European Journal of Operational Research 307 (2023) 233–248

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Stochastics and Statistics

Approximation schemes for districting problems with probabilistic constraints

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

[a] Universitá degli Studi di Napoli Federico II, Department of Industrial Engineering (DII), Piazzale Tecchio, 80 Naples 80125, Italy
[b] Departament d'Estadestica i Investigació Operativa, Facultat de Ciéncies Matemátiques, Universitat de Valéncia, 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 8 March 2022
Accepted 5 September 2022
Available online 9 September 2022

Keywords:
Location
Districting
Stochastic demand
Chance-constraint balancing
Heuristics

ABSTRACT

In this work a districting problem with stochastic demand is investigated. Chance-constraints are used to model the balancing requirements. Explicit contiguity constraints are also considered. After motivating the problem and discussing several modeling aspects, an approximate deterministic counterpart is proposed which is the core of new solution algorithms devised. The latter are based upon a location-allocation scheme, whose first step consists of considering either a problem with a sample of scenarios or a sample of single-scenario problems. This leads to two variants of a new heuristic. The second version calls for the use of a so-called attractiveness function as a means to find a good trade-off between the (approximate) solutions obtained for the single-scenario problems. Different definitions of such functions are discussed. Extensive computational tests were performed whose results are reported.

1. Introduction

A districting problem consists of partitioning a geographical area containing many Territory Units (TUs) into smaller districts taking into account aspects such as integrity, balancing, contiguity, and compactness (Kalcsics & Ríos-Mercado 2019 and Ríos-Mercado 2020).

This is a research field with many applications such as: design of commercial sales areas (Ríos-Mercado & Escalante 2016, Ríos-Mercado & Fernández 2009), design of police districts (Camacho-Collados & Liberatore 2015, Camacho-Collados, Liberatore, & Angulo 2015), distribution logistics (Bender, Kalcsics, Nickel, & Pouls 2018, Bender, Kalcsics, & Meyer 2020, Konur & Geunes 2019), energy and power distribution (Yanık, Sürer, & Öztayşi 2016, De Assis, Franca, & Usberti 2014), political districting (Dugošija, Savić, & Maksimović 2020, Liu, Erdogan, Lin, & Tsao 2020, Ricca, Scozzari, & Simeone 2013), strategic service planning, e.g., in health care systems (Mostafayi Darmian, Fattahi, & Keyvanshokooh 2021,Farughi, Tavana, Mostafayi, & Arteaga 2019, Yaník, Kalcsics, Nickel, & Bozkaya 2019, Benzarti, Sahin, & Dallery 2013), administrative services (Bruno, Diglio, Melisi, & Piccolo 2017a, Bruno, Genovese, & Piccolo 2017b), education (Caro, Shirabe, Guignard, & Weintraub 2004, Schoepfle & Church 1991), transportation (Butsch, Kalcsics, & Laporte 2014, Carlsson 2012), and waste collection (Ríos-Mercado & Bard 2019, Mourão, Nunes, & Prins 2009).

The districting problem we are focusing on is motivated by the need of considering the uncertain nature of the demand originating from TUs in service-oriented applications. When designing a district map, one of the most important properties to be guaranteed is the balancing requirement, according to which all districts should be as similar as possible in terms of their demand or activity level (see the discussion in Ríos-Mercado, Álvarez-Socarrás, Castrillón, & López-Locés 2021). If the uncertain nature of the demand is not adequately accounted for, one may easily end up with a set of districts that fails to be balanced.

In this paper we consider a districting problem with stochastic demand. Given the number of districts to be designed, the goal is to find a districting solution whose probability of being balanced is above a certain user-defined threshold. Contiguity constraints are explicitly considered, ensuring that, when moving between two TUs within the same district, there is always a path that does not cross a different district. Integrity is also ensured by imposing that each TU should be included in one and only one district. Finally, a surrogate measure of compactness is adopted which defines an objective to optimize. In practice, we are dealing with a chance-

* Corresponding author.
E-mail address: antonio.diglio@unina.it (A. Diglio).

https://doi.org/10.1016/j.ejor.2022.09.005
0377-2217/

constrained districting problem (for chance-constrained programming, the interested readers are referred to the seminal works by Charnes & Cooper, 1959 and Miller & Wagner, 1965, or the studies by Nemirovski & Shapiro, 2007 and Nemirovski, 2012, to name a few).

For the above problem we discuss an optimization model that, unfortunately, cannot be tackled directly either because the support of the underlying random vector is infinite (e.g., the underlying joint cumulative distribution function is continuous) or because that support has too large cardinality, thus inducing a large-scale extensive deterministic equivalent.

Thinking ahead to developing at least approximate procedures for the problem, we derive an approximate deterministic counterpart that we consider as a starting point for developing a matheuristic. The goal is naturally to find (hopefully) good feasible solutions to the problem.

The matheuristic we propose consists of an iterative location-allocation scheme: a set of centers is selected with the demand nodes being allocated to them. This, in turn, may call for a better choice in terms of the centers and consequently to another reallocation of the demand points. The process continues until no improvement is obtained. We propose two variants of this mechanism that differ in the way the initial set of centers is determined. The new procedures are assessed by the results of an extensive set of computational tests performed that we report on.

The problem we are investigating is, in fact, the same that was studied by Diglio, Peiró, Piccolo, & Saldanha-da-Gama (2021). In that paper the authors proposed a simheuristic (Juan, Faulin, Grasman, Rabe, & Figueira 2015), combining simulation and optimization, as a means to find feasible solutions to the problem. Due to the general shape of the underlying cumulative distribution function, simulation was carried out for assessing the probabilistic constraints. In the current paper we focus on the case in which uncertainty is captured by a finite number of scenarios, even if that number is quite large (Correia & Saldanha-da-Gama 2019).

Other alternative frameworks to use chance-constraints have been investigated in the context of districting problems under uncertainty. Elizondo-Amaya, Rıos-Mercado, Morton, & Kutanoglu (2013) look into a single-stage stochastic programming problem. Assuming stochastic demand, the goal is to make a here-and-now decision minimizing the expected value of the maximum demand across the districts. A Tabu Search algorithm is developed for finding feasible solutions. Diglio, Nickel, & Saldanha-da-Gama (2020) proposed a two-stage stochastic programming model for the case in which demand is uncertain and a recourse action can be taken as a means to react to a possibly not-so-good initial districting solution. The recourse includes re-districting, i.e., some TUs may be moved from one district to another as a way to better handle the actually occurring demand. Stochastic programming was also considered by Enayati, Özaltın, & Mayorga (2020), who seek to plan for the location of a set of ambulances in such a way that the expected coverage is maximized. The extensive form of the deterministic equivalent is solved using an off-the-shelf solver. The relevance of uncertainty in districting problems emerging in healthcare had been also stressed by Yanık & Bozkaya (2020). In particular, the authors identify visiting and travel times as sources of uncertainty in addition to demand. In line with the same type of uncertainty, Nikzad, Bashiri, & Abbasi (2021) consider a problem in which districting is to be integrated with dimensioning (or resources needed) and routing. The authors propose a progressive-hedging procedure for finding feasible solutions to the resulting stochastic programming problem. The relevance of capturing uncertainty in spatial data when performing territory design has also been recognized by Wei, Rey, & Grubesic (2022), who present a statistical approach for districting. Finally, we quote the work by Mostafayi Darmian et al. (2021), who capture uncertainty in a healthcare districting problem by means of uncertainty sets: a robust optimization model is derived. Uncertainty is associated with the demand for the service. An evolutionary algorithm is proposed for finding feasible solutions to the deterministic counterpart.

The above references reveal that the work aiming at embedding uncertainty aspects in districting problems is still scarce. Nevertheless, the increase trend in terms of such research avenue is clear.

As mentioned above, in this paper, we develop a matheuristic for a chance-constrained districting problem. This is a type of heuristic that makes use of optimization models as a component in the process leading to a feasible solution. In the context of districting, the use of such type of methodology has been proposed by Nikzad et al. (2021), although, as we underlined, those authors focused in a problem that is clearly beyond districting due to the additional decisions to make. In the very specific context of districting, a matheuristic was proposed by Solana, Díaz, & Luna (2019) as well as by Kong, Zhu, & Wang (2019). In the first work, the authors consider a model with a surrogate dispersion and look for a balanced solution. Afterwards, the solutions are modified so that contiguity is ensured. In the second work, a multi-start weighted k-medoids algorithm is embedded in a center model proposed by the authors to efficiently obtain good feasible solutions.

A central aspect in the matheuristic we propose in the current paper is the way for tackling a set of constraints that are relevant in districting: contiguity (Diglio et al. 2021, Salazar-Aguilar, Ríos-Mercado, & Cabrera-Ríos 2011, and Shirabe 2005; 2009). In particular, we adopt the mathematical representation proposed by Salazar-Aguilar et al. (2011) as well as the row-generation scheme those authors adopt for dealing, on the fly, with the exponential number of such constraints.

For the sake of completeness, it has to be noted that one may resort also to Sample Average Approximation (SAA) methods to deal with stochastic programs with a high number of scenarios (possibly infinite). Introduced by Kleywegt, Shapiro, & Homem-de Mello (2002), such technique has seen various applications in the field of facility location and network design (e.g., Kiya & Davoudpour, 2012; Santoso, Ahmed, Goetschalckx, & Shapiro, 2005). Also, these procedures have also been used to tackle general chance-constrained problems, as in Pagnoncelli, Ahmed, & Shapiro (2009) and in Luedtke & Ahmed (2008). Specifically, in the latter reference, the authors introduced a so-called Sample Approximation procedure, which we also resort to in the present paper. Here, in fact, we develop an approximate deterministic counterpart for chance-constrained districting (the “true” problem), in which uncertainty can be captured by a finite (but still large and randomly drawn) number of scenarios. The model is then tackled by a new matheuristic scheme, for which different variants are proposed. All the obtained solutions are validated through an ex-post simulation, involving a different large sample of demand values, to avoid potential “overfitting” issues and assess their “feasibility” to the true problem, in terms of meeting the balancing contraints with the required probability. We test the proposed approach through an extensive computational experience, involving instances with up to 1000 TUs and 1000 scenarios, which prove our procedure to be competitive also w.r.t. to an alternative heuristic proposed for the same problem under investigation.

The remainder of this paper is organized as follows. In Section 2, we provide all details about the problem we are investigating which includes an enhanced mathematical model. In Section 3, an approximation is proposed for the deterministic equivalent problem. Such approximation is at the core of the procedures proposed in the following Section 4. Section 5 reports on a series of extensive computational tests performed for assessing the new methodologies proposed. The paper ends with an overview of the work done and some directions for further research.

2. Mathematical modeling

As we have already pointed out, the starting point for the current paper is the problem investigated by Diglio et al. (2021) as well as the initial mathematical model they derived. We consider a finite set, N, of TUs that we want to partition into p districts. For each district, a representative TU is to be designated, so that the inclusion of a TU in a district is established by allocating it to the selected representative of the district.

The geographical region of interest is assumed to be well captured by a connected network. We assume that the demand in each district (or an adequate measure for activity of interest) is stochastic, that is, can be represented by a random variable with a given CDF. The customary balancing imposed to districting solutions is captured in our case by means of probabilistic constraints, i.e., we seek to find a solution such that the probability of being balanced (every district is balanced) is above a minimum threshold. Finally, as often done in districting problems, we consider a surrogate measure for compactness. In particular, we consider it somehow embedded in a cost we adopt for allocating a TU to a district.

We denote by $G = (N, A)$ the network underlying the region to partition into districts, with N denoting the set of TUs and A containing the direct connection between TUs. From this network we can build a set N_i, for every $i \in N$, containing the TUs adjacent to $i \in N$: $N_i = \{j \in N : (i, j) \in A \vee (j, i) \in A\}$. Additionally, the following notation is considered.

Deterministic Parameters:

- c_{ij}, cost for assigning TU i to TU j $(i, j \in N)$;
- p, number of districts to create;
- α, maximum desirable deviation for the demand in a district w.r.t. a reference value, where $\alpha \in [0, 1]$;
- γ, probability that a districting solution turns out to be balanced.
- d_i, demand in TU $i \in N$; $\boldsymbol{\xi} = [d_i]_{i \in N}$;
- D, total demand $(D = \sum_{i \in N} d_i)$.

The problem we are investigating can be formulated mathematically considering the set of binary variables introduced in the seminal paper by Hess, Weaver, Siegfeldt, Whelan, & Zitlau (1965): for $i, j \in N$, x_{ij} is equal to one if and only if TU i is allocated to (the district whose representative is) TU j. The particular case where $x_{jj} = 1$ indicates that j is the TU representative (or *center*) of its district. With these variables we have the following optimization model that we call (M_1):

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

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

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

$$\mathbb{P}_{\boldsymbol{\xi}}\left[(1-\alpha)\frac{D}{p} x_{jj} \le \sum_{i \in N} d_i x_{ij} \le (1+\alpha)\frac{D}{p} x_{jj}, \ j \in N \right] \ge \gamma, \tag{4}$$

$$\sum_{i \in S} x_{ij} - \sum_{i \in \cup_{\nu \in S}(N_\nu \setminus S)} x_{ij} \le |S| - 1, \quad j \in N, \ S \subset [N \setminus (N_j \cup \{j\})], \tag{5}$$

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

The objective function (1) quantifies the total assignment cost to be minimized. Constraints (2) ensure that each TU is assigned to exactly one district; Equality (3) states that p districts are to be created.

The chance-constrained balancing requirement are formulated according to Inequality (4): solutions must be balanced with probability at least equal to γ. Constraints (5) ensure the solution contiguity: for any given subset S of nodes assigned to center j not containing node j there must be an arc between S and the set containing j, that is, at least one element in S assigned to j. Finally, (6) define the domain of the decision variables.

This model captures features from three models that have been proposed in the literature for districting problems. The probabilistic constraints are adopted from Diglio et al. (2021). The exponentially-many contiguity constraints were introduced by Salazar-Aguilar et al. (2011) and embedded them in a cutting plane approach for solving the problem they were focusing on. Note, however, that this representation of contiguity has a clear advantage: no additional decision variables are necessary (unlike other possibilities adopted in the literature). Finally, the other components of the above model are gathered from the seminal paper by Hess et al. (1965). In particular, the costs c_{ij} are related with the distances. Denoting by ℓ_{ij} the distance between i and j $(i, j \in N)$ and by taking $c_{ij} = \ell_{ij}$ we turn the objective function into a so-called compactness measure known as *moment of inertia* (see that work for further details).

Salazar-Aguilar et al. (2011) enhanced their mathematical model by adding the following inequalities:

$$\sum_{\ell \in N_i} x_{\ell j} \ge x_{ij}, \qquad j \in N, \ i \in N \setminus (\{j\} \cup N_j). \tag{7}$$

These constraints are, in fact, valid inequalities for our model: they state that, if some TU i assigned to TU j is not adjacent to j, then at least another TU ℓ adjacent to i that is also allocated to j so that contiguity is not broken. In the above mentioned work, the authors realize that the addition of these inequalities to their original formulation helps reducing the computational effort needed to solve the problem up to proven optimality using a row-generation approach. We also consider this enhancement in our model. Summing up, model (M_1) we seek to solve consists of minimizing (1), subject to (2)–(7).

3. In search for a deterministic counterpart

Handling the model presented in the previous section requires deriving a (desirably compact) deterministic equivalent model, in which the probabilistic constraints (4) are replaced by a deterministic counterpart. At this aim, let us assume that uncertainty can be captured by a finite set of scenarios, say Ω, and that we can estimate accurately the probability associated to the occurence of each scenario $\omega \in \Omega$, say π_ω. In this case, we may index in Ω the demand vector as well as the reference thresholds for balancing constraints, in the following way:

- $d_{i\omega}$, demand of TU $i \in N$ under scenario $\omega \in \Omega$;
- μ_ω, reference value for the demand to be served in each district under scenario $\omega \in \Omega$. A natural choice for μ_ω is D_ω / p, where D_ω is the total demand under scenario ω $(D_\omega = \sum_{i \in N} d_{i\omega})$.

Moreover, we may also introduce the following decision variables:

$$\chi_\omega = \begin{cases} 1, \text{ if a districting plan is not balanced under scenario } \omega \in \Omega, \\ 0, \text{ otherwise.} \end{cases}$$

The model that allows us to find an approximate solution to our problem, based upon a finite random sample of demand scenarios Ω, can be formulated as:

$$\begin{aligned} &\text{minimize} && (1) \\ &\text{subject to} && (2)-(3), (5)-(7), \\ & && \sum_{i \in N} d_{i\omega} x_{ij} + h_\omega \chi_\omega \ge (1-\alpha)\mu_\omega x_{jj}, \qquad j \in N, \omega \in \Omega, \end{aligned} \tag{8}$$

$$\sum_{i\in N} d_{i\omega}x_{ij} - l_\omega\chi_\omega \le (1+\alpha)\mu_\omega x_{jj}, \qquad j\in N, \omega\in\Omega, \tag{9}$$

$$\sum_{\omega\in\Omega}\pi_\omega\chi_\omega \le 1-\gamma, \tag{10}$$

$$\chi_\omega \in \{0,1\}, \qquad \omega\in\Omega. \tag{11}$$

Inequalities (8)–(9) are the "new" balancing constraints, which are now restricted to the random sample Ω. Parameters h_ω and l_ω are large numbers that help to activate variables χ_ω for the scenarios in which the solution is not balanced. In other words, if, for a given scenario ω, at least one district is under(over)-served, then the corresponding χ_ω variable becomes equal to 1, thus rendering inequalities (8) (respectively, inequalities (9)) inactive. Constraint (10) guarantees that the joint probability of having solutions not balanced under the considered scenarios is at most equal to $(1-\gamma)$. Hence, this allows to find a solution that has a probability at least equal to γ of being balanced, which is exactly what we want.

When tackling real applications, the support of the random vector $\boldsymbol{\xi}$ may have very large cardinality, possibly infinite. This is the case of demands that are reasonably described by continuous probability distributions. In this situation, one possibility for finding an approximate solution to our problem consists of drawing a random sample (even of a very large size) from the distribution underlying $\boldsymbol{\xi}$, thus obtaining a finite set Ω of scenarios for the demand. For such a random sample, an estimate for the probability of a given districting solution to be balanced is the proportion of scenarios for which the solution is balanced. An approximate solution to our problem can thus be found as a solution that is balanced for at least $\gamma|\Omega|$ of its elements. In order to guarantee this, it is possible to replace constraints (10) with the following one in the above model formulation:

$$\sum_{\omega\in\Omega}\chi_\omega \le \lfloor(1-\gamma)|\Omega|\rfloor. \tag{12}$$

Inequality (12) ensures that the number of unbalanced scenarios is at most $\lfloor(1-\gamma)|\Omega|\rfloor$, i.e., that the districting is balanced with the given probability γ.

We focus the remainder of the paper on the latter case, since it is the most challenging and also the most likely to face in practice. Hence, we seek to solve model (M_2), which consists of minimizing (1), subject to (2)–(3), (5)–(9), (11)–(12) to find an approximate solution to our problem. As underlined above, the purpose of constraints (8) and (9) is to activate a χ-variable when the solution is unbalanced in the corresponding scenario. To this end, an appropriate setting of parameters h_ω and l_ω is important. Of course, one could just set them to very large arbitrary numbers, but this can easily lead to a very weak formulation. Given that the total demand assigned to each district is for sure in between the trivial lower bound $LB_\omega = 0$ and the trivial upper bound $UB_\omega = D_\omega$, we can improve these bounds as stated in the following result.

Proposition 1. *Constraints (8) and (9) can be rewritten, respectively, as follows:*

$$\sum_{i\in N} d_{i\omega}x_{ij} + [(1-\alpha)\mu_\omega - d_{[1]\omega}]\chi_\omega \ge (1-\alpha)\mu_\omega x_{jj}, \quad j\in N, \omega\in\Omega, \tag{13}$$

$$\sum_{i\in N} d_{i\omega}x_{ij} - [(p-1-\alpha)\mu_\omega - \sum_{\ell=1}^{p-1} d_{[\ell]\omega}]\chi_\omega \le (1+\alpha)\mu_\omega x_{jj}, \quad j\in N, \omega\in\Omega, \tag{14}$$

with $d_{[\ell]\omega}$ denoting the ℓth demand value when the demands under scenario ω are sorted non-decreasingly.

Proof: We first note that each representative is assigned to itself and thus each district will contain at least one TU. For each scenario ω, let $(d_{[1]\omega}, d_{[2]\omega}, .., d_{[N]\omega})$ represent the demands sorted non-decreasingly.

Given the second above observation, we directly conclude that the minimum demand assigned to each district is greater than or equal to the lowest value of the demands under that scenario, i.e., $LB_\omega = d_{[1]\omega}$, $\omega\in\Omega$.

On the other hand, under each scenario $\omega\in\Omega$, the maximum demand a district can serve is given by the total demand D_ω, minus the lowest $(p-1)$ values of the demands occurring in that scenario.

Let us focus on a single district. In case we are very "unlucky", the other $(p-1)$ districts will only serve the demand of their representatives. Accordingly, the lowest value of the total demand served by the other districts would be equal to the sum of the lowest $(p-1)$ values of the demands under a given scenario.

Accordingly, for $\omega\in\Omega$ we can set:

$$h_\omega = (1-\alpha)\mu_\omega - d_{[1]\omega}, \qquad \omega\in\Omega;$$

$$l_\omega = (p-1-\alpha)\mu_\omega - \sum_{\ell=1}^{p-1} d_{[\ell]\omega}, \qquad \omega\in\Omega.$$

From here, the result follows.◇

Note that the discussed problem is $\mathcal{NP}$-hard. In fact, if we consider very high values for α (i.e., balancing constraints are relaxed), and the particular case of a complete underlying graph G, a general districting problem reduces to a p-median problem, which is well-known to be $\mathcal{NP}$-hard (see the discussion provided by Ríos-Mercado et al., 2021). Our problem is no exception. Indeed, if we set $\gamma = 0$ in (12), we are essentially relaxing balancing constraints (8)–(9) across the overall sample of scenarios Ω. Accordingly, under the more restrictive hypothesis that the underlying graph G is complete, our problem (excluding valid inequalities (7), which help ensuring connectivity) also reduces to a p-median problem.

4. Heuristic solutions

We introduce in this section two heuristic procedures for tackling model (M_2). Both are what we call matheuristics since they make use of the optimization model as a component of the solution search process. Furthermore, both procedures adopt a common scheme in location analysis which consists of iterating between location and allocation decisions until no further improvement is obtained. We revisit this process in Algorithm 1.

Algorithm 1 Classical location-allocation scheme.

1: Select p initial centers and denote the resulting set by I_C;
2: **repeat**
3: $I^* \leftarrow I_C$; // best set of centers so far
4: Solve the underlying optimization model fixing $x_{jj} = 1, \forall j \in I^*$;
5: Find the best center within each group;
6: $I_C \leftarrow$ new set of p centers;
7: **until** $I_C = I^*$ // no improvement achieved trying to change the centers
8: **return** $x^\star$; // a feasible districting solution

In Algorithm 1 we see the location phase in lines 1 and 5, and the allocation one in line 4. Note that the center update in line 5 is accomplished by means of solving a 1-median problem within each district.

In the context of districting, the above scheme leads to initially fixing p representatives (*Location phase*). Then we assign TUs to these representatives to obtain a balanced and contiguous districts (*Allocation phase*). Afterwards we update the representatives. We keep iterating until the representatives do not change in two consecutive iterations. The main loop in fact defines an *Improvement phase*. The algorithm returns a feasible districting solution, say $x^\star$, whose representatives are $I^\star$.

Given that the above location-allocation scheme defines actually a local search descent method, it is of major relevance to find a good set of initial centers, which is what we explain next. Not only can this impact on the quality of the final solution but also on the computational effort required to execute the algorithm. The two algorithms that we devise can be looked at as variants of the above general search scheme with different mechanisms proposed for finding an initial set of representative TUs.

4.1. A sub-sampling scheme

The first procedure we propose is based upon considering an initial restricted model, say ($M_2[\Omega']$), that results from restricting model (M_2) to a random sample of scenarios Ω' drawn from set Ω. Of course, we target a sample such that $|\Omega'|$ is clearly smaller than $|\Omega|$. The restricted model is then used to obtain an initial promising set of representative TUs. After that, we recover the full model and we iterate the location-allocation scheme. We formalize the procedure in Algorithm 2.

Algorithm 2 A sub-sampling scheme.

1: Draw a random sample Ω' from sample Ω;
2: Consider the restricted model ($M_2[\Omega']$);
3: Select p centers for the restricted model ($M_2[\Omega']$) and denote the resulting set by I_C;
4: Solve the restricted model ($M_2[\Omega']$) fixing $x_{jj} = 1$, $\forall j \in I_C$;
5: Find the best center within each district;
6: $I'_C \leftarrow$ new set of centers;
7: **repeat**
8: $I^* \leftarrow I'_C$; // best set of centers so far
9: Solve model (M_2) fixing $x_{jj} = 1$, $\forall j \in I^*$;
10: Find the best center within each district;
11: $I'_C \leftarrow$ new set of p centers;
12: **until** $I'_C = I^*$; // no improvement achieved trying to change the centers
13: **return** $x^\star$; // a feasible districting solution

After drawing a sample Ω' of scenarios we solve the corresponding restricted model. We do not iterate at this stage since we want this phase of the procedure to be relatively fast. The set of centers corresponding to the obtained solution for model ($M_2[\Omega']$) is then imposed to model (M_2) which, in turn, is tackled in the main loop. Solving model ($M_2[\Omega']$) corresponds to the location phase in the location-allocation scheme for the problem at hand. The underlying idea behind this solution approach is that solving a "smaller problem" may be useful to obtain a promising initial set of representative TUs for the original problem.

We perform the selection of p centers in line 3 of Algorithm 2 in two ways: either by using the heuristic by Resende & Werneck (2004), or by using the well-known p-means algorithm as did by Ríos-Mercado et al. (2021) in a similar problem. The resolution of model ($M_2[\Omega']$) in line 4 and model (M_2) in line 9 are accomplished using the row-generation algorithm proposed by Salazar-Aguilar et al. (2011). In this step, we restrict the model by considering the decision variables that are strictly necessary, namely, those defined only w.r.t. to the specific set of centers just identified. This choice leads to a significant reduction in terms of the number of decision variables and constraints. Finally, the best center in each district (lines 5 and 10) is found solving the 1-median problem induced by the TUs in the district, minimizing the allocation cost (i.e., optimizing compactness).

4.2. A single-scenario-based scheme

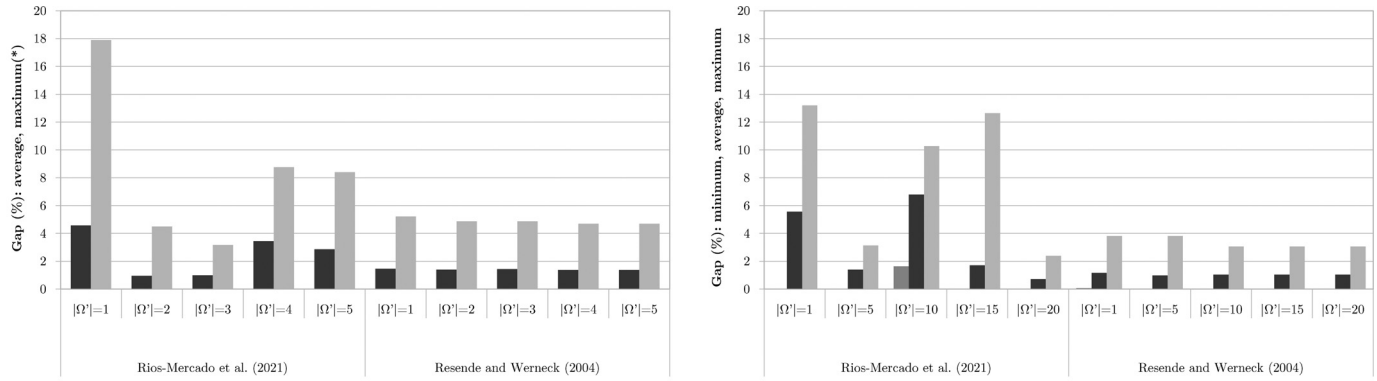
In the second solution method we propose, the idea for finding the initial set of p centers is to consider a certain maximum number of single-scenario problems. Each scenario to consider is sampled (without repetition) from Ω. Once a single-scenario problem is defined, it is tackled by the solver with a time limit imposed. To avoid spending too much time in this stage of the algorithm, such time limit may be rather short and thus some single-scenario problems may not be successfully solved (we provide additional details when reporting on the computational tests performed in Section 5).

An important note here is that when we plan for a specific scenario, we are facing a deterministic setting: the balancing constraints has to be satisfied for the single scenario at hand (which corresponds to directly set $\gamma = 1$).

Ideally we would like to successfully solve a certain number m of single-scenario problems with m being a user-defined parameter. Thus we must set the maximum number of single-scenario problems to consider larger than or equal to m. Taking the m (or less) sets of p centers found in the above process (corresponding to the sets of the single-scenario problems successfully solved within the time limit), we define the *attractiveness* of each set or TU for becoming a representative of a district. The p most attractive TUs are then selected and set as the initial set of centers to be fixed in the complete model (M_2). From here, the classical location-allocation scheme is executed. The procedure is formalized in Algorithm 3.

Algorithm 3 A single-scenario-based scheme.

1: Draw a random sample Ω' without repetition from sample Ω;
2: counter $\leftarrow 0$;
3: **for** each $\omega \in \Omega'$ **do**
4: Consider the restricted model ($M_2[\omega]$); // single-scenario problem
5: Select p centers for the restricted model ($M_2[\omega]$) and denote the resulting set by I_C;
6: Attempt solving the restricted model ($M_2[\omega]$) fixing $x_{jj} = 1$, $\forall j \in I_C$;
7: **if** succeded **then**
8: counter $\leftarrow$ counter + 1; // the restricted model was successfully solved
9: Find the best center within each district;
10: Update the attractiveness measure;
11: **if** (counter $\geq$ m) **then**
12: break; // exit the `for` loop
13: **end if**
14: **end if**
15: **end for**
16: Select the most attractive p centers and denote the resulting set by I'_C;
17: **repeat**
18: $I^* \leftarrow I'_C$; // best set of centers so far
19: Solve model (M_2) fixing $x_{jj} = 1$, $\forall j \in I^*$;
20: Find the best center within each district;
21: $I'_C \leftarrow$ new set of p centers;
22: **until** $I'_C = I^*$; // no improvement achieved trying to change the centers
23: **return** $x^\star$; // a feasible districting solution



(a) $|\Omega| = 10$. (*) minimum values are omitted since they are all equal to zero.

(b) $|\Omega| = 100$.

Fig. 1. Procedure 1—percentage gap, $|N| = 100$.

In this algorithm, lines 1–10 correspond to the location phase in the overall location-allocation scheme. The selection of p centers in line 5 is accomplished again using one of two existing heuristics: the one by Resende & Werneck (2004) and the p-means algorithm as in Ríos-Mercado et al. (2021). The resolution of models ($M_2[\omega]$) in line 6 and model (M_2) in line 19 are accomplished as in the first procedure, using the row-generation algorithm proposed by Salazar-Aguilar et al. (2011). Such models are restricted considering the decision variables (and the constraints) that are strictly necessary. Finally, like in the first solution procedure, the best center in each district (lines 9 and 20) is found solving the 1-median problem induced by the TUs in each district to optimize compactness.

4.2.1. Measuring the attractiveness of the TUs

What remains to be discussed in the second procedure just presented regards measuring the attractiveness of the TUs (line 16 in Algorithm 3).

Let us denote by m the actual number of single-scenario problems successfully solved (desirably the original number we targeted). For $k = 1, \ldots, m$, let us denote by $\hat{\mathbf{x}}^k$ the solution obtained for the kth single-scenario successfully solved, and by $z(\hat{\mathbf{x}}^k)$ its corresponding objective function value. Additionally, we consider $b(\hat{\mathbf{x}}^k)$ the number of balanced scenarios associated with solution $\hat{\mathbf{x}}^k$. This number can be easily computed by considering the demand vector associated with each scenario $\omega \in \Omega$ and checking whether the solution is balanced for that demand occurrence. We propose two ways for measuring the attractiveness of a TU.

Probabilistic attractiveness measure. In this case, for each TU $j \in N$, we compute the probability, say P_j, of having j as a representative across the m scenarios:

$$P_j = \frac{1}{m} \sum_{k=1}^{m} \hat{x}^k_{jj}.$$

In Algorithm 3, line 16 we select the p centers having the highest of such probabilities.

A merit attractiveness measure. In this case, for each solution $\hat{\mathbf{x}}^k$ ($k = 1, \ldots, m$), we evaluate the attractiveness of the corresponding set of centers to be selected as the initial representatives to solve problem (M_2). Following Ríos-Mercado et al. (2021), we compute the value of a so-called merit function G as:

$$G(\hat{\mathbf{x}}^k) = \lambda \left(\min_{\ell \in \{1,\ldots,m\}} z(\hat{\mathbf{x}}^\ell) \times \frac{1}{z(\hat{\mathbf{x}}^k)} \right) + (1-\lambda) \left(\frac{1}{\max_{\ell \in \{1,\ldots,m\}} b(\hat{\mathbf{x}}^\ell)} \times b(\hat{\mathbf{x}}^k) \right), \quad \lambda \in [0,1].$$

In the above expression, λ is a predefined weight that allows giving more relevance to (i) a smaller objective function value, or (ii) to a higher number of balanced scenarios. When adopting this merit function, in line 16 of Algorithm 3 we select the set of p centers (among the m available from the solutions of the single-scenario problems) having the highest value of such measure.

5. Experimental results

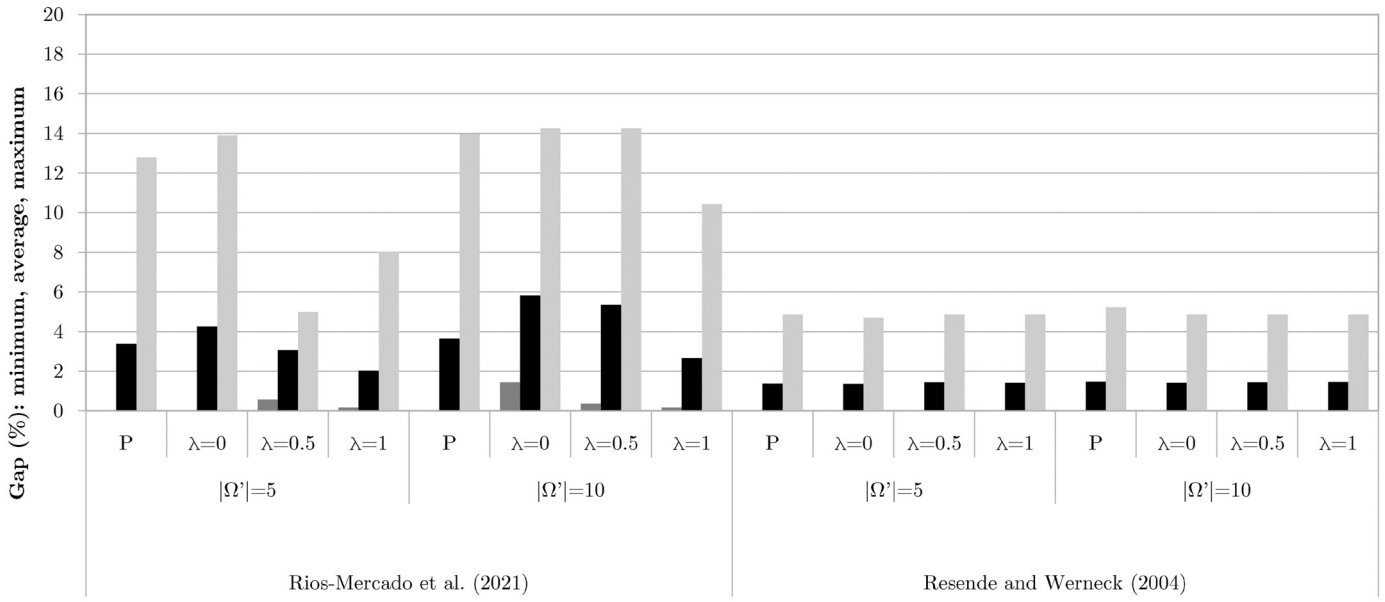
In this section, we present the results of the experiments performed to assess the contribution of our work. We start by describing the test instances considered and also by detailing the experimental setting adopted. Afterwards, we focus on a set of small instances for which the problem can be solved up to proven optimality. Those results are used in the following section for assessing the quality of the new procedures proposed in Section 4. We then proceed by reporting on the tests performed on instances comprising up to 1000 TUs and 1000 base scenarios. Finally, we conclude comparing the new procedures with an alternative one that exists in the literature.

5.1. Test data

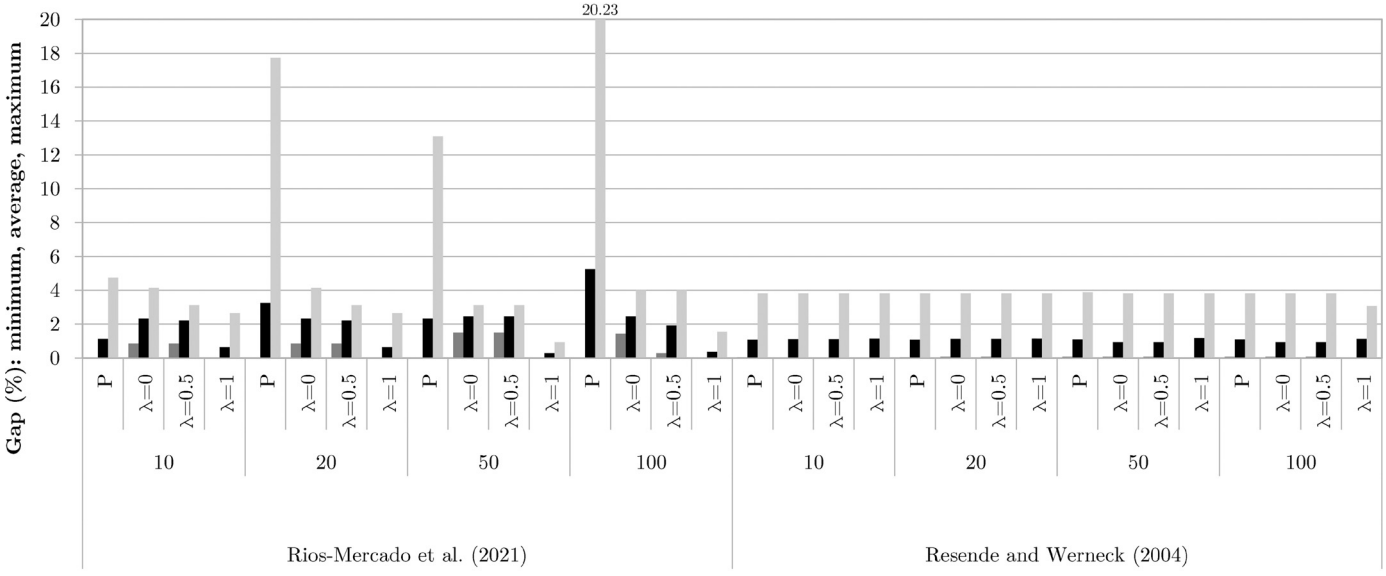
For simulating a territory that is to be divided into districts, we consider a so-called Gabriel graph (Gabriel & Sokal 1969). This is a graph induced by a set of points in the Euclidean plane such that two nodes are adjacent if and only if they are the two closest nodes their midpoint. The nodes were randomly generated in a square $[0, 1000] \times [0, 1000]$ according to a uniform distribution. Each node is then associated with a TU and thus generates some demand.

The demands of the TUs were randomly generated according to a Uniform distribution in an interval $[a, b]$. The extreme points of that interval were set in such a way that the expected value is the same but different standard deviations are considered. In particular, the expected value was fixed equal to 50 and different values were considered for the relative standard deviation (RSD). Recall that, for a random variable d, its RSD is given by $\frac{\sqrt{\mathbb{V}ar[d]}}{\mathbb{E}[d]}$. In the case of a uniformly distributed random variable d in interval $[a, b]$ it is easy to conclude that if we are given $\mathbb{E}[d]$ and RSD$[d]$, then $a = \mathbb{E}[d] \cdot (1 - \sqrt{3} \cdot \text{RSD}[d])$ and $b = \mathbb{E}[d] \cdot (1 + \sqrt{3} \cdot \text{RSD}[d])$. In Table 1 we detail the values considered for the different parameters inducing an instance.

In total we obtained 324 instances, which are available to anyone interested upon writing to the authors.



(a) $|\Omega| = 10$.



(b) $|\Omega| = 100$.

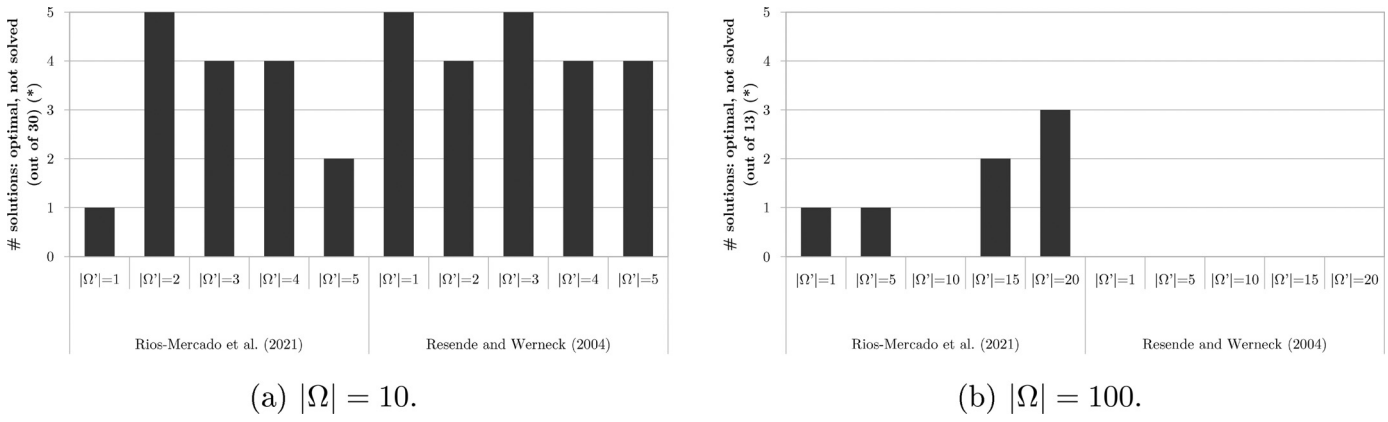
Fig. 2. Procedure 2—percentage gap, $|N| = 100$.

5.2. Experimental setting

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 procedures were implemented in Python 3.6. The required optimization models were solved using IBM ILOG CPLEX 12.10.

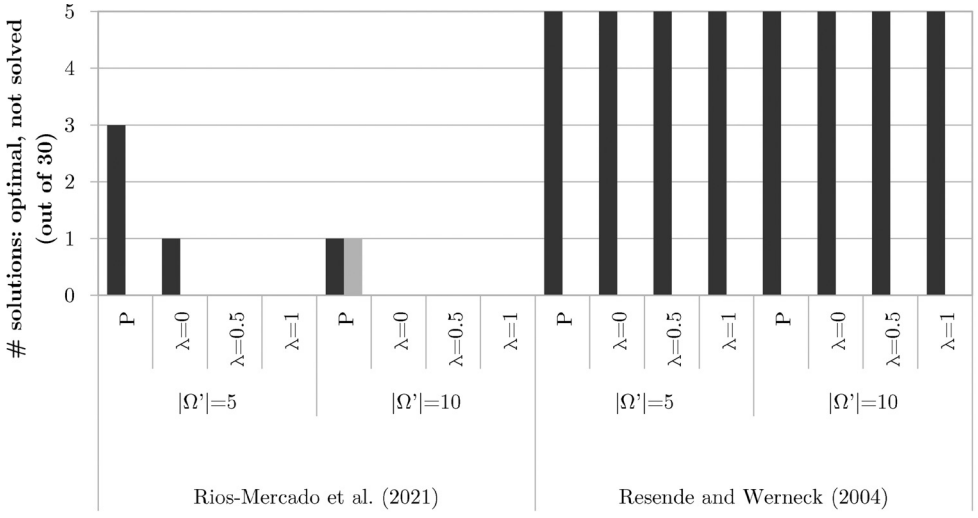
When solving a problem by using the row generation procedure proposed by Salazar-Aguilar et al. (2011) (Algorithm 2, lines 4 and 9; Algorithm 3, lines 6 and 19) we allow the whole row generation procedure to run for a maximum time limit, say `globalTimeLimit`. Additionally, in that same algorithm, each integer problem (IP) is executed by the solver for a maximum time limit, say `localTimeLimit`. Finally, when solving an IP we consider a termination gap, say `IPgap`. Recall that every time a new IP is solved, we first check if the solution is contiguous. If this is the case, we stop. Otherwise, we solve the separation problem and add new cuts.[1] We also update the total elapsed time and exit the

[1] In practice, we start running a relaxed version of the model, i.e., without explicit contiguity requirements. Clearly, the resulting solution may be not contiguous. It has to be noted that each solution induces a p-partition of the underlying graph (corresponding to the p obtained districts). Hence, we should check if each partition contains some connected components (that is, groups of connected Territorial Units) that are not connected to the center (or representative) of their district. We implemented a Depth First Search algorithm to this end. Note that this is, in fact, the "separation" problem mentioned in the text. If that is the case, each of these connected components induces a violated constraint, in the form of inequalities (5), where the role of set S is played by the connected component itself. The model is then resolved with the addition of these new inequalities ("the added cuts"), until the solution is proven to be contiguous. For the sake of brevity, we cannot elab-

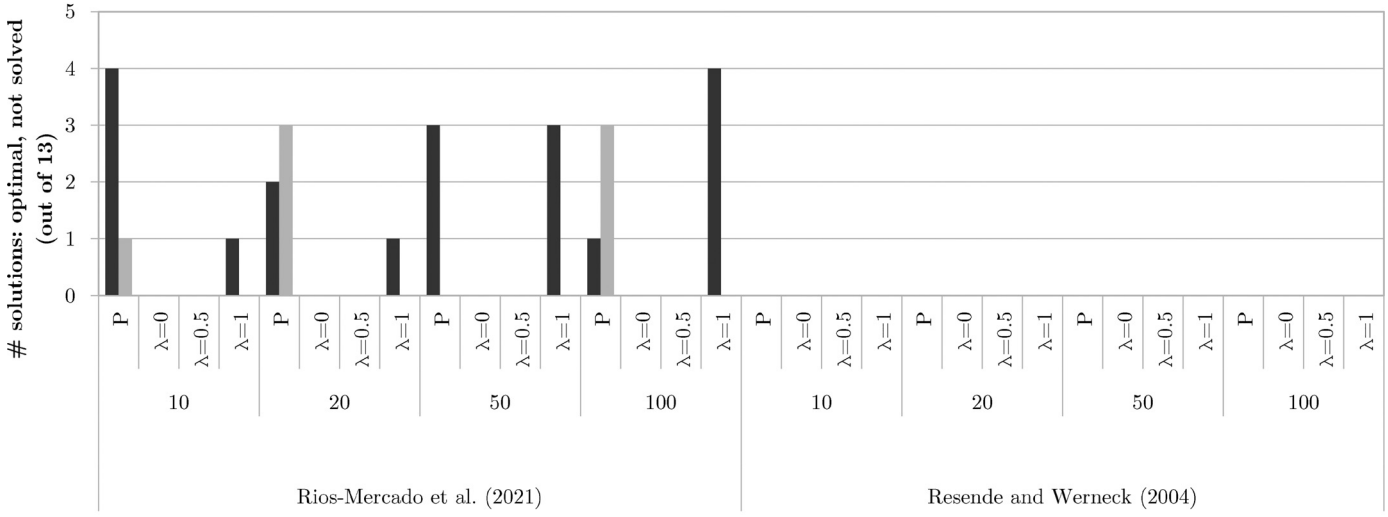


(a) $|\Omega| = 10$. (b) $|\Omega| = 100$.

Fig. 3. Procedure 1—number of optimal solutions found, $|N| = 100$. (*) number of not solved omitted since they are all equal to zero.



(a) $|\Omega| = 10$.



(b) $|\Omega| = 100$.

Fig. 4. Procedure 2—number of optimal solutions found (darker bars), and not solved instances, $|N| = 100$.

Table 1
Values considered for different parameters.

parameter	value
$\|N\|$	100, 500, 1000
p	6, 8 (for $\|N\| = 100$); 10, 20 (for $\|N\| = 500$); 20, 40 (for $\|N\| = 1000$)
RSD	0.125, 0.250, 0.500
α	0.1, 0.2
γ	0.7, 0.8, 0.9
$\|\Omega\|$	10, 100, 1000

Table 2
CPU time (seconds) required to solve the problem up to optimality, $|N| = 100$, $|\Omega| = 10$.

	$p = 6$						$p = 8$					
	$\alpha = 0.1$			$\alpha = 0.2$			$\alpha = 0.1$			$\alpha = 0.2$		
RSD	$\gamma = 0.7$	$\gamma = 0.8$	$\gamma = 0.9$	$\gamma = 0.7$	$\gamma = 0.8$	$\gamma = 0.9$	$\gamma = 0.7$	$\gamma = 0.8$	$\gamma = 0.9$	$\gamma = 0.7$	$\gamma = 0.8$	$\gamma = 0.9$
0.125	76.5	58.9	13.9	18.6	8.4	6.7	26.8	10.7	4.6	5.2	5.5	3.5
0.250	250.9	212.2	12.2	30.8	22.9	7.7	50.6	2411.6	1540.7	6.7	7.9	4.9
0.500	t.l.	t.l.	t.l.	81.4	38.7	6.0	t.l.	t.l.	t.l.	85.1	277.3	271.0
# solved	2	2	2	3	3	3	2	2	2	3	3	3
Avg time	163.71	135.53	13.02	43.61	23.32	6.78	38.72	1211.14	772.64	32.36	96.9	93.13

Table 3
CPU time (seconds) required to solve the problem up to optimality, $|N| = 100$, $|\Omega| = 100$.

	$p = 6$						$p = 8$					
	$\alpha = 0.1$			$\alpha = 0.2$			$\alpha = 0.1$			$\alpha = 0.2$		
RSD	$\gamma = 0.7$	$\gamma = 0.8$	$\gamma = 0.9$	$\gamma = 0.7$	$\gamma = 0.8$	$\gamma = 0.9$	$\gamma = 0.7$	$\gamma = 0.8$	$\gamma = 0.9$	$\gamma = 0.7$	$\gamma = 0.8$	$\gamma = 0.9$
0.125	t.l.	t.l.	963.1	205.2	282.8	275.8	237.4	1331.8	t.l.	84.6	67.1	43.8
0.250	t.l.	t.l.	t.l.	t.l.	t.l.	387.1	t.l.	t.l.	t.l.	103.5	81.3	277.3
0.500	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.	t.l.
# solved	0	0	1	1	1	2	1	1	0	2	2	2
Avg time	N/A	N/A	963.1	205.2	282.8	331.5	237.4	1331.8	N/A	94.0	74.2	160.6

solution process if it exceeds the `globalTimeLimit`. Otherwise, we keep on iterating. In our experiments, we set `IPgap` = 0.01, and `globalTimeLimit` = `localTimeLimit` = 1000 seconds. By doing so, when the fully relaxed problem (no contiguity constraints) is already hard to solve, we give up solving that particular instance.

When testing our solutions procedure, different possibilities were considered in terms of the sub-sample size Ω' in Algoritm 2, and concerning the number of single-scenario problems m to solve in Algorithm 3. Since these parameters were tuned based upon some first sets of experiments, more details are provided in the next sections.

Finally, concerning Procedure 2, four different attractiveness measures were investigated: the probabilistic one, as well as the merit attractiveness measure considering $\lambda \in \{0, 0.5, 1\}$. Next we present the results structured according to the aspects we are focusing.

5.3. First results: Exact solutions—small instances

A first set of tests was conducted to find the exact solution to our problem so that we can later assess, up to some extent, the quality of the heuristics proposed. This set of tests could be accomplished using only the smaller instances, namely the 72 instances with $|N| = 100$, $|\Omega| = 10$, $|\Omega| = 100$. In particular, we underline we solved model (M_2), consisting of minimizing (1), subject to (2)–(3), (5)–(9), (11)–(12). The results are presented in Table 2 for $|\Omega| = 10$ and in Table 3 for $|\Omega| = 100$.

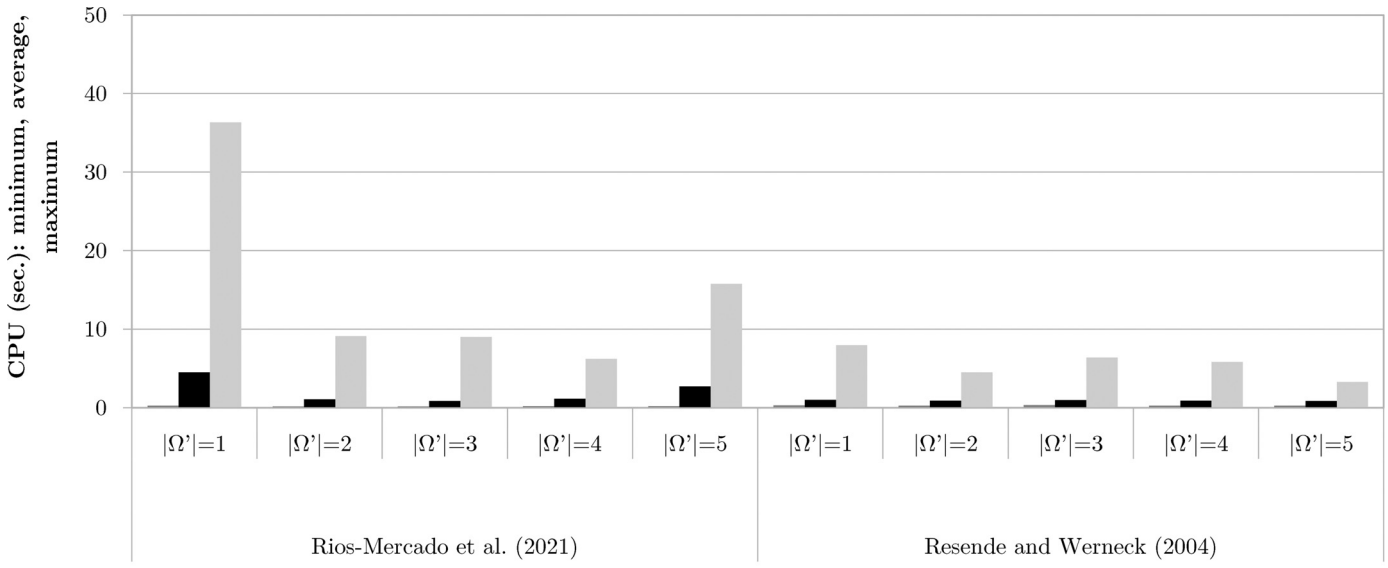
In these four-dimensional tables corresponding to (p, α, γ, RSD), we present the computing time in seconds required to solve each instance. We also present the number of instances solved as well as the average computing time for each value of p, α and γ (last two rows). In these tables, "t.l." indicates that the time limit imposed was reached and, thus, the solver stopped prematurely; finally, "N/A" stands for not applicable.

Observing these tables we conclude that the larger the RSD the more challenging is to solve the problem up to proven optimality. This is an indication that more variability makes it more difficult to find a districting solution satisfying the probabilistic constraints. This may be due to a larger difficulty in finding a trade-off when demands fluctuate more. Furthermore, when α increases from 0.1 to 0.2, more instances can be solved within the time limit. This is expected since a larger α corresponds to a larger flexibility for calling a solution balanced. No major difference can be devised in the results for $p = 6$ and $p = 8$. Overall, we conclude that for $|\Omega| = 10$, 30 out of 36 instances were solved up to proven optimality whereas this number drops to 13 out of 36 for $|\Omega| = 100$. It should be pointed out that all attempts made to solve up to optimality instances with 1000 scenarios (keeping $|N| = 100$) failed, which explains why no results are presented in that case.

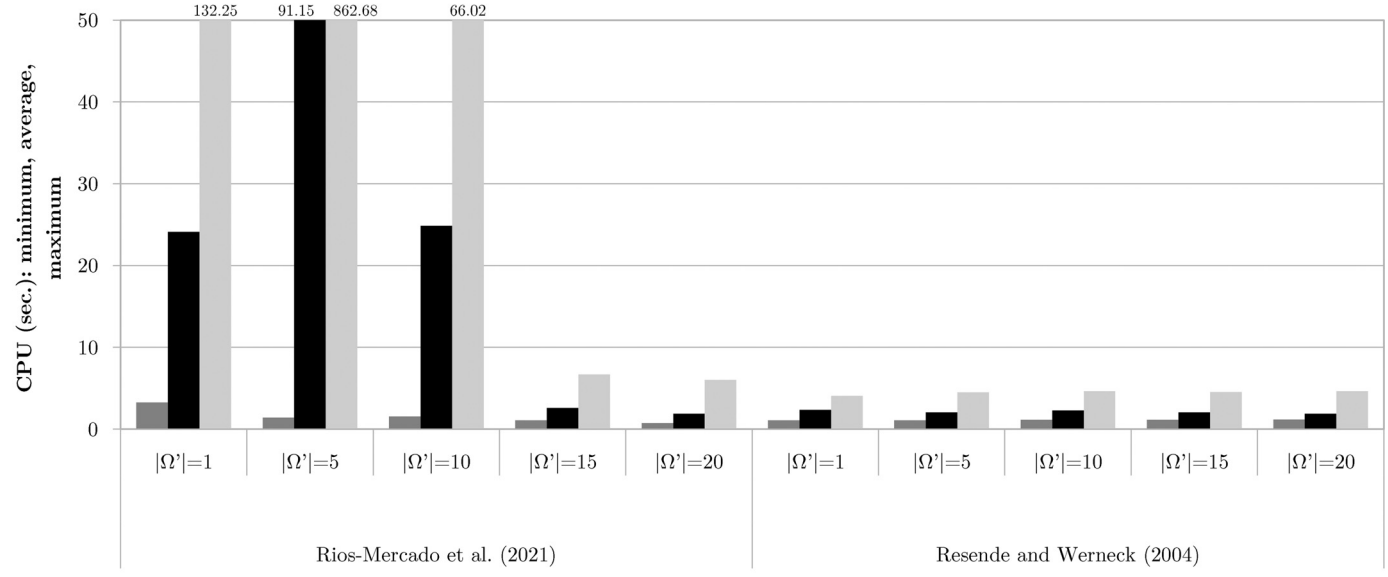
5.4. Assessing the quality of the two heuristic procedures

The next step in our computational experience was to assess the quality of the heuristic solutions obtained by our proposed Procedures 1 and 2. Again, we consider the instances with $|N| = 100$ TUs. As we observed in the previous section, in this case, 30 and 13 instances could be solved up to proven optimality for $|\Omega| = 10$ and $|\Omega| = 100$, respectively.

orate much on this aspect. Hence, we refer the interested readers to the work by Salazar-Aguilar et al. (2011) for more detailed explanations.



(a) $|\Omega| = 10$.



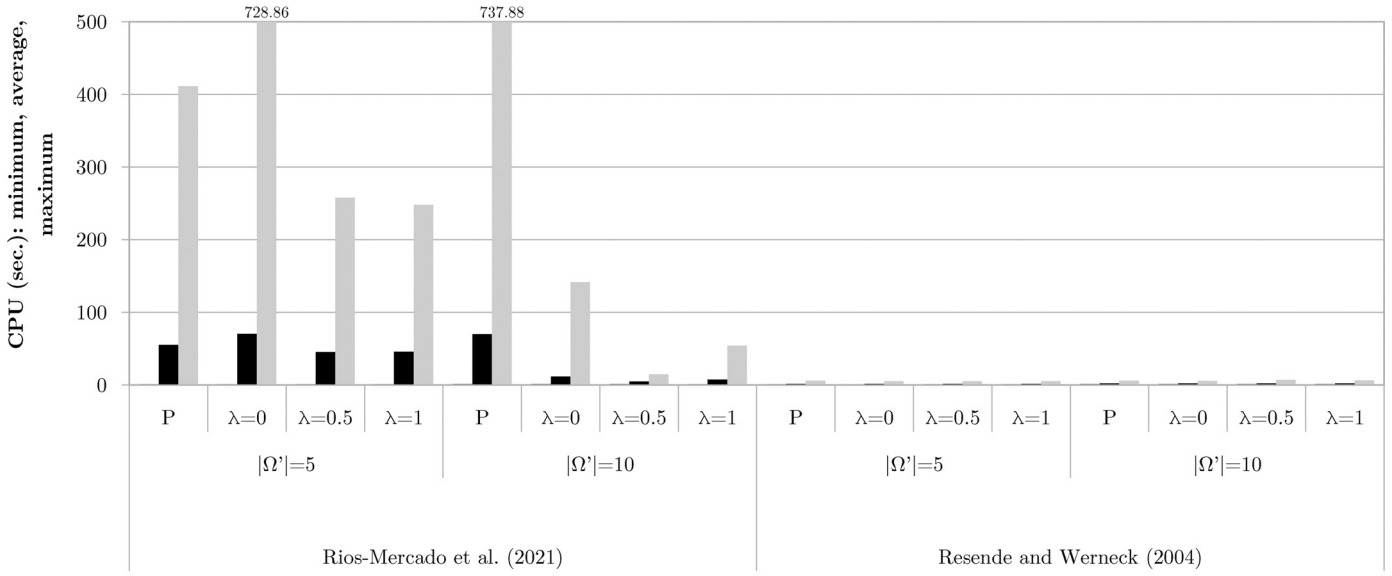
(b) $|\Omega| = 100$.

Fig. 5. Procedure 1—CPU time (seconds), $|N| = 100$.

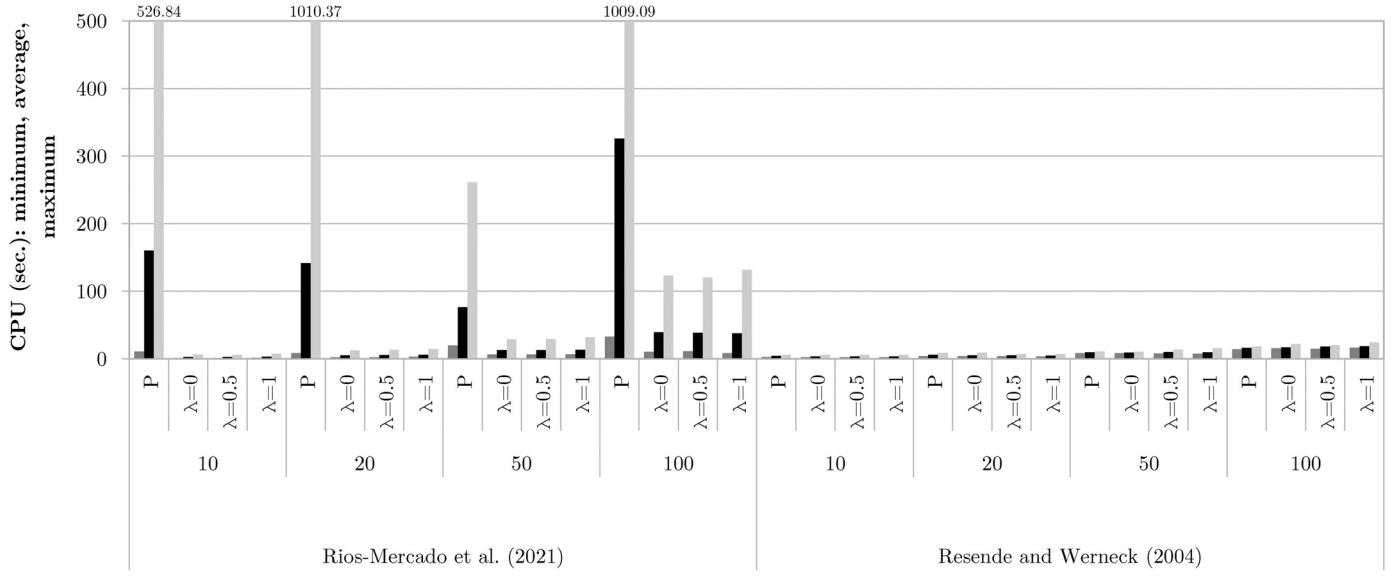
We recall that our heuristic procedures call for an initial selection of p centers when a restricted model is considered. As explained before, for this purpose, two possibilities were investigated: the algorithm by Resende & Werneck (2004) or the one by Ríos-Mercado et al. (2021).

For testing our first procedure (Algorithm 2) on instances with $|\Omega| = 10$, five possibilities were considered in terms of the sub-sample Ω' size: $|\Omega'| \in \{1, 2, 3, 4, 5\}$. For the second procedure, the decision concerns the number of single-scenario problems to solve in Algorithm 3. We tested two possibilities: $|\Omega'| = \{5, 10\}$. We use the same notation (i.e., Ω') for both procedures since, in fact, it corresponds to the same idea: a sub-sample drawn from the underlying set of scenarios. However, in the case of Procedure 1, such sub-sample is used to built a model involving all the scenarios in the sub-sample, whereas in Procedure 2, each scenario in the sub-sample is used to formulate a single scenario model. For instances with $|\Omega| = 100$, we set: $|\Omega'| \in \{1, 5, 10, 15, 20\}$ in Procedure 1, and $|\Omega'| \in \{10, 20, 50, 100\}$ in Procedure 2. In our results, we write "P" to specify the use of the probabilistic attractiveness measure used in Procedure 2. The merit function, instead, is indicated by the particular value set for the weighting factor λ ($\lambda \in \{0, 0.5, 1\}$). We underline that these experiments also aimed at fine-tuning the above mentioned parameters for the extensive tests discussed in the next section.

In Fig. 1 we can see the percentage gap of the solutions obtained by Procedure 1. Figure 1a contains the results for the instances with 10 scenarios, whereas Fig. 1b refers to those with 100 scenarios. Figure 2 depicts the results for Procedure 2. Observing these two figures, namely the darker bars (corresponding to average values), we see that for the instances that could be solved up to proven optimality, the relative gaps of the solutions provided by our two procedures are fairly small. Furthermore, the re-



(a) $|\Omega| = 10$.



(b) $|\Omega| = 100$.

Fig. 6. Procedure 2—CPU time (seconds), $|N| = 100$.

sults indicate that the use of the heuristic by Resende & Werneck (2004) when finding initial sets of p centers leads to a more robust behavior of both procedures. In fact, the maximum gaps are clearly smaller in this case.

A supplementary information to the gaps is the number of instance for which our procedures could find an optimal solution. These results are depicted in Figs. 3 and 4. Again, each figure contains two subfigues: one for $|\Omega'| = 10$ and another for $|\Omega'| = 100$. Other than the number of optimal solutions (darker bars), these figures also report on the number of cases the heuristics failed finding a feasible solution to the problem. As expected, when $|\Omega|$ increases, the problem becomes more challenging and the heuristics find it more difficult to find optimal solutions. Nevertheless, for 10 scenarios we see that, often, the heuristics end up with an optimal solution. Noticeably, when the number of scenarios increase, it is still possible to find optimal solutions when the procedure by Ríos-Mercado et al. (2021) is embedded in the matheuristics. Besides, we also point out that the use of a probabilistic attractiveness measure does not appear to be a very good choice when embedded in Procedure 2. Indeed, this procedure, using the constructive method by Ríos-Mercado et al. (2021), fails in one and seven cases, for $|\Omega'| = 10$ and $|\Omega'| = 100$, respectively. Considering that we had an optimal solution to the problem in such cases, we decided to avoid continuing using the probabilistic attractiveness measure in our extensive experiments (see next section).

Finally, we consider the computing times for the small instances that we are analyzing. In Figs. 5 and 6, we observe that the heuristics run in very limited computing times, especially when the constructive method by Resende & Werneck (2004) is used. The latter seems to guarantee a more stable behavior also in this perspective. Higher computing times are noted when we employ the constructor by Ríos-Mercado et al. (2021), which, however, are

Table 4
Intensive testing: number of experiments performed.

$\|\Omega\| = 10$										
Procedure	Constructor	$\|\Omega'\|$			λ			# experiments		
1	(R)	5						36		
	(M)	2	5					72		
2	(R)	5	10		0	0.5	1	216		
	(M)	5	10		0	0.5	1	216		
						Total:		540		
$\|\Omega\| = 100$										
Procedure	Constructor	$\|\Omega'\|$				λ			# experiments	
1	(R)	10	20						72	
	(M)	10	20						72	
2	(R)	10	20	50	100	0	0.5	1	432	
	(M)	10	20	50	100	0	0.5	1	432	
								Total:	1008	
$\|\Omega\| = 1000$										
Procedure	Constructor	$\|\Omega'\|$				λ			# experiments	
1	(R)	10	20						72	
	(M)	10	20						72	
2	(R)	10	20	50	100	0	0.5	1	432	
	(M)	10	20	50	100	0	0.5	1	432	
								Total:	1008	

fairly limited in average terms. A few rare occurrences of very high computational times are observed when employing Procedure 2, using the probabilistic attractiveness measure (P). This is another indication that such a measure does not constitute a promising choice for our heuristic setting.

For the sake of a clearer visualization, in Figs. 1–6, the maximum values we set on the y-axis do not always correspond to the actual maximum of the displayed indicators. Extreme values are therefore explicitly reported on top of each bar. Note that the detailed results underlying Figs. 1–6 can be found in the electronic supplement, Appendix A. Tables A1–A4 concern Procedure 1, while Tables A5–A12 refer to Procedure 2.

5.5. Extensive experiments

Given the preliminary analysis performed in the previous sections, we report now on an intensive series of tests performed with the new procedures. For this purpose, recall that given a number of TUs and a number of scenarios, we have 36 instances (two values of p, two values of α, three values of RSD, and three values of γ).

In Table 4 we summarize the experiments performed for each number of TUs (i.e., $|N|$) considered. All the values for the calibration parameters of the heuristics were chosen based on their performance in the set of computational experiments presented in the previous section (see also Appendix A for further details). Note, in fact, that we do not make use of the probabilistic attractiveness measure in these tests, because of the poor results above discussed. For the sake of brevity, we indicate in the tables and in the computational results by (M) and (R) the use of the constructive methods by Ríos-Mercado et al. (2021) and Resende & Werneck (2004), respectively. Recall that in Procedure 1, a single restricted problem is solved considering $|\Omega'|$ scenarios, whereas in Procedure 2, $|\Omega'|$ single-scenario problems are solved.

5.5.1. Setting the time limit for solving the single-scenario problems in Procedure 2

An important decision to make when testing Procedure 2 is the time limit to impose when solving each single-scenario problem. Since a single value of γ makes sense in the single scenario problem ($\gamma = 1$), for a certain number of nodes $|N|$, our data induces 12 instances (two values of p, two values of α, and three values of RSD). To assess the time limit that should be set we considered such 12 instances for each $|N|$. Then, we considered ten different random seeds for generating the demand. This means that we ended up with 120 instances for each value of $|N|$. In turn, each of such instance is tackled using the two constructors. Hence, we have results for 240 experiments: 120 using constructor (M) and 120 using constructor (R). These results are reported in Appendix B (Table B1), where we show the number of instances (absolute values and percentages) solved within a certain time for the different dimensions considered: $|N| = 100, 500$, and 1000.

Given the results in that table, we decided to not set any time limit for the instances with 100 TUs and we set a time limit of 10 and 30 seconds for the instances with 500 and 1000 TUs, respectively.

5.5.2. Results

We finally report on the results obtained for the extended tests in terms of the computing time required by the procedures we are testing. The synthesis of the obtained results can be found in Table 5, where we present average values for the instances tackled successfully by our new procedures within the time limit imposed. The detailed results can be found in the Electronic Supplement, Appendix B, Tables B2–B19.

In Table 5 we observe solutions found within the time limit in many cases, with the averages reported being quite acceptable given the type of problem being tackled. No clear different can be foreseen neither between both procedures nor between the use of different values of λ for Procedure 2. Furthermore, no clear difference can be devised when changing the constructor from (M) to (R). For $|N| = 1000$ and $|\Omega| = 1000$, using Procedure 1 with the constructive method by Ríos-Mercado et al. (2021), we note that no instance corresponding to $|\Omega'| = 20$ could be solved within the time limit. The same held for Procedure 2 (with the same constructor) when $|\Omega'| = 10$ and $\lambda = 1$. Indeed, the major difference we can highlight is in the number of solved instances based on the specific constructor used (see Appendix B for details). This fact is not particular relevant for smaller instances, namely, with $|N| = 100$. In this case, both the procedures, regardless of the constructor used, led to a feasible solution in 31 out of 36 cases for $|\Omega| = 10$, in 19

Table 5
Average computing time (seconds).

(a) Results—Initial centers (M).

		Procedure 1		Procedure 2											
				$\lambda = 0$				$\lambda = 0.5$				$\lambda = 1$			
$\|\Omega\|$	$\|N\|$	$\|\Omega'\| = 2$	$\|\Omega'\| = 5$	$\|\Omega'\| = 5$	$\|\Omega'\| = 10$			$\|\Omega'\| = 5$	$\|\Omega'\| = 10$			$\|\Omega'\| = 5$	$\|\Omega'\| = 10$		
10	100	2.50	2.74	63.84	55.37			37.83	36.83			38.30	36.98		
	500	31.98	69.43	70.74	75.68			34.81	42.23			34.76	41.11		
	1000	283.02	529.48	477.82	498.38			442.89	462.35			427.98	458.44		
		$\|\Omega'\| = 10$	$\|\Omega'\| = 20$	$\|\Omega'\| = 10$	$\|\Omega'\| = 20$	$\|\Omega'\| = 50$	$\|\Omega'\| = 100$	$\|\Omega'\| = 10$	$\|\Omega'\| = 20$	$\|\Omega'\| = 50$	$\|\Omega'\| = 100$	$\|\Omega'\| = 10$	$\|\Omega'\| = 20$	$\|\Omega'\| = 50$	$\|\Omega'\| = 100$
100	100	24.92	11.89	62.13	64.10	70.67	141.13	62.61	65.72	70.42	130.37	20.21	32.07	56.78	97.15
	500	242.91	424.49	147.13	189.18	247.73	397.06	176.60	187.09	254.65	387.78	182.33	212.56	269.03	405.89
	1000	1429.03	1465.77	1108.56	1472.15	2284.27	2667.65	1138.69	1361.08	2309.17	2700.65	1051.45	1415.79	2364.00	2724.89
		$\|\Omega'\| = 10$	$\|\Omega'\| = 20$	$\|\Omega'\| = 10$	$\|\Omega'\| = 20$	$\|\Omega'\| = 50$	$\|\Omega'\| = 100$	$\|\Omega'\| = 10$	$\|\Omega'\| = 20$	$\|\Omega'\| = 50$	$\|\Omega'\| = 100$	$\|\Omega'\| = 10$	$\|\Omega'\| = 20$	$\|\Omega'\| = 50$	$\|\Omega'\| = 100$
1000	100	687.50	138.16	175.46	125.15	118.91	334.05	175.12	124.81	156.16	327.43	205.90	252.60	255.73	423.30
	500	433.02	1068.32	371.09	373.28	575.86	711.02	308.88	371.02	520.40	720.49	765.74	581.81	1037.45	927.73
	1000	1063.96	—	1335.48	1746.81	2245.62	3372.53	1328.47	1971.56	2451.16	3116.19	—	1812.05	1605.78	2603.77

(b) Results—Initial centers (R).

		Procedure 1		Procedure 2											
				$\lambda = 0$				$\lambda = 0.5$				$\lambda = 1$			
$\|\Omega\|$	$\|N\|$		$\|\Omega'\| = 5$	$\|\Omega'\| = 5$	$\|\Omega'\| = 10$			$\|\Omega'\| = 5$	$\|\Omega'\| = 10$			$\|\Omega'\| = 5$	$\|\Omega'\| = 10$		
10	100		1.09	2.43	3.21			2.38	2.78			1.90	2.83		
	500		11.55	24.35	37.19			23.10	36.20			22.56	36.38		
	1000		297.62	299.07	367.50			269.35	329.62			272.45	323.82		
		$\|\Omega'\| = 10$	$\|\Omega'\| = 20$	$\|\Omega'\| = 10$	$\|\Omega'\| = 20$	$\|\Omega'\| = 50$	$\|\Omega'\| = 100$	$\|\Omega'\| = 10$	$\|\Omega'\| = 20$	$\|\Omega'\| = 50$	$\|\Omega'\| = 100$	$\|\Omega'\| = 10$	$\|\Omega'\| = 20$	$\|\Omega'\| = 50$	$\|\Omega'\| = 100$
100	100	12.72	11.71	15.09	18.34	21.40	28.88	14.22	17.35	21.42	29.52	14.26	16.38	19.33	27.57
	500	59.64	58.39	89.47	107.20	195.84	340.34	84.35	104.25	195.94	343.56	96.47	120.47	198.95	337.66
	1000	553.49	603.80	391.03	490.43	874.98	1310.12	449.37	542.90	686.87	1362.42	411.54	485.13	633.21	1150.35
		$\|\Omega'\| = 10$	$\|\Omega'\| = 20$	$\|\Omega'\| = 10$	$\|\Omega'\| = 20$	$\|\Omega'\| = 50$	$\|\Omega'\| = 100$	$\|\Omega'\| = 10$	$\|\Omega'\| = 20$	$\|\Omega'\| = 50$	$\|\Omega'\| = 100$	$\|\Omega'\| = 10$	$\|\Omega'\| = 20$	$\|\Omega'\| = 50$	$\|\Omega'\| = 100$
1000	100	287.17	274.17	234.84	234.89	208.09	209.56	232.39	226.15	206.59	209.65	242.17	233.60	370.44	444.17
	500	502.04	494.23	538.32	531.68	840.27	998.86	503.11	531.46	809.21	1003.78	407.94	426.46	511.91	748.31
	1000	1034.96	1268.24	1790.38	1722.62	1840.51	2377.52	1535.81	1807.68	1874.58	2380.46	1536.38	1352.07	2090.02	2264.11

Table 6
Benchmarking the new heuristic.

					Computing time (seconds)		Deviation (%)
$\|N\|$	RSD	p	α	γ	New heuristic	Existing heuristic	
100	0.125	6	0.1	0.7	1.32	0.65	5.71
100	0.125	6	0.1	0.8	17.34	0.64	8.45
100	0.125	6	0.1	0.9	16.29	0.70	8.45
100	0.125	6	0.2	0.7	7.45	0.42	5.06
100	0.125	6	0.2	0.8	6.87	0.41	5.06
100	0.125	6	0.2	0.9	0.83	2.00	4.88
100	0.125	8	0.1	0.7	72.03	0.60	2.70
100	0.125	8	0.2	0.7	0.28	0.56	3.07
100	0.125	8	0.2	0.8	0.28	0.58	3.07
100	0.125	8	0.2	0.9	0.22	0.59	3.07
100	0.25	6	0.2	0.7	10.66	0.73	5.20
100	0.25	6	0.2	0.8	35.30	0.78	4.51
100	0.25	6	0.2	0.9	74.79	0.79	5.63
100	0.25	8	0.2	0.7	27.70	0.75	3.12
100	0.25	8	0.2	0.8	1.10	0.76	3.67
100	0.25	8	0.2	0.9	31.12	0.84	3.44
500	0.125	10	0.2	0.7	5.97	6.95	0.00
500	0.125	10	0.2	0.8	8.15	6.45	0.00
500	0.125	10	0.2	0.9	187.28	7.24	0.02
500	0.125	20	0.1	0.7	317.46	67.13	1.85
500	0.125	20	0.1	0.8	223.61	120.80	2.02
500	0.125	20	0.1	0.9	196.94	126.87	1.92
500	0.125	20	0.2	0.7	33.08	18.11	0.57
500	0.125	20	0.2	0.8	22.33	16.53	0.54
500	0.125	20	0.2	0.9	32.19	18.29	0.59
500	0.25	10	0.1	0.7	121.10	37.84	1.20
500	0.25	10	0.1	0.8	119.25	40.34	1.14
500	0.25	10	0.1	0.9	312.84	67.80	-1.64
500	0.25	10	0.2	0.7	205.45	7.11	0.00
500	0.25	10	0.2	0.8	27.78	7.45	0.03
500	0.25	10	0.2	0.9	9.25	7.60	-0.03
500	0.25	20	0.2	0.7	109.24	23.69	0.47
500	0.25	20	0.2	0.8	291.17	27.52	0.68
500	0.25	20	0.2	0.9	89.38	29.91	1.49
500	0.5	10	0.2	0.7	620.48	22.27	0.37
500	0.5	10	0.2	0.8	1344.69	28.10	0.39
500	0.5	10	0.2	0.9	94.41	36.03	-4.37
1000	0.125	20	0.1	0.7	481.11	150.42	0.24
1000	0.125	20	0.1	0.8	120.18	161.75	0.34
1000	0.125	20	0.1	0.9	119.13	172.63	0.34
1000	0.125	20	0.2	0.7	59.48	62.30	0.01
1000	0.125	20	0.2	0.8	121.75	62.74	0.06
1000	0.125	20	0.2	0.9	45.14	69.42	0.07
1000	0.25	20	0.1	0.7	204.57	160.98	0.05
1000	0.25	20	0.2	0.7	804.13	69.65	0.11
1000	0.25	20	0.2	0.8	1201.95	77.51	0.13
1000	0.25	20	0.2	0.9	1258.56	86.38	0.11
1000	0.5	20	0.2	0.7	666.25	127.09	0.07

instances for $|\Omega| = 100$, and 17 instances for $|\Omega| = 1000$. Again, no major differences are noted for $|N| = 500$ and $|\Omega| = 10$, where 33 out of 36 instances were solved irrespective of the adopted heuristic setting. However, when $|\Omega| = 100$, we observe that Procedure 1 with the constructive method by Ríos-Mercado et al. (2021) tackled successfully only 17 and 15 instances for $|\Omega'| = 10, 20$, respectively. Instead, 24 instances were solved under the same settings using the constructor by Resende & Werneck (2004). More similar performance between the constructors are found when using Procedure 2. Similar evidences are observed when $|\Omega| = 1000$. In this case, the highest number of solved instances equals 16.

When focusing on $|N| = 1000$ and $|\Omega| = 10$, we highlight the superiority of the method induced by embedding the constructor by Resende & Werneck (2004). In this case, we were able to solve 33 instances with Procedure 1, and at least 30 with Procedure 2. Regardless of the solution procedure, the constructor by Ríos-Mercado et al. (2021) allowed solving at most 27 instances. Similar results are found in the other cases: indeed, Procedure 1, with the method by Resende & Werneck (2004), solved up to 22 and 8 instances for $|\Omega| = 100$ and 1000, respectively.

As a final note, one may wonder about which procedure and setting should be adopted for solving our problem. Our results show a strong robustness of the location-allocation schemes both in terms of the constructor used and in terms of the attractiveness measure to adopt. In our opinion, this is an indication of a mechanism that naturally suits this type of problem. While no major differences are devised in terms of computational times, we should note that the constructor by Resende & Werneck (2004) seems to be more effective in solving a higher number of instances, especially if embedded within Procedure 1. However, when more feasible solutions are found for a particular instance, none of the heuristics is clearly dominant in terms of solutions quality (i.e., objective function). Hence, we can conclude that the proposed procedures can be considered competitive, allowing to produce a variety of good quality solutions at an acceptable computational effort even for larger-sized instances.

5.6. Comparison with an alternative heuristic

As mentioned before, our problem is the same that was investigated by Diglio et al. (2021). Hence, it can be tackled by the heuristic proposed in that work. Regarding the new heuristic we are proposing in this paper, it is based upon an approximate model, contrary to the heuristic by Diglio et al. (2021). In particular, the latter uses a mechanism that computes exactly the probability of a solution to be balanced, which essentially resorts to a Monte-Carlo simulation using 1000 demand scenarios. In our case, the final solution obtained may end up not ensuring the minimum desired balancing probability. In fact, we recall that model (M_2) is an approximate deterministic counterpart to model (M_1), meaning that, in practice, there is no guarantee that a feasible solution to model (M_2) meets the balancing requirements with the probability γ. This is especially true if the number of scenarios ($|\Omega|$) considered in model (M_2) is relatively small. Thus, after finding a solution using our heuristic based upon the approximate model, we use the same mechanism as in Diglio et al. (2021) for evaluating the probability that a solution is balanced. In practice, we perform an ex-post simulation on each solution found using the various settings of our heuristics, relying upon a brand-new sample of values for the demands (1000 new scenarios). The seed for generating the random numbers was randomized only once—at the beginning of the process. Then, we consider the best solution, in terms of the objective function, among the set of "feasible" ones (i.e., those that met the required probability γ in the ex-post simulation) to compare against the approach proposed by Diglio et al. (2021).

Results about a competitive comparison are presented in Table 6. In this table the deviation gap is computed as $100\frac{\mathcal{V}(\text{Existing}_{\text{heur}})-\mathcal{V}(\text{New}_{\text{heur}})}{\mathcal{V}(\text{New}_{\text{heur}})}$, with $\mathcal{V}(\text{Existing}_{\text{heur}})$ and $\mathcal{V}(\text{New}_{\text{heur}})$ denoting the objective function value of the solutions obtained by the heuristic from Diglio et al. (2021) and the best solution obtained by the procedures proposed in this work, respectively. Thus, a positive deviation indicates that the heuristic proposed in this work outperforms the existing one.

The major conclusions drawn from this table are clear. First, the new heuristic procedure outperforms the existing one for, at least, the instances in which a fair comparison can be performed. Of course, one may argue that the reported relative percentage gaps are relatively small, especially for the 1000-nodes instances. However, it has to be noted that we are talking about objective functions with very high values (millions, in absolute terms), which constitute a very high denominator in our relative gap computation. Second, this improvement comes at a cost since the computing time required by the new heuristic is often above the one required by the existing heuristic. Nevertheless, the difference is not substantial when one observes the involved values. In any case, we feel the need to point out that our goal in this work is to propose an approximate model for solving chance-constrained districting problems, which we tackle employing two general heuristics. This, in our opinion, has one major contribution, namely, stimulating scholars' effort and interest towards more efficient and effective solution approaches exploiting the proposed mathematical formulation.

6. Conclusions

In this paper we investigated a chance-constrained districting problem with explicit contiguity constraints. We derived an approximate deterministic counterpart model for which we developed a heuristic solution method with different variants.

The results show that the methodologies proposed solve (even if approximately) instances with up to 1000 territory units and 1000 scenarios. This is quite relevant since it allows tackling approximately, by means of large sampling, instances with a continuous cumulative distribution function for the random vector underlying the problem. For smaller instances, it was possible to assess the quality of the results produced by the new approximate algorithms, and to classify them as efficient and effective for the investigated problem. Also, the proposed procedures proved competitive w.r.t. to an existing heuristic in the literature.

The idea proposed in this paper for finding an approximate deterministic counterpart of the chance-constraint problem can be easily adapted to other types of facility location problems and, thus, this is clearly a research direction worth pursuing. Moreover, further effort towards the refinement of the heuristics can be made in order to solve larger sized instances, namely involving more TUs and —above all— a higher number of base scenarios, thus making it applicable to those instances of real world cases (health services, postal service collection/delivery, etc.) whose nature may involve such large cardinality of scenario set.

Acknowledgments

This work was supported by National Funding from FCT — Fundação para a Ciência 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 projects PGC2018-099428-B-100, PID2021-122344NB-I00, and RED2018-102363-T.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.ejor.2022.09.005.

References

Bender, M., Kalcsics, J., & Meyer, A. (2020). Districting for parcel delivery services–a two-stage solution approach and a real-world case study. *Omega, 96*, 102283.
Bender, M., Kalcsics, J., Nickel, S., & Pouls, M. (2018). A branch-and-price algorithm for the scheduling of customer visits in the context of multi-period service territory design. *European Journal of Operational Research, 269*(1), 382–396.
Benzarti, E., Sahin, E., & Dallery, Y. (2013). Operations management applied to home care services: Analysis of the districting problem. *Decision Support Systems, 55*(2), 587–598.
Bruno, G., Diglio, A., Melisi, A., & Piccolo, C. (2017a). A districting model to support the redesign process of Italian provinces. In A. Sforza, & C. Sterle (Eds.), *International conference on optimization and decision science* (pp. 245–256). Springer, Cham..
Bruno, G., Genovese, A., & Piccolo, C. (2017b). Territorial amalgamation decisions in local government: Models and a case study from Italy. *Socio-Economic Planning Sciences, 57*, 61–72.
Butsch, A., Kalcsics, J., & Laporte, G. (2014). Districting for arc routing. *INFORMS Journal on Computing, 26*(4), 809–824.
Camacho-Collados, M., & Liberatore, F. (2015). A decision support system for predictive police patrolling. *Decision Support Systems, 75*, 25–37.
Camacho-Collados, M., Liberatore, F., & Angulo, J. M. (2015). A multi-criteria police districting problem for the efficient and effective design of patrol sector. *European Journal of Operational Research, 246*(2), 674–684.
Carlsson, J. G. (2012). Dividing a territory among several vehicles. *INFORMS Journal on Computing, 24*(4), 565–577.
Caro, F., Shirabe, T., Guignard, M., & Weintraub, A. (2004). School redistricting: Embedding GIS tools with integer programming. *Journal of the Operational Research Society, 55*(8), 836–849.
Charnes, A., & Cooper, W. W. (1959). Chance-constrained programming. *Management science, 6*(1), 73–79.
Correia, I., & Saldanha-da-Gama, F. (2019). Facility location under uncertainty. In G. Laporte, S. Nickel, & F. Saldanha-da-Gama (Eds.), *Location science* (pp. 185–213). Springer International Publishing. chapter 8
De Assis, L. S., Franca, P. M., & Usberti, F. L. (2014). A redistricting problem applied to meter reading in power distribution networks. *Computers & Operations Research, 41*, 65–75.
Diglio, A., Nickel, S., & Saldanha-da-Gama, F. (2020). Towards a stochastic programming modeling framework for districting. *Annals of Operations Research, 292*, 248–285.
Diglio, A., Peiró, J., Piccolo, C., & Saldanha-da-Gama, F. (2021). Solutions for districting problems with chance-constrained balancing requirements. *Omega, 103*, 102430.

Dugošija, D., Savić, A., & Maksimović, Z. (2020). A new integer linear programming formulation for the problem of political districting. *Annals of Operations Research, 288*, 247–263.
Elizondo-Amaya, M., Rıos-Mercado, R. Z., Morton, D. P., & Kutanoglu, E. (2013). A tabu search approach for a territory design problem with stochastic demands. In *Spanish congress on informatics (CEDI)*.
Enayati, S., Özaltın, O. Y., & Mayorga, M. E. (2020). Designing ambulance service districts under uncertainty. In R. Z. Ríos-Mercado (Ed.), *Optimal districting and territory design* (pp. 153–170). Cham: Springer International Publishing.
Farughi, H., Tavana, M., Mostafayi, S., & Arteaga, F. J. S. (2019). A novel optimization model for designing compact, balanced, and contiguous healthcare districts. *Journal of the Operational Research Society*. https://doi.org/10.1080/01605682.2019.1621217.
Gabriel, K. R., & Sokal, R. R. (1969). A new statistical approach to geographic variation analysis. *Systematic Biology, 18*, 259–278.
Hess, S. W., Weaver, J. B., Siegfeldt, H. J., Whelan, J. N., & Zitlau, P. A. (1965). Nonpartisan political redistricting by computer. *Operations Research, 13*(6), 998–1006.
Juan, A. A., Faulin, J., Grasman, S. E., Rabe, M., & Figueira, G. (2015). A review of simheuristics: Extending metaheuristics to deal with stochastic combinatorial optimization problems. *Operations Research Perspectives, 2*, 62–72.
Kalcsics, J., & Ríos-Mercado, R. (2019). Districting problems. In G. Laporte, S. Nickel, & F. Saldanha-da-Gama (Eds.), *Location science* (pp. 705–743). Springer International Publishing. chapter 25
Kiya, F., & Davoudpour, H. (2012). Stochastic programming approach to re-designing a warehouse network under uncertainty. *Transportation Research Part E: Logistics and Transportation Review, 48*(5), 919–936.
Kleywegt, A. J., Shapiro, A., & Homem-de Mello, T. (2002). The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization, 12*(2), 479–502.
Kong, Y., Zhu, Y., & Wang, Y. (2019). A center-based modeling approach to solve the districting problem. *International Journal of Geographical Information Science, 33*, 368–384.
Konur, D., & Geunes, J. (2019). Integrated districting, fleet composition, and inventory planning for a multi-retailer distribution system. *Annals of Operations Research, 273*, 527–559.
Liu, H., Erdogan, A., Lin, R., & Tsao, H.-S. J. (2020). Mathematical models of political districting for more representative governments. *Computers & Industrial Engineering, 140*, 106265.
Luedtke, J., & Ahmed, S. (2008). A sample approximation approach for optimization with probabilistic constraints. *SIAM Journal on Optimization, 19*(2), 674–699.
Miller, B. L., & Wagner, H. M. (1965). Chance constrained programming with joint constraints. *Operations Research, 13*(6), 930–945.
Mostafayi Darmian, S., Fattahi, M., & Keyvanshokooh, E. (2021). An optimization-based approach for the healthcare districting under uncertainty. *Computers & Operations Research, 135*, 105425.
Mourão, M. C., Nunes, A. C., & Prins, C. (2009). Heuristic methods for the sectoring arc routing problem. *European Journal of Operational Research, 196*, 856–868.
Nemirovski, A. (2012). On safe tractable approximations of chance constraints. *European Journal of Operational Research, 219*(3), 707–718.
Nemirovski, A., & Shapiro, A. (2007). Convex approximations of chance constrained programs. *SIAM Journal on Optimization, 17*(4), 969–996.
Nikzad, E., Bashiri, M., & Abbasi, B. (2021). A matheuristic algorithm for stochastic home health care planning. *European Journal of Operational Research, 288*, 753–774.
Pagnoncelli, B. K., Ahmed, S., & Shapiro, A. (2009). Sample average approximation method for chance constrained programming: Theory and applications. *Journal of Optimization Theory and Applications, 142*(2), 399–416.
Resende, M. G., & Werneck, R. F. (2004). A hybrid heuristic for the p-median problem. *Journal of Heuristics, 10*, 59–88.
Ricca, F., Scozzari, A., & Simeone, B. (2013). Political districting: From classical models to recent approaches. *Annals of Operations Research, 204*(1), 271–299.
(2020). In R. Z. Ríos-Mercado (Ed.), *Optimal districting and territory design*. Cham: Springer International Publishing.
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. *Computers & Operations Research, 126*, 105106.
Ríos-Mercado, R. Z., & Bard, J. F. (2019). An exact algorithm for designing optimal districts in the collection of waste electric and electronic equipment through an improved reformulation. *European Journal of Operational Research, 276*, 259–271.
Ríos-Mercado, R. Z., & Escalante, H. J. (2016). GRASP with path relinking for commercial districting. *Expert Systems with Applications, 44*, 102–113.
Ríos-Mercado, R. Z., & Fernández, E. (2009). A reactive GRASP for a commercial territory design problem with multiple balancing requirements. *Computers & Operations Research, 36*, 755–776.
Salazar-Aguilar, M. A., Ríos-Mercado, R. Z., & Cabrera-Ríos, M. (2011). New models for commercial territory design. *Networks and Spatial Economics, 11*(3), 487–507.
Santoso, T., Ahmed, S., Goetschalckx, M., & Shapiro, A. (2005). A stochastic programming approach for supply chain network design under uncertainty. *European Journal of Operational Research, 167*(1), 96–115.
Schoepfle, O., & Church, R. (1991). A new network representation of a 'classic' school districting problem. *Socio-Economic Planning Sciences, 25*(3), 189–197.
Shirabe, T. (2005). A model of contiguity for spatial unit allocation. *Geographical Analysis, 37*, 2–16.
Shirabe, T. (2009). Districting modeling with exact contiguity constraints. *Environment and Planning B: Planning and Design, 36*, 1053–1066.
Solana, M. A., Díaz, J. A., & Luna, D. E. (2019). Math-heuristic for a territory design problem. In C. Paternina-Arboleda, & S. Voß (Eds.), *Computational logistics* (pp. 67–82). Cham: Springer International Publishing.
Wei, R., Rey, S., & Grubesic, T. H. (2022). A probabilistic approach to address data uncertainty in regionalization. *Geographical Analysis*. https://doi.org/10.1111/gean.12282.
Yanık, S., & Bozkaya, B. (2020). A review of districting problems in health care. In R. Z. Ríos-Mercado (Ed.), *Optimal districting and territory design* (pp. 31–55). Cham: Springer International Publishing.
Yanık, S., Sürer, Ö., & Öztayşi, B. (2016). Designing sustainable energy regions using genetic algorithms and location-allocation approach. *Energy, 97*, 161–172.
Yaník, S., Kalcsics, J., Nickel, S., & Bozkaya, B. (2019). A multi-period multi-criteria districting problem applied to primary care scheme with gradual assignment. *International Transactions in Operational Research, 26*, 1676–1697.