obtain extremely good results thus showing that a chance-constrained model is worth considering.

In our view, the proposed procedure can help practitioners and decision-makers in various practical settings where uncertainty in the demands occurs. As the (stochastic) districting literature highlights, this is especially relevant in territory design problems in the service and distribution industries. Such an approach may help designing compact and connected districts where only information on the distribution of the demands is available, and hence several demand scenarios can be figured out. In those cases, our approach provides practitioners with a practical solution method guaranteeing (in most of the cases) the satisfaction of a workload balancing requirement with a minimum desired confidence/service level. Additionally, it may be worthwhile investigating its application to some healthcare problems, e.g., home care planning (26,28) or blood supply chain management (4).

The work done opens some directions for further research. First, in case the demand follows an Exponential distribution, it is clear that probabilistic constraints are not a good tool to consider since hardly will we be able to find an acceptable solution to the problem. One possibility, in this case, is to resort to stochastic programming with recourse although the infinite support of the underlying random vector calls, again, for an approximate method such as Sample Average Approximation (19).

Since the major contribution of our work is an approximate algorithm for the chance-constrained districting problem, it was intensively tested using a set of instances specifically designed for that. Still, other possibilities could have been considered for testing our procedure, which calls for more intensive testing. First, the use of the Gabriel graph (12). Second, the application of the model to real districting instances like the ones in the field of distribution districting (e.g., postal service) to test the solution procedure in a real setting where balancing on customers' routing is a planning requirement. Moreover, we could also think of considering various demand distributions within the same instance, so as to address cases where demands may vary in dependence on some socio-economic and geographical variables (e.g., income level, rural vs. urban areas, etc.).

Finally, as we note from our experiments, very often the solution obtained from the constructive phase is very far from satisfying the balancing requirements with the given probability. Sometimes, the solution obtained from the deterministic equivalent model seems to provide a better reference point in this sense. This observation opens to further developments aimed at exploring various constructive methods (possibly taking somehow balancing into account) to ease the balancing phase of our solution procedure.

CRediT authorship contribution statement

Antonio Diglio: Conceptualization, Methodology, Investigation, Validation. **Juanjo Peiró:** Conceptualization, Methodology, Investigation, Validation. **Carmela Piccolo:** Conceptualization, Methodology, Investigation, Validation. **Francisco Saldanha-da-Gama:** Conceptualization, Methodology, Investigation, Validation.

Acknowledgments

This work was supported by National Funding from FCT — Fundação para a Ciéncia e Tecnologia, Portugal, under the project UIDB/04561/2020, by Gobierno de España through Ministerio de Ciencia, Innovación y Ministerio de Universidades and Fondo Europeo de Desarrollo Regional (FEDER) under the project PGC2018-099428-B-100, and by Gobierno de España through Ministerio de Educación y Formación Profesional under the grant CAS19/00076.

The authors would like to thank the anonymous referees for their valuable comments on our work, which helped us improving the manuscript.

References

[1] Bard JF, Jarrah AI. Large-scale constrained clustering for rationalizing pickup and delivery operations. Transportation Research Part B: Methodological 2009;43(5):542–61.

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

[3] Bergey BK, Ragsdale CT, Hoskote M. A simulated annealing genetic algorithm for the electrical power districting problem. Ann Oper Res 2003;121(1–4):33–55.

[4] Bruno G, Diglio A, Piccolo C, Cannavacciuolo L. Territorial reorganization of regional blood management systems: evidences from an italian case study. Omega (Westport) 2019;89:54–70.

[5] Butsch A, Kalcsics J, Laporte G. Districting for arc routing. INFORMS J Comput 2014;26(4):809–24.

[6] Carlsson JG, Delage E. Robust partitioning for stochastic multivehicle routing. Oper Res 2013;61:727–44.

[7] Diglio A, Nickel S, Saldanha-da-Gama F. Towards a stochastic programming modeling framework for districting. Ann Oper Res 2020;292:248–85.

[8] Dugošija D, Savić A, Maksimović Z. A new integer linear programming formulation for the problem of political districting. Ann Oper Res 2020;288:247–63.

[9] Elizondo-Amaya M, Rıos-Mercado RZ, Morton DP, Kutanoglu E. A tabu search approach for a territory design problem with stochastic demands. Spanish Congress on Informatics (CEDI); 2013.

[10] Erkut E, Ülküsal Y, Yeniçerioğlu O. A comparison of p-dispersion heuristics. Computers & Operations Research 1994;21:1103–13.

[11] Farughi H, Tavana M, Mostafayi S, Arteaga FJS. A novel optimization model for designing compact, balanced, and contiguous healthcare districts. Journal of the Operational Research Society 2019. doi:10.1080/01605682.2019.1621217.

[12] Gabriel K, Sokal R. A new statistical approach to geographic variation analysis. Systematic Biology, Society of Systematic Biologists 1969;18:259–78.

[13] Haugland D, Ho SC, Laporte G. Designing delivery districts for the vehicle routing problem with stochastic demands. Eur J Oper Res 2007;180(3):997–1010.

[14] Hess SW, Weaver JB, Siegfeldt HJ, Whelan JN, Zitlau PA. Nonpartisan political redistricting by computer. Oper Res 1965;13(6):998–1006.

[15] Juan AA, Faulin J, Grasman SE, Rabe M, Figueira G. A review of simheuristics: extending metaheuristics to deal with stochastic combinatorial optimization problems. Oper Res Perspect 2015;2:62–72.

[16] Kalcsics J, Ríos-Mercado R. Districting problems. In: Laporte G, Nickel S, Saldanha-da-Gama F, editors. Location Science. Springer International Publishing, 2nd edition; 2019. p. 705–43. 25

[17] Kınay ÖB, Kara BY, Saldanha-da Gama F, Correia I. Modeling the shelter site location problem using chance constraints: a case study for istanbul. Eur J Oper Res 2018;270(1):132–45.

[18] Kınay ÖB, Saldanha-da Gama F, Kara BY. On multi-criteria chance-constrained capacitated single-source discrete facility location problems. Omega (Westport) 2019;83:107–22.

[19] Kleywegt A, Shapiro A, de Mello TH. The sample average approximation method for stochastic discrete optimization. SIAM J Optim 2001;12(2):479–502.

[20] Kong Y, Zhu Y, Wang Y. A center-based modeling approach to solve the districting problem. International Journal of Geographical Information Science 2019;33:368–84.

[21] Lei H, Laporte G, Guo B. Districting for routing with stochastic customers. EURO Journal on Transportation and Logistics 2012;1:67–85.

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

[23] Liu H, Erdogan A, Lin R, Tsao H-SJ. Mathematical models of political districting for more representative governments. Computers & Industrial Engineering 2020;140:106265.

[24] Marín A, Pelegrín M. Facility location nder uncertainty. In: Laporte G, Nickel S, Saldanha-da-Gama F, editors. Location Science. Springer International Publishing, 2nd edition; 2019. p. 25–50. 2

[25] Mayorga M, Bandara D, McLay L. Districting and dispatching policies for emergency medical service systems to improve patient survival. IIE Trans Healthc Syst Eng 2013;3:39–56.

[26] Mosquera F, Smet P, Berghe GV. Flexible home care scheduling. Omega (Westport) 2019;83:80–95.

[27] Resende MG, Werneck RF. A hybrid heuristic for the p-median problem. Journal of Heuristics 2004;10(1):59–88.

[28] Restrepo MI, Rousseau L-M, Vallée J. Home healthcare integrated staffing and scheduling. Omega (Westport) 2020;95:102057.

[29] Ricca F, Scozzari A, Simeone B. Political districting: from classical models to recent approaches. Ann Oper Res 2013;204(1):271–99.

[30] Optimal districting and territory design. Ríos-Mercado RZ, editor. Cham: Springer International Publishing; 2020.

[31] Ríos-Mercado RZ, Álvarez-Socarrás AM, Castrillón A, López-Locés MC. A location-allocation-improvement heuristic for districting with multiple-activity balancing constraints and p-median-based dispersion minimization. Computers & Operations Research 2021;126:105106.

[32] Ríos-Mercado RZ, Fernández E. A reactive GRASP for a commercial territory design problem with multiple balancing requirements. Computers & Operations Research 2009;36:755–76.

[33] Ríos-Mercado RZ, López-Pérez JF. Commercial territory design planning with realignment and disjoint assignment requirements. Omega (Westport) 2013;41(3):525–35.

[34] Salazar-Aguilar MA, Ríos-Mercado RZ, Cabrera-Ríos M. New models for commercial territory design. Networks and Spatial Economics 2011;11(3):487–507.

[35] Sandoval MG, Díaz JA, Ríos-Mercado RZ. An improved exact algorithm for a territory design problem with p-center-based dispersion minimization. Expert Syst Appl 2020;146:113150.

[36] Shirabe T. A model of contiguity for spatial unit allocation. Geogr Anal 2005;37:2–16.

[37] Shirabe T. Districting modeling with exact contiguity constraints. Environment and Planning B: Planning and Design 2009;36:1053–66.

[38] Xie S, Ouyang Y. Railroad caller districting with reliability, contiguity, balance, and compactness considerations. Transportation Research Part C: Emerging Technologies 2016;63:65–76.

[39] Yaník S, Kalcsics J, Nickel S, Bozkaya B. A multi-period multi-criteria districting problem applied to primary care scheme with gradual assignment. International Transactions in Operational Research 2019;26:1676–97.