



Towards a stochastic programming modeling framework for districting

Antonio Diglio¹ · Stefan Nickel^{2,3} · Francisco Saldanha-da-Gama^{4,5}

Published online: 20 May 2020

© Springer Science+Business Media, LLC, part of Springer Nature 2020

Abstract

In this paper a stochastic districting problem is investigated. Demand is assumed to be represented by a random vector with a given joint probability distribution function. A two-stage mixed-integer stochastic programming model is proposed. The first stage comprises the decision about the initial territory design: the districts are defined and all the territory units assigned to one and exactly one of them. In the second stage, i.e., after demand becomes known, balancing requirements are to be met. This is ensured by means of two recourse actions: outsourcing and reassignment of territory units. The objective function accounts for the total expected cost that includes the cost for the first-stage territory design plus the expected cost incurred at the second stage by outsourcing and reassignment. The (re)assignment costs are associated with the distances between territory units, i.e., the focus is put on the compactness of the solution. The model is then extended in different ways to account for aspects of practical relevance such as a maximum desirable dispersion, reallocation constraints, or similarity of the second-stage solution w.r.t. the first-stage one. The new modeling framework proposed is tested computationally using instances built using real geographical data.

Keywords Stochastic Districting · Equity objective · Uncertain Demand · Stochastic Programming · Complete recourse

The research of the third author was supported by the Portuguese Science Foundation (FCT—Fundação para a Ciência e Tecnologia) under the Projects UID/MAT/ 04561/2013 (CMAF-CIO/FCUL) and SFRH/BSAB/130291/2017.

✉ Antonio Diglio
antonio.diglio@unina.it

¹ Department of Industrial Engineering (DII), Università degli Studi di Napoli Federico II, Piazzale Tecchio, 80, 80125 Naples, Italy

² Institute of Operations Research, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany

³ Department of Logistics and Supply Chain Optimization, Research Center for Information Technology (FZI), Karlsruhe, Germany

⁴ Dept. Estatística e Investigação Operacional, Faculdade de Ciências da Universidade de Lisboa, 1749-016 Lisbon, Portugal

⁵ Centro de Matemática, Aplicações Fundamentais e Investigação Operacional, Faculdade de Ciências da Universidade de Lisboa, 1749-016 Lisbon, Portugal

1 Introduction

Districting Problems (DPs) aim at partitioning a set of basic geographic areas, named Territorial Units (TUs), into a set of clusters, called districts, according to some criteria. The latter typically refer to balancing, which expresses the need for districts of equitable size in terms of dimension, as well as to topological properties like contiguity and compactness. Contiguity means that in order to travel between TUs in the same district there is no need to cross other districts. A contiguity requirement is relevant for dealing appropriately with enclaves (a district within a district). A good districting plan does not contain enclaves. Compactness indicates that a district is somewhat round-shaped and undistorted (Kalcics and Ríos-Mercado 2019). Nevertheless, other relevant criteria in DPs include respecting natural boundaries, existing administrative subdivisions, similarity w.r.t existing districting plans, and socio-economic and cultural homogeneity (Bozkaya et al. 2003; Kalcics et al. 2005; Kalcics and Ríos-Mercado 2019).

The relevance of DPs resides in the strategic and long-term nature of the decisions involved, motivated by a wide spectrum of practical applications arising in different sectors. DPs have been extensively applied to tackle problems emerging in the context of political districting (Ricca and Simeone 2008; Ricca et al. 2013) and in strategic service planning and management like health-care (Blais et al. 2003; Benzarti et al. 2013), school systems (Ferland and Guénette 1990; Schoepfle and Church 1991; Caro et al. 2004; Bruno et al. 2016a), energy and power distribution networks (Bergey et al. 2003; De Assis et al. 2014; Yanık et al. 2016), police districts (D’Amico et al. 2002), waste collection (Mourão et al. 2009), and transportation (Bruno and Laporte 2002; Tavares et al. 2007). Other core applications of DPs regard the design of commercial areas to be assigned to a given sales force (Zoltners and Sinha 2005; Ríos-Mercado and López-Pérez 2013) and distribution logistics (Zhong et al. 2007).

For extensive reviews on DPs readers are encouraged to refer to Kalcics et al. (2005) and, for a more up-to-date overview, to Kalcics and Ríos-Mercado (2019). Given these references, next we focus on some recent contributions found in literature.

Bianchi et al. (2016) and Kim et al. (2017) explore an interesting application of DPs, which consists of the territory design of functional regions. In this kind of problems the hypotheses of complete and exclusive assignment of TUs to districts are relaxed and only clusters with a strong spatial interaction are created. Accordingly, the authors seek to optimize a distance-based compactness measure. The latter is frequently used as an objective function in DPs and represents a good proxy of users’ accessibility to public facilities (Bruno et al. 2017b); it usually leads to shorter travel times when designing distribution areas for service provision (Bender et al. 2016; García-Ayala et al. 2016).

The minimization of maximum territory dispersion, namely the maximum distance between any TU and the center of the districts they are assigned to, is considered in Ríos-Mercado (2016) and Ríos-Mercado and Escalante (2016). Due to the multiplicity of planning goals to be simultaneously achieved, some works adopt a multicriteria setting (Camacho-Collados and Liberatore 2015; Camacho-Collados et al. 2015; Xie and Ouyang 2016).

The objective to be optimized can also be specifically tailored according to the application. For instance, De Fréminville et al. (2015) introduce the so-called Financial Product Districting Problem, where the goal is to partition a set of geographical units in such a way that a cost homogeneity is achieved within the designed territories. Such homogeneity is expressed in terms of the cost variance. Bruno et al. (2016b) define a model to support the rationalization process of public facilities aimed at optimizing the total cost needed to the activation of additional capacities to satisfy the reallocated demand generated by the closure of some

facilities. Lin et al. (2017) propose a mixed-integer programming formulation for a problem emerging in the context of home health-care services: the so-called Meals-On-Wheels service districting problem. The goal is to design the minimum number of districts covering all the TUs. Akdoğan et al. (2018) consider the minimization of the mean response time in a problem involving the location of emergency services.

One aspect of practical relevance in DPs is the need to cope with changing demand. This may stem for instance from the expansion of urban areas or migration movements. Depending on the particular problem we are dealing with, different possibilities emerge. One is to assume a reactive posture and solve a so-called redistricting problem. This is an optimization problem that aims at redesigning existing districts. De Assis et al. (2014) tackle such a problem in the context of meter reading in power distribution networks. A bi-objective problem is investigated. The objectives are related with compactness and homogeneity of districts. The authors develop a heuristic algorithm to approximate the Pareto front. Other works dealing with redistricting problems include those by Caro et al. (2004) and Bruno et al. (2017b). In particular, Caro et al. (2004) propose a mathematical model and a GIS-based approach to solve a school redistricting problem, whereas Bruno et al. (2017b) present several formulations to address a redistricting problem emerging in the redesign of administrative boundaries of the Italian provinces.

One alternative to cope with changes in demand is to become proactive and make a decision that directly accounts for such changes. When accurate forecasts are available, we can embed time in the optimization models. To the best of our knowledge, the only papers addressing multi-period territory design are those by Lei et al. (2015), Bender et al. (2016), Lei et al. (2016), and Bender et al. (2018). A multicriteria optimization framework is considered in the first and third works.

Finally, if demand changes are uncertain then embedding uncertainty in the models may be desirable. Assuming that uncertainty can be measured using a probability function a stochastic programming model emerges as appropriate.

To the best of the authors' knowledge, the existing work closely related to stochastic districting has been developed in the context of vehicle routing problems. In that case,

the problem is cast as a two-stage stochastic program, where districts are designed in the first stage and routing decisions are planned in the second stage, once demands (Haugland et al. 2007) or customers (Lei et al. 2012) are revealed. These are problems in which the districting decisions are triggered by the need to build "good" routes for visiting the customers. In the above studies, tailored heuristic and metaheuristic solution methods are proposed. Stochastic vehicle routing problems based on efficient partitioning procedures of the service region are also exploited by Carlsson (2012) and Carlsson and Delage (2013).

In this paper we introduce a Stochastic Districting Problem with Recourse (SDPR) aiming at partitioning a given set of TUs into a previously given number of clusters in order to maximize the overall compactness while meeting balancing constraints, expressed in terms of average demand per district. The demand associated to each TU is represented by a random variable. Districts are created in the first stage by allocating the basic areas to those TUs chosen as representative of the districts (centers). In the second stage, two recourse actions are considered: (i) outsourcing shortage/surplus demand; (ii) solving a redistricting problem.

The new modeling framework we propose can be useful to solve practical problems, namely: (i) problems emerging in the context of service districting, where the need to provide users with high service levels and fair accessibility conditions is a top priority for decision makers (schools, hospitals, postal and emergency services); (ii) problems aiming

at redesigning political and administrative boundaries; (iii) planning of sales territory where demands for goods and services can be highly unpredictable.

Changing conditions in the labor market, the phenomenon of migratory flows and the strong impact of technology development on customers' habits and behaviors, are examples of triggering factors that may push towards profound reorganization processes to meet new socio-economic and cultural homogeneity requirements and future demand trends (ESPAS, 2015). In all these cases, it is desirable to devise a plan able to hedge against uncertainty.

The contribution of this paper to the literature is fourfold: (i) to introduce a new modeling framework for a two-stage stochastic districting problem; (ii) to embed redistricting decisions as a way to hedge against uncertainty; (iii) to show the relevance of capturing stochasticity in districting problems; (iv) to show that the new models proposed in this paper make sense i.e., lead to plausible solutions.

Due to the strong connection between districting problems and facility location problems, we refer the reader to the book chapter by Correia and Saldaha-da-Gama (2019) and to the references therein for an overview on discrete stochastic facility location problems. Throughout the current paper we emphasize the specific aspects emerging in the context of districting.

The remainder of this paper is organized as follows: in Sect. 2 we introduce an optimization model for stochastic districting with outsourcing. This model is extended in Sect. 3 where the redistricting is assumed as a recourse action. In Sect. 4 we discuss the relevance of considering a stochastic modeling framework for our problem. Some extensions to the base model are then discussed in Sect. 5. Section 6 reports on the computational tests performed for assessing the relevance of our findings. Then, the paper ends with an overview of the work done and the main conclusions drawn from it.

2 A stochastic districting problem with outsourcing

In this section we introduce a first stochastic districting problem. For modeling purposes our starting point is the model proposed by Hess et al. (1965) that we revisit briefly to ensure that this manuscript is self-contained. We start by introducing some notation that we adapt when necessary according to the specific problem we are investigating.

We consider a set I of territorial units (TUs) that we want to divide into a fixed number, say p , of districts. Each district has a representative TU. Hence, when some other TU is assigned to the district we abuse the language and say that we are assigning a TU to the representative of the district. Single-assignment is assumed for the TUs as customary in districting problems. Additionally, we consider the following parameters:

- d_i , demand of TU i ($i \in I$);
- c_{ij} , cost for assigning TU i to TU j ($i, j \in I$);
- μ , target value for the demand in every district.

Typically $\mu = (1/p) \sum_{i \in I} d_i$, which we are also assuming unless stated otherwise. α , maximum desirable deviation of the demand in a district w.r.t. the reference value μ . We consider $\alpha \in (0, 1)$.

A districting problem was first formulated as an optimization problem by Hess et al. (1965) who considered the following binary variables:

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

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

Using the above decision variables, Hess et al. (1965) proposed the following mathematical model:

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

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

$$\sum_{j \in I} x_{jj} = p, \quad (3)$$

$$(1 - \alpha)\mu x_{jj} \leq \sum_{i \in I} d_i x_{ij} \leq (1 + \alpha)\mu x_{jj}, \quad j \in I, \quad (4)$$

$$x_{ij} \leq x_{jj}, \quad i, j \in I, \quad (5)$$

$$x_{ij} \in \{0, 1\}, \quad i, j \in I, \quad (6)$$

The objective function (1) quantifies the total 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. Constraints (4) are the balancing constraints and Inequalities (5) state that we can only assign TUs to representatives of a district. Finally (6) define the domain of the decision variables.

Observing the model we easily conclude that Constraints (5) are redundant in the presence of (4). We are considering them because they will be relevant to ensure the consistency of the stochastic programming models to be presented.

In districting problems, the costs c_{ij} are typically related with the distances (Kalcics and Ríos-Mercado 2019). 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 (Hess et al. 1965). The reader may refer to Kalcics and Ríos-Mercado (2019) for variants of distance-based compactness measures. 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 (Bergey et al. 2003; Bard and Jarrah 2009).

Remark 1 The above model does not solve exactly a districting problem but only a relaxation of it; it solves a location-allocation model with a demand balancing requirement. The model ignores one important aspect in districting: contiguity (and consequently the exclusion of enclaves). Therefore, apparently, a solution provided by the model may yield contiguous solutions only by chance. Nevertheless, when we seek to optimize a compactness measure in a districting problem the resulting solution is often strong in terms of contiguity.

It is also true that the introduction of balancing constraints may disfavor compactness and, thus districts' contiguity. In other words, balancing and compactness may easily become conflicting measures. Despite these facts, most of the authors do not explicitly address contiguity

in their models (Kalcsics and Ríos-Mercado 2019). Hence the model proposed by Hess et al. (1965) keeps being the basic model considered in the literature on districting problems.

For this reason, we also consider it as a starting point for the developments proposed in the current paper. \square

2.1 A stochastic districting problem with auxiliary recourse decisions

A natural source of uncertainty in districting problems are the demands d_i , $i \in I$ and, consequently, the reference value μ (average demand per district). If the organization of the territory into districts is a here-and-now decision to make (i.e. made before uncertainty is revealed) it is desirable that it hedges against such uncertainty.

Under uncertainty, looking for a solution that is feasible for all the possible future observations (scenarios) of the demand vector $\xi = [d_1, \dots, d_{|I|}]$ may be impossible. Even if it is possible, it may lead to a “fat” solution because we may end up planning for realizations that are too extreme although occurring with a very low probability. An alternative is to devise a plan that takes uncertainty into account without being too strict in terms of imposing its feasibility for every “future” but considering some recourse action in case the initial solution is rendered infeasible for the demands actually observed.

The above setting can be looked into in the context of two-stage stochastic programming when $\xi = [d_1, \dots, d_{|I|}]$ is a random vector with some known joint cumulative distribution function (e.g. estimated using historical data).

Under uncertainty, the balancing constraints (4) are not well-defined before the actual demands become known. Therefore, we must relax such constraints when looking for a here-and-now solution and assume that extra costs are incurred if, upon observing the actual demand, we realize that in some district it is above [below] the maximum [minimum] threshold. These costs correspond to additional measures due to having unbalanced demand (e.g., outsourcing for surplus demand). As we show next, the deterministic model introduced by Hess et al. (1965) can be reformulated to capture the new setting we are considering.

Let us denote by g_j (> 0) the extra cost in district j for every unit of demand above the maximum threshold and by h_j (> 0) the extra cost for every unit of demand below the minimum threshold (w.r.t a here-and-now solution).

Additionally, let us consider two sets of auxiliary variables accounting for the “shortage” and “surplus” demand in each district w.r.t the thresholds initially set:

$$\begin{aligned}\psi_j &= \text{demand surplus w.r.t. the maximum threshold, } j \in I; \\ \varphi_j &= \text{demand shortage w.r.t. the minimum threshold, } j \in I;\end{aligned}$$

If the surplus demand is outsourced, the corresponding amount is represented by variables ψ_j . The new problem we are dealing with can be formulated mathematically as a two-stage stochastic programming problem:

$$\begin{aligned}\text{minimize} \quad & \sum_{i \in I} \sum_{j \in I} c_{ij} x_{ij} + \mathcal{Q}(\mathbf{x}), \\ \text{subject to} \quad & (2), (3), (5), (6),\end{aligned}\tag{7}$$

with $\mathcal{Q}(\mathbf{x}) = E_{\xi}[\mathcal{Q}(\mathbf{x}, \xi)]$ and

$$\mathcal{Q}(\mathbf{x}, \xi) = \min \sum_{j \in I} g_j \psi_j(\xi) + \sum_{j \in I} h_j \varphi_j(\xi),\tag{8}$$

$$\text{s. t.} \quad (1 - \alpha)\mu x_{jj} \leq \sum_{i \in I} d_i x_{ij} + \varphi_j(\xi) - \psi_j(\xi) \leq (1 + \alpha)\mu x_{jj}, \quad j \in I, \quad (9)$$

$$\varphi_j(\xi) \geq 0, \quad j \in I, \quad (10)$$

$$\psi_j(\xi) \geq 0, \quad j \in I. \quad (11)$$

In the above model, the first stage problem seeks a here-and-now territory design (possibly violating the balancing constraints for some—or all—observations of the demand vector). The second stage model (8)–(11) accounts for the extra cost incurred due to the actual demand observed and given a first-stage decision. Note that in the second-stage model the values of x_{ij} are constants. Although a feasible solution to the second-stage problem may have $\varphi_j > 0$ and $\psi_j > 0$ for some $j \in I$, it is easy to see that every optimal solution to that problem will have at most one of those variables above zero (due to $g_j, h_j > 0$).

The above stochastic programming problem has simple recourse since the coefficient matrix of the second-stage decision variables is deterministic and can be written as $[I] - I$. In particular, we note that for every here-and-now feasible decision there is a feasible completion in the second stage, which is a nice (and desirable) feature of a stochastic program. Related with this we also note that the values for the variables $\varphi_j(\xi)$ and $\psi_j(\xi)$ can be trivially obtained for every observation of the underlying random vector. For this reason such variables do not actually correspond to a decision we can make—they are auxiliary variables that help us to formulate the problem in a more elegant way (see Remark 3). For this reason we name the problem we are dealing with as a stochastic districting problem with auxiliary recourse (SDPAR).

We note that $Q(\mathbf{x}, \xi)$ is a random variable since the quantities d_i (and thus μ) are random variables. By considering its expected value for defining the recourse function $Q(\mathbf{x})$, we are assuming a so-called neutral attitude of the decision maker towards risk which in our opinion defines a reasonable starting point for the study of stochastic districting problems. Other attitudes towards risk often lead to measures that generalize the one we are considering and thus their analysis should be performed as follow-up to what we are presenting in the current paper.

Finally, we note that like the original model proposed by Hess et al. (1965) this model represents in fact a stochastic location-allocation problem with balancing requirements. Nevertheless, as the computational results presented in Sect. 6 show, it renders plausible solutions to our districting problem.

Assuming a finite support, say Ξ , for the random vector ξ , we can go farther in terms of the model. In that case, we can index the different scenarios in a finite set, say $S = \{1, \dots, |\Xi|\}$. Moreover, we can index in S the stochastic demands, the assignment costs as well as the second-stage decision variables, as follows:

d_{is} ,	demand of TU $i \in I$ under scenario $s \in S$;
φ_{js}	demand shortage in district $j \in I$ w.r.t. the minimum threshold under scenario $s \in S$;
ψ_{js}	demand surplus in district $j \in I$ w.r.t. the maximum threshold, $j \in I$ under scenario $s \in S$.

We assume that the probabilities associated with the different scenarios are known (for instance estimated using historical data). We define by π_s the probability that scenario s

occurs, $s \in S$. Naturally, $\pi_s \geq 0$, $s \in S$ and $\sum_{s \in S} \pi_s = 1$. Additionally, we denote by $\hat{\mu}$ the reference value to be used in the balancing constraints (to be detailed later).

We can finally write the so-called extensive form of the deterministic equivalent that we call model (M_1) :

$$\text{minimize} \quad \sum_{i \in I} \sum_{j \in I} c_{ij} x_{ij} + \sum_{s \in S} \pi_s \left(\sum_{j \in I} g_j \psi_{js} + \sum_{j \in I} h_j \varphi_{js} \right), \quad (12)$$

subject to (2), (3), (5), (6),

$$(1 - \alpha) \hat{\mu} x_{jj} \leq \sum_{i \in I} d_{is} x_{ij} - \psi_{js} + \varphi_{js} \leq (1 + \alpha) \hat{\mu} x_{jj}, \quad j \in I, s \in S, \quad (13)$$

$$\varphi_{js} \geq 0, \quad j \in I, s \in S, \quad (14)$$

$$\psi_{js} \geq 0, \quad j \in I, s \in S. \quad (15)$$

The objective function (12) can be written in a different way:

$$\sum_{i \in I} \sum_{j \in I} c_{ij} x_{ij} + \sum_{j \in I} (g_j \sum_{s \in S} \pi_s \psi_{js}) + \sum_{j \in I} (h_j \sum_{s \in S} \pi_s \varphi_{js}), \quad (16)$$

Remark 2 In the original model by Hess et al. (1965) the target demand per district is defined as the average demand per district (total demand divided by the number of districts). Under uncertainty, demand is not known beforehand, the same holding with the average demand per district that becomes a random variable. For this reason, in general, the best we can do is to consider the expected average. Considering that uncertainty is captured by a finite number of scenarios such expectation is obtained as follows:

For every scenario $s \in S$, $\sum_{i \in I} d_{is}$ is the total demand under that scenario. Accordingly, $\sum_{s \in S} (\pi_s \sum_{i \in I} d_{is})$ is the expected total demand. Hence, we get the value we are looking for:

$$\hat{\mu} = \frac{1}{P} \sum_{s \in S} (\pi_s \sum_{i \in I} d_{is}). \quad (17)$$

□

Remark 3 The knowledge about the values of the x -variables as well as of the occurring scenario immediately determines the values of the (auxiliar) variables φ_{js} and ψ_{js} ($j \in I$, $s \in S$):

$$\varphi_{js} = \max \left\{ 0, (1 - \alpha) \hat{\mu} x_{jj} - \sum_{i \in I} d_{is} x_{ij} \right\}, \quad j \in I, s \in S;$$

$$\psi_{js} = \max \left\{ 0, \sum_{i \in I} d_{is} x_{ij} - (1 + \alpha) \hat{\mu} x_{jj} \right\}, \quad j \in I, s \in S.$$

Hence, we could easily reformulate our problem as a single-stage problem. However, we omit this reformulation since it is not helpful when it comes to solving the problem. □

3 A stochastic districting-redistricting problem

In the stochastic model presented in the previous section, a here-and-now decision is made and there is no effective recourse decision. As stated in Remark 3, the second-stage decision variables considered simply help us to account for the costs of having demand surplus or shortage w.r.t. the minimum and maximum thresholds set for the districts.

In practice, it may be relevant to be proactive. This means that upon observing a violation in the balancing constraints in some scenarios, instead of simply taking recourse actions that overcome infeasibilities of the here-and-now solution (and thus accounting for the corresponding costs), we may try to adapt slightly the territory design (the first-stage solution) to the occurring demands. In other words, we may think of performing a so-called redistricting, in line with works such as those by Bruno et al. (2017b), Caro et al. (2004), and De Assis et al. (2014). This is what we propose next.

We still seek to define a fixed number p of districts. This corresponds to a major decision that keeps being done here-and-now. As mentioned before, we may end up facing a scenario for which the demand in some district is above or below the thresholds initially set. In this case, in addition to the recourse actions discussed in the previous section we also consider a “redistricting” recourse decision for some (hopefully just a few) territories. This means that such territories would be included in other district(s) as a reaction to the occurring scenario. This type of recourse action makes sense since it corresponds to a “redistribution” of the demand in order to get an overall balanced solution. The only TUs that cannot be reassigned are those that were set as district representatives by the first stage decision.

We keep assuming that the support of the random vector underlying the problem is finite. Hence, we consider directly scenario-indexed demands. Furthermore, we assume that reassigning a TU incurs an extra cost. We define by r_{ijs} the cost for reassigning the demand of TU i to the district represented by TU j , under scenario $s \in S$. Like for the previous models, the reassignment costs r_{ijs} can be related with the distances. Nevertheless, it may be desirable to “penalize” reassignments. For this extension we consider one additional set of decision variables:

$$w_{ijs} = \begin{cases} 1, & \text{if TU } i \text{ is assigned to district } j \text{ under scenario } s; \\ 0, & \text{otherwise.} \end{cases} \quad (i, j \in I, s \in S)$$

In addition to these variables, we consider a set of auxiliary variables that help us to present a linear model: for every $i, j \in I, s \in S$, v_{ijs} is a binary variable equal to 1 if the assignment of TU i to TU j under scenario s corresponds actually to a reassignment. More formally, we can define these variables as:

$$v_{ijs} = \begin{cases} 1, & \text{if } w_{ijs} = 1 \text{ and } x_{ij} = 0; \\ 0, & \text{otherwise.} \end{cases} \quad (i, j \in I, s \in S)$$

The new problem will be called a Stochastic Districting Problem with Recourse (SDPR). Given all the aspects already discussed as well as the notation above introduced we can directly present the extensive form of the deterministic equivalent denoted by (M_2) :

$$\begin{aligned} \text{minimize} \quad & \sum_{i \in I} \sum_{j \in I} c_{ij} x_{ij} \\ & + \sum_{s \in S} \pi_s \left(\sum_{i \in I} \sum_{j \in I} r_{ijs} v_{ijs} + \sum_{j \in I} g_j \psi_{js} + \sum_{j \in I} h_j \varphi_{js} \right), \end{aligned} \quad (18)$$

subject to (2), (3), (5), (6), (14), (15),

$$\sum_{j \in I} w_{ijs} = 1, \quad i \in I, s \in S \quad (19)$$

$$(1 - \alpha)\hat{\mu}x_{jj} \leq \sum_{i \in I} d_{is}w_{ijs} - \psi_{js} + \varphi_{js} \leq (1 + \alpha)\hat{\mu}x_{jj}, \quad j \in I, s \in S, \quad (20)$$

$$\sum_{\ell \neq j} w_{j\ell s} \leq 1 - x_{jj}, \quad j \in I, s \in S, \quad (21)$$

$$v_{ijs} \geq w_{ijs} - x_{ij}, \quad i, j \in I, s \in S, \quad (22)$$

$$v_{ijs} \geq 0 \quad i, j \in I, s \in S, \quad (23)$$

$$w_{ijs} \in \{0, 1\}, \quad i, j \in I, s \in S. \quad (24)$$

The objective function (18) accounts for the initial territory design plus the expected cost for redesigning the territory, and the expected costs for demand shortage and surplus in each district (w.r.t. the minimum and maximum thresholds, respectively). Constraints (19) and (20) seek a territory redesign (dependent on the observed scenario). Constraints (21) guarantee that a district that is selected as a district representative according to the first-stage solution is not reassigned in the second stage. Constraints (22) and (23) are an alternative way of writing that $v_{ijs} \geq \max\{0, w_{ijs} - x_{ij}\}$, $i, j \in I, s \in S$. On the other hand, due to the non-negativity costs r_{ijs} we know that in every optimal solution the equality holds, i.e., $v_{ijs} = \max\{0, w_{ijs} - x_{ij}\}$, $i, j \in I, s \in S$. Therefore, we are paying the reassignment cost for the demand that is actually reassigned. Finally, constraints (24) define the domain of the w -variables.

In principle, the above model allows the reassignment of all territories. However, since there are extra costs for reassignments as well as for demand shortage and surplus, an optimal solution to the problem will seek to reassigning as little demand as necessary to achieve an overall balancing.

Remark 4 The above model is a generalization of model min (16), s. t. (2), (3), (5), (6), (13)–(15): setting the costs $r_{ijs} = \infty$, $i, j \in I, s \in S$, all the v -variables become equal to zero in an optimal solution which leads to the redundancy of all constraints involving these variables. \square

Remark 5 Due to the auxiliary variables ψ_{js} and φ_{js} ($j \in I, s \in S$) there is always a feasible solution to the above model. Hence, for every feasible first-stage solution, there is always a feasible completion at the second stage, which means that we are again dealing with a stochastic programming with complete recourse. \square

4 The relevance of considering a stochastic modeling framework

In the previous sections we have considered stochastic programming modeling frameworks for a districting problem under uncertainty. One important question in this case concerns the

relevance of considering such frameworks instead of simpler ones (e.g. deterministic). Two measures are usually considered for evaluating such relevance: the Value of the Stochastic Solution (VSS) and the Expected Value of Perfect Information (EVPI). The reader can refer to Birge and Louveaux (2011) for further details. Next we specialize these measures to the more general problem that we considered: SDPR.

Consider the model:

$$\begin{array}{ll} \text{minimize} & (18), \\ \text{subject to} & (2), (3), (5), (6), (14), (15), \\ & (19) - (24), \end{array}$$

and denote by SP its optimal value.

Consider the single-scenario problem obtained by replacing all the random variables with the corresponding expected values. By solving that model, we obtain a first-stage solution, say \hat{x} , that is also feasible for the stochastic problem. Hence, we can evaluate its cost as a feasible solution to the stochastic problem. To do so, we just need to fix the values of the x -variables according to \hat{x} in the problem $\min (18) \text{ s. t. } (2), (3), (5), (6), (14), (15), (19) - (24)$. The resulting value is denoted by EEV (the Expected value of the optimal solution provided by the Expected Value problem, see Birge and Louveaux 2011).

The value of the stochastic solution is computed as $VSS = EEV - SP$. This (non-negative) value indicates how good is the optimal solution to the expected value problem as an approximation to the optimal solution of the stochastic problem.

The EVPI is a measure of how much a decision maker would be willing to pay to get access to perfect information. A high EVPI indicates that having access to that information is quite relevant which means that uncertainty plays an important role in the problem. In order to compute the EVPI we start by solving the single scenario problem for every possible scenario. Denote by \mathcal{V}_s the corresponding optimal value ($s \in S$). The so-called Wait-and-See solution value is defined as $WS = \sum_{s \in S} \pi_s \mathcal{V}_s$. The expected value of perfect information is equal to $EVPI = SP - WS$.

5 Extensions of the proposed model

In this section, we discuss some possible extensions to the problem investigated in the previous sections. These extensions are motivated by the need to include features of practical relevance that were not considered in our formulations. Examples of such features include territory dispersion and similarity w.r.t. to an existing districting plan. Furthermore, it may be relevant to manage some practical issues related with the second-stage decisions since the reassignment recourse action that we have considered can lead to disadvantageous allocations for TUs as we discuss below.

The additional aspects discussed in this section are modeled mathematically by the introduction of new sets of constraints in the optimization model presented in Sect. 3. Unless stated otherwise, we adopt the assumptions and modeling considerations presented in that section.

5.1 Maximum dispersion

As we mentioned directly in Sect. 1, territory dispersion indicates the maximum distance between any TU and the center of the district it is allocated to (Ríos-Mercado 2016; Ríos-

Mercado and Escalante 2016). In the context of the stochastic districting problem we are studying, let us suppose that there is a maximum desirable dispersion, say l_{\max} , for the districting to be obtained. This can be easily ensured for the first-stage solution by setting $x_{ij} = 0$, for all $(i, j) \in I$ such that $\ell_{ij} > l_{\max}$. Moreover, if a maximum value for the dispersion should also hold for the second-stage solution then all variables x_{ij} , w_{ijs} , and v_{ijs} with $\ell_{ij} > l_{\max}$ are set equal to 0 for every $s \in S$. In this case, such variables do not even need to be considered/defined.

There is a trivial maximum dispersion that makes sense to impose to a solution which is the one that results from solving the plain model (M_2). This aspect is taken into account later when we report on the computational tests performed.

5.2 Reallocation constraints

In the SDPR that we are studying, second-stage decisions ensure TUs reallocations to avoid demand shortages or surplus in the created districts. However, as our computational experiments show, adapting a solution to the occurring scenarios may disfavor the compactness (and thus contiguity) of the districts. Hence, a natural extension to the problem consists of limiting the number of reassignments in the second stage.

Denoting by m the maximum number of allowed reassignments, we can make use of the v -variables presented in the previous section, and impose:

$$\sum_{i \in I} \sum_{j \in I} v_{ijs} \leq m, \quad s \in S. \quad (25)$$

5.3 Similarity with respect to an initial plan

Another relevant aspect often considered in DPs, regards the need to guarantee a certain degree of “similarity” w.r.t. some territory organization already planned (Bozkaya et al. 2003; Caro et al. 2004; Bruno et al. 2017a).

In our stochastic setting we can consider similarity constraints associated with the second stage in order to ensure that we redesign districts keeping a certain degree of similarity w.r.t. the first-stage districting plan. This can be ensured using the following inequalities:

$$\sum_{i \in I} x_{ij} w_{ijs} \geq \gamma \sum_{i \in I} x_{ij}, \quad j \in I, s \in S, \quad (26)$$

In this expression, $\gamma \in [0, 1]$ is the minimum percentage of TUs to be kept together in the same district according to the first-stage plan.

Constraints (26) can be linearized by considering the following linear constraints:

$$v_{ijs} \leq w_{ijs}, \quad i, j \in I, s \in S, \quad (27)$$

$$x_{ij} + v_{ijs} \leq 1, \quad i, j \in I, s \in S, \quad (28)$$

Constraints (27) guarantee that in each scenario the v -variables do not account for a reassignment to some district if the assignment does not hold in the second stage solution. Constraints (28) ensure that TU i can be either assigned to district j in the first-stage solution or reassigned to the same district in a second-stage districting plan for every scenario $s \in S$.

Hence, Constraints (26) can be rewritten as follows:

$$\sum_{i \in I} (w_{ijs} - v_{ijs}) \geq \gamma \sum_{i \in I} x_{ij}, \quad j \in I, s \in S, \quad (29)$$

6 Computational experiments

In this section, we report on the computational tests performed to assess the relevance of our contribution. The tests aim at checking whether the models yield plausible solutions. In this analysis we consider real geographical data.

In Sect. 6.1 we describe the test data used. In Sect. 6.2 we focus on one particular instance as a way to better understand the type of solutions we obtain using our model and also to understand the consequences driven by changing the reallocation costs. We proceed in Sect. 6.3 by using the same instance to analyze the effect in the solutions by considering the extensions proposed in Sect. 5. In Sect. 6.4 we extend the results to all the instances with a fixed number of TUs. In this case we focus on the VSS, EVPI and CPU time. Finally, in Sect. 6.5 we extend the results farther by considering all the generated instances (i.e., varying number of TUs). In this case our focus will be the CPU time required to solve the model.

6.1 Test data

The stochastic programming models proposed in the previous sections were tested using data corresponding to the province of Novara, in northern Italy. This is a province divided into 88 municipalities (ISTAT, 2011) that we take as the reference TUs for our study.

Making use of the GIS software QGIS we discretized the region by determining the centroids of the municipalities. Afterwards, we identified set I with the set of the extracted centroids, where we suppose that demands d_i are concentrated. Euclidean distances ℓ_{ij} between all pairs of centroids $i, j \in I$ have been computed. The province of Novara, its municipalities, and their centroids are depicted in Fig. 1.

Since no real data could be found in terms of the demand for existing services we decided to randomly generate the missing data. A so-called *intermediate* scenario was obtained by randomly generating the 88 demands according to a uniform distribution in the range $[1, 10]$.

Fig. 1 Novara province: the municipalities and the corresponding centroids



Table 1 Probability distributions across the scenarios

s (scenario)	π_{s1}	π_{s2}	π_{s3}	π_{s4}
1 (20% below intermediate)	1/3	1/6	1/6	2/3
2 (intermediate)	1/3	1/6	2/3	1/6
3 (20% above intermediate)	1/3	2/3	1/6	1/6

Then, two additional scenarios were computed by considering a 20% positive and negative deviation respectively, from the *intermediate* scenario.

The reason for considering only three scenarios has to do with the purpose of our computational tests: to check the relevance of the stochastic models proposed in the previous sections. If they are relevant for a small number of scenarios—as we show—they should also be relevant when a larger number of scenarios is considered.

In order to obtain extended data that can better help evaluating the modeling framework we are proposing in this paper, we tested instances that differ among each other in terms of the number of TUs involved. We considered the data corresponding to Fig. 1 and produced a larger instance with 120 TUs. This was done by looking for additional municipalities that have a border in common with some TU(s) already considered. Additionally, we considered 2 subsets of the 88 initial TUs: one with 60 TUs drawn from the 88-TU-instance and another one with 40 TUs drawn from the 60-TU-instance. The selection of the 60 TUs to draw from the 88 was made in such a way that the remaining region would be meaningful (e.g., no holes). The same was done when drawing the 40-TU-instance from the one with 60 TUs. In the appendix, Tables 5 and 6 we detail the relevant information.

Four probability distributions, indexed in $K = \{1, 2, 3, 4\}$, were considered inducing an equal number of base data sets for the computational tests. We denote by π_{sk} the probability of scenario s according to the k -th probability distribution ($k \in K$). The information is provided in Table 1.

As we discussed in Sect. 2.1, the target demand ($\bar{\mu}$) to be used in the balancing constraints needs to be specified in every instance of the problem. In our case, we follow expression (17) and thus we consider a value that depends on the underlying probability distribution. For a particular probability distribution $k \in K$ we can specify it as follows:

$$\bar{\mu}_k = \frac{1}{P} \sum_{i \in I} \sum_{s \in S} \pi_{sk} d_{is}, \quad k \in K.$$

Demands have been chosen as weighting factors in the computation of the assignment costs. For the instance associated with probability distribution k ($k \in K$) we defined:

$$c_{ij}^{(k)} = \ell_{ij} \sum_{s \in S} \pi_{sk} d_{is}, \quad i, j \in I.$$

Concerning the reassignment costs, they were set as follows:

$$r_{ijs} = \omega d_{is} \ell_{ij}, \quad i, j \in I, s \in S.$$

We make the above costs dependent on the observed demand, and Euclidean distance between the centroids of interest and a parameter ω that accounts for the penalty due to the reassignment. The chosen expressions for generating the reassignment costs bare resemblance with those used in the redistricting problems investigated by Caro et al. (2004) and Bruno et al. (2017b).

Remark 6 In case $\omega = 1$ we have $c_{ij}^{(k)} = \sum_{s \in S} \pi_{sk} r_{ijs}$. This means that we have a relation between the initial assignment costs and the second stage (re)assignment costs. This relation allows us to quantify the expected “total second stage (re)assignment cost”, which can be accomplished by considering the expression $\sum_{i \in I} \sum_{j \in I} \sum_{s \in S} \pi_{sk} r_{ijs} w_{ijs}$. We recall that w -variables define the second stage territory design in which some TUs remain assigned like in the first stage (the w -variable coincides with the corresponding x -variable but some other TUs are assigned to other TU and a reassignment occurs). This expression is directly comparable with $\sum_{i \in I} \sum_{j \in I} c_{ij} x_{ij}$.

In other words, if $\omega = 1$ we can directly compare the (expected) compactness of the second-stage territory design with the compactness of the first-stage one. \square

In our experiments, in addition to $\omega = 1$ (to allow the above mentioned comparison) we also considered $\omega = 2$.

Concerning the unit costs for shortage and surplus demand (w.r.t. the reference value) we defined:

$$g_j^{(k)} = h_j^{(k)} = M_k = \max_{i \in I} \{ \ell_{ij} \sum_{s \in S} \pi_{sk} d_{is} \}, \quad j \in I, k \in K.$$

The above values seem too high. However, we think they are appropriate to foster reallocation mechanism by adapting the solution to the occurring demands.

Finally, in order to better test the behavior of our modeling framework, parameter α was varied from 0.05 up to 0.30 with a step of 0.05, while p was set equal to 4 and 6.

In total, for the 88-TU territory depicted in Fig. 1 we generated 96 instances: four values for k , two values for ω , six values for α and two values for p .

All the models were solved using the commercial solver IBM CPLEX v12.8 on an Intel(R) Celeron(R) with 1.50GHz and 4GB of RAM. Next, we report on the computational results obtained by the implementation of the models proposed for the SDPR. In particular, we focus only on model (M_2) since it includes (M_1) as a particular case.

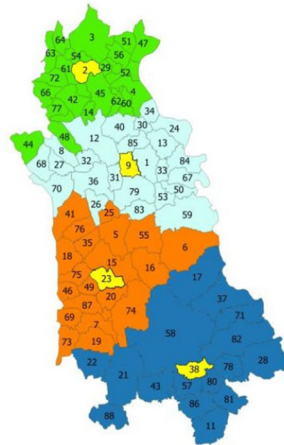
6.2 First observations

We start by analyzing one specific instance to illustrate the relevance of capturing uncertainty in districting problems. The instance considered is the one defined by $p = 4$, $k = 3$, $\omega = 1$, and $\alpha = 0.20$. The results obtained for the SDPR are depicted in Fig. 2, where

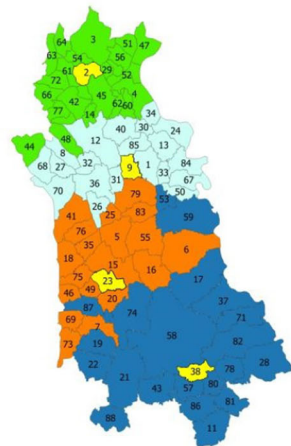
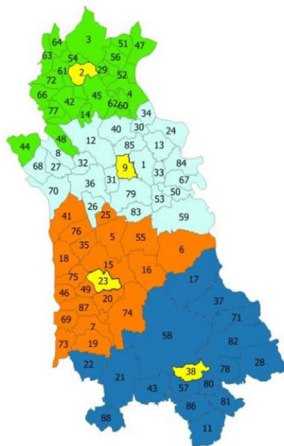
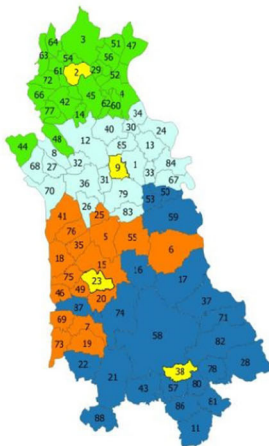
different colors refer to different districts. The TUs selected as districts’ centers or representatives are depicted in yellow. We also note that each TU is labeled with a unique ID code.

In Fig. 2a we observe that the stochastic model led to a first-stage solution with very compact clusters. The cost associated with this territory design is $\sum_{i \in I} \sum_{j \in I} c_{ij} x_{ij} = 3299.38$. Additionally we observe that apart from TU 44, contiguity holds although it was not explicitly imposed (TU 44 belongs to the green district although it is contiguous only to TU 68 and TU 8 that belong to the cyan district).

In Fig. 2b–d we observe the second-stage solutions—one for each possible realization of the demand vector. Looking into these figures we conclude that when demand becomes known the model “suggests” some reassignments in order to meet balancing constraints without incurring in too high penalty costs. For instance, when scenario 1 (“below intermediate”) is observed, 6 TUs (16, 50, 53, 59, 74, and 87) are reallocated from the orange and cyan districts to the dark blue one. This is a way to let the latter to comply with the minimum demand threshold. Similarly, when scenario 3 (“above intermediate”) occurs, 7 TUs (19, 53,



(a) First stage.



(b) Second stage: Scenario 1.

(c) Second stage: Scenario 2.

(d) Second stage: Scenario 3.

Fig. 2 Solutions for the instance defined by $p = 4$, $k = 3$, $\omega = 1$, $\alpha = 0.20$

59, 74, 79, 83, and 87) are reassigned to avoid above-threshold demand in the districts. To accomplish this, for example, the model “suggests” the reallocation of TUs 79 and 83 to the orange district. According to the chosen probability distribution ($k = 3$), the first stage solution remains unchanged when scenario 2 (“intermediate”) is considered.

Finally, we note that the total expected reassignment cost is $\sum_{i \in I} \sum_{j \in I} \sum_{s \in S} \pi_{sk} r_{ijs} v_{ijs} = 163.82$. Since the total penalty cost is equal to zero (all variables ψ_{js} and φ_{js} are equal to zero) we obtain $3299.38 + 163.82 = 3463.2$ as the optimal value to the stochastic problem.

Not surprisingly, we observe that the second stage (re)allocation decisions lead to lower quality solutions both in terms of contiguity and compactness. The latter is testified by the expected total (re)assignment costs $\sum_{i \in I} \sum_{j \in I} \sum_{s \in S} \pi_s r_{ijs} w_{ijs} = 3396.09$ (see Remark 6).

Overall, in this illustrative instance we observe what we would expect (and wish) in practice: a major plan is initially devised (the first-stage solution) that suffers several minor

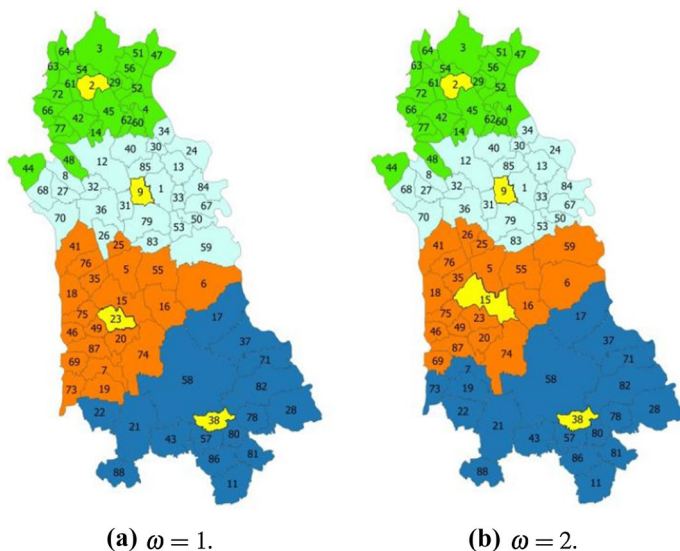


Fig. 3 First stage solutions for the instances with $p = 4$, $k = 3$, and $\alpha = 0.20$

changes according to how uncertainty is revealed. A major change in the first-stage solution would indicate that it was not hedging appropriately against uncertainty.

In a second phase of the experiments, we wanted to see the effect of changing the penalty factor ω . Hence, we considered the instance similar to the previous one but with $\omega = 2$. In Fig. 3 we depict side by side the first-stage solution for both instances. Figure 3a is the same as Fig. 2a but we repeat it for the sake of an easier comparison.

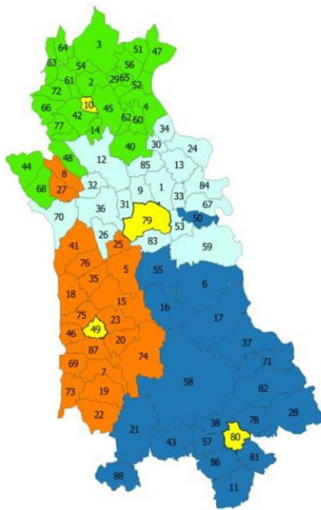
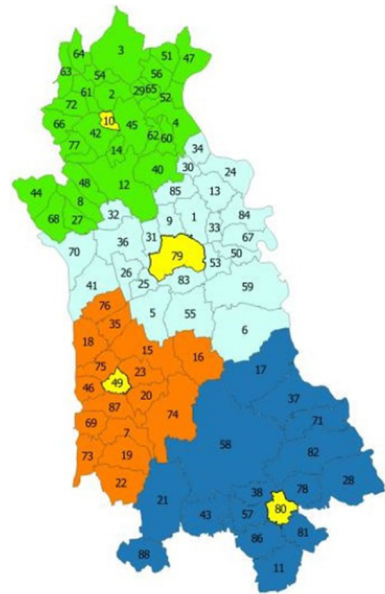
When $\omega = 2$, the penalty for reassignment decisions becomes higher. In this case, the model suggests no reassignment independently from the demand observed. However, in order to better hedge against uncertainty, some changes are observed in the first-stage solution: for $\omega = 2$, TUs 7, 19, and 73 are assigned to the dark blue district, while units 26 and 59 are included in the orange one whose representative is now TU 15. These changes are reflected in a lower compactness of the solution that is reflected in a higher value of the objective function of the model, which increases up to 3498.56.

Finally, in addition to determining the solutions for the above two instances we computed both the VSS and the EVPI. This is done in percentage w.r.t. the optimal value of the stochastic problem (SP), i.e., we compute $100 \times VSS/SP$ and $100 \times EVPI/SP$. As explained before, this is a means to quantify the relevance of considering the stochastic approach we have proposed.

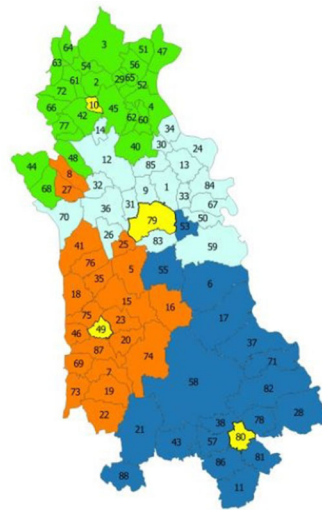
We start by computing the solution for the single-scenario (deterministic) problem that results from replacing the random demands by their expectation. Since the difference between the two above instances regards the value of ω it is easy to conclude that the single scenario problem has the same optimal solution for both $\omega = 1$ and $\omega = 2$. Note that when we consider just one scenario, no reassignment is made. The resulting solution is depicted in Fig. 4. Clearly, this solution differs from the optimal first-stage solutions depicted in Fig. 3.

The optimal solution to the expected value problem presents a higher compactness value than the two first-stage solutions previously analyzed (the territory design costs— $\sum_{i \in I} \sum_{j \in I} c_{ij} x_{ij}$ —have decreased to 3249.18). Nevertheless, it turns out to be very weak when implemented as a first-stage solution in our stochastic problem. When scenarios 1 and

Fig. 4 Optimal solution to the expected value problem ($p = 4$, $k = 3$, $\alpha = 0.20$)



(a) Scenario 1.



(b) Scenario 3.

Fig. 5 Map of the solutions to the expected value problem ($p = 4$, $\alpha = 0.20$, $k = 3$)

3 occur the model forces the reallocation of 10 TUs in order to meet the balancing constraints (Fig. 5). Thus, we obtained $100 \times VSS/SP$ equal to 3.87% ($\omega = 1$) and to 12.77% ($\omega = 2$). This shows that for the two instances considered the expected value problem provides a poor approximation to the stochastic problem, which indicates a high relevance of capturing uncertainty in these cases.

Finally, we computed the percentage value of the EVPI w.r.t. SP, obtaining 6.00% ($\omega = 1$) and 17.64% ($\omega = 2$). This means that the EVPI is equal to 6% of the optimal value of the

Table 2 Results for different thresholds (l_{\max}) defining the maximum dispersion ($l_{\max} = \infty$ corresponds to ignoring the additional constraints)

l_{\max}	SP	EEV	$100 \times \frac{VSS}{SP}$	WS	$100 \times \frac{EVPI}{SP}$	CPU (s)
∞	3463.21	3597.21	3.87	3249.18	6.18	534.33
31.00	3463.99	3597.21	3.85	3249.18	6.20	300.44
27.00	3464.43	3601.97	3.97	3249.18	6.21	313.25
26.00	3464.91	3601.97	3.96	3249.18	6.23	401.59
25.00	3465.68	3601.97	3.93	3249.18	6.25	122.50
22.00	3474.10	3620.88	4.23	3249.18	6.63	293.56
18.00	3479.85	6731.63	93.45	3249.18	12.02	662.77
16.00	3693.27	7080.40	91.71	3249.18	50.08	222.99
15.00	6551.23	7099.49	8.37	3270.69	50.00	86.98
14.00	8890.62	The corresponding EEV is infeasible				39.08

stochastic problem for $\omega = 1$ and 17.64% for $\omega = 2$. Again, we conclude for a clear relevance of the introduced stochastic approach in the analyzed cases.

6.3 The effect of extending the model

We focus now on the extensions to the base model that were proposed in Sect. 5. We consider the same instance as before ($p = 4$, $k = 3$, $\omega = 1$, $\alpha = 0.20$). This allows us to perform some sensitivity analysis and thus to better capture the effect of considering either a maximum dispersion, reallocation constraints or similarity w.r.t. the first-stage districting plan.

6.3.1 Maximum dispersion

To evaluate the effect of including a maximum dispersion constraint, we considered the exclusion of subsets of decision variables as discussed in Sect. 5.1.

In order to ensure the effectiveness of these constraints and thus to analyze their effect, we established a maximum value for l_{\max} defined by

$$l_{\max} = \left\lceil \max_{i,j \in I, s \in S} \{\ell_{ij} \tilde{w}_{ijs}\} \right\rceil,$$

where \tilde{w} stands for the second-stage districting obtained using model (M_2).

In the particular case of the instance we are analyzing we have $\max_{i,j \in I, s \in S} \{\ell_{ij} w_{ijs}\} = 31.60$, which gives $l_{\max} = 31$. Starting from this value, we produced several instances by decreasing the value of l_{\max} successively until the expected value problem becomes infeasible. The results are depicted in Table 2.

Before commenting on Table 2 we note that if we focus on the single-scenario solutions we have $\max_{i,j \in I} \{\ell_{ij} x_{ij}\} = 15.60$. This explains why in this table the wait-and-see values are the same before l_{\max} drops to 15.00.

Concerning the relative values of VSS and EVPI w.r.t SP we observe an increasing trend with the decrease of l_{\max} . They become quite large when we set more stringent values for l_{\max} . We have $100 * VSS/SP$ larger than 90% for l_{\max} equal to 16.00 and 18.00. This holds due to the fact that the expected value problem produces a first-stage solution that implies

higher penalties for reallocations in the second stage namely, when the allowed maximum dispersion decreases. Interestingly, when l_{\max} decreases to 15.00, $100 \times VSS/SP$ decreases significantly. This is due to the fact that the value for that parameter is so stringent that not even the stochastic model can find a good first-stage solution (avoiding high penalties in the second stage). We see that the optimal value to the stochastic problem increases significantly when l_{\max} decreases from 16.00 to 15.00. This is an indication that stringent values of l_{\max} that are not extreme give relevance to a stochastic modeling framework when accounting for maximum dispersion.

Concerning the CPU time required to solve the model with maximum dispersion constraints, the trend is not clear. Nevertheless, we observe that the model can be solved up to proven optimality within a reasonable CPU time.

6.3.2 Reallocation constraints

To evaluate the impact of considering reallocation constraints, we added Constraints (25) to model (M_2). Like for l_{\max} we tried several values of m starting with the first non-binding one that is given by $\max_{s \in S} \{ \sum_{i \in I} \sum_{j \in I} \tilde{v}_{ijs} \}$, where \tilde{v} stands for the optimal values of v -variables when the plain model (M_2) is considered. In our specific instance we obtained a value of 7, i.e., if m is larger than or equal to 7 then Constraints (25) are non-binding. We note that the smallest value we can consider for m is zero which corresponds to the case in which we do not allow for reallocations in the second stage. The results obtained can be observed in Table 3.

The first aspect to point out in Table 3 concerns the wait-and-see value (WS) which is the same regardless the value of m . This holds because reallocations affect only the second-stage solutions, which under a single scenario does not produce any change w.r.t. the first-stage solution. In other words, the reallocation constraints are redundant in a single-scenario instance.

From the above table we can draw other conclusions. First, the relative values of VSS and EVPI w.r.t SP have an increasing trend with the reduction of m .

Concerning the relative values of EVPI we see that the values are in general small. The stochastic program is not much affected by penalties and thus the differences between the wait-and-see solution and the solution to the stochastic problem are rather limited.

Concerning the relative values of VSS (w.r.t. SP), we can see how they rapidly increase when m decreases. This is explained by the fact that the (first-stage) solution for the expected

Table 3 Results corresponding to reallocation constraints for different values of m ($m = \infty$ corresponds to ignoring the additional constraints)

m	SP	EEV	$100 \times \frac{VSS}{SP}$	WS	$100 \times \frac{EVPI}{SP}$	CPU (s)	$\max_{s \in S} \{ \sum_{i,j \in I} v_{ijs} \}$
∞	3463.21	3597.21	3.87	3249.18	6.18	534.33	7
6	3464.00	3629.55	4.78	3249.18	6.20	519.21	5
4	3465.67	5332.71	53.87	3249.18	6.25	209.87	4
3	3469.52	6961.19	100.64	3249.18	6.35	236.44	3
2	3473.69	8625.23	148.30	3249.18	6.46	664.57	2
1	3480.25	10,398.22	198.78	3249.18	6.64	461.14	1
0	3498.56	12,175.12	248.00	3249.18	7.13	27.00	0

value problem implies significant changes (penalties) in the second stage which is not the case when we consider the first-stage solution produced by our stochastic model. This shows that the stochastic programming problem is able to produce solutions that can better hedge against uncertainty.

Finally, as far as the CPU time required to solve the models to proven optimality is concerned, we do not observe a clear trend. In any case, we can solve the model within an acceptable CPU time.

6.3.3 Similarity w.r.t. the first-stage districts

To analyze the effect of imposing a some degree of similarity between the first- and second-stage solutions, we included Constraints (27)–(29) in model (M_2).

To ensure that Constraints (29) are binding, we looked for the «minimum value of similarity», say γ^* , between the first- and second-stage solution that we could observe in our instance. Such a value was found as follows: for every district and scenario we computed the percentage of TUs belonging to the district that remain so in the second stage. Then we look for the minimum value. In synthesis, we set

$$\gamma^* = \min_{j \in I, s \in S} \left\{ \frac{\sum_{i \in I} x_{ij} w_{ijs}}{\sum_{i \in I} x_{ij}} \right\}.$$

For the instance we have been working with we obtained $\gamma^* = 0.84$ (using the solutions depicted in Fig. 2). The cyan district is the one to which this lowest value corresponds. In the first-stage solution (Fig. 2a), 25 TUs are in this district out of which 21 remain so in the second stage both for scenarios 1 and 3 (Figs. 2b, d). Higher values are found in the other cases.

Hence, when implementing Constraints (29) we set $\gamma = 0.85$. To get a deeper insight on parameter γ we then tested other possibilities, namely: 0.90, 0.95, and 1.00. The last one corresponds to the case in which we have full similarity between the first-stage and the second-stage solutions (no reallocation occurs in the second stage).

The results are summarized in Table 4.

Like for the reallocation constraints, similarity constraints are redundant in a single-scenario setting, which explains the constant value observed for the wait-and-see solution (WS).

Both the the relative values of VSS and EVPI w.r.t SP exhibit an increasing trend when γ increases.

Again, the values observed for the (relative) EVPI are not very high. The explanation is similar to that presented for a similar aspect when considering the reallocation constraints.

Table 4 The effect of constraints imposing similarity between the first- and the second-stage solutions

γ	SP	EEV	$100 \times \frac{VSS}{SP}$	WS	$100 \times \frac{EVPI}{SP}$	CPU (s)
0.00	3463.21	3597.21	3.87	3249.18	6.18	534.33
0.85	3464.00	3606.23	4.11	3249.18	6.20	516.58
0.90	3468.80	4184.58	20.63	3249.18	6.33	1405.86
0.95	3473.69	8718.12	150.98	3249.18	6.46	850.02
1.00	3498.56	12,175.12	248.00	3249.18	7.13	2095.78

The relative values of VSS increases significantly when γ increases. This is an indication that the first-stage solution obtained when solving the expected value problem is not good hedging against uncertainty (and thus incurs on much penalties in the second stage). In any case, this is a clear indication that our stochastic programming model produces solutions that can better hedge against uncertainty.

Regarding the CPU time required to solve the model up to proven optimality, we see an increasing trend with the increase in γ . It is interesting to note that for $\gamma = 1.00$ the solver required more than 2000 s of CPU. However, the same solution can be obtained by imposing a maximum number of reallocations $m = 0$, which requires only 27 s.

6.4 Computational results for the tested instances

Having presented some detailed results for one instance, we proceeded with results obtained by solving the mathematical model proposed for the SDPR (M_2) for all the generated instances described in Sect. 6.1. For obvious reasons we do not present the solutions obtained. Instead, we focus on measures that help us to quantify the relevance of capturing stochasticity in our problem: the relative values of VSS and EVPI w.r.t. SP.

The results for the relative VSS are depicted in Fig. 6. The values used in this figure can be found in Tables 7 and 8 that we include in the appendix.

One expected conclusion from observing Fig. 6 is that we obtain larger values for $\omega = 2$. As explained above, the optimal solution to the expected value problem does not change with ω (in a single scenario, no relocation occurs because we can directly plan optimally for that scenario). Therefore, when that solution is considered as a first-stage solution for the stochastic problem, relocations cost more when $\omega = 2$ than when $\omega = 1$. This explains why the optimal solution to the expected value problem provides a better approximation for the stochastic problem when $\omega = 1$. Overall, we observe an average $100 \times VSS/SP$ equal to 1.29 for $\omega = 1$ and 3.60 for $\omega = 2$. Additionally, this value decreases with an increase in p . This means that when more districts are considered, the expected value problem provides a better approximation to the stochastic problem. This indicates that with a larger number of districts we typically observe less reassignments required.

When we focus our analysis on the pairs (ω, p) , we observe that probability distribution 1 seems to “dominate” the others in terms of VSS. This is an indication that a stochastic approach is less effective when one scenario is more likely than the others.

When we focus our attention on the pairs (k, p) , we see that the value of the stochastic solution is typically higher for $\alpha = 0.20, 0.25$. On average, $100 \times VSS/SP$ is equal to 14.32% when $\alpha = 0.20$ and $\omega = 2$. Given the generated demand scenarios, on one hand, the high penalties paid for demand shortages or surplus make the two approaches equivalent when $\alpha \leq 0.15$. On the other, no significant differences are detected when balancing constraints are

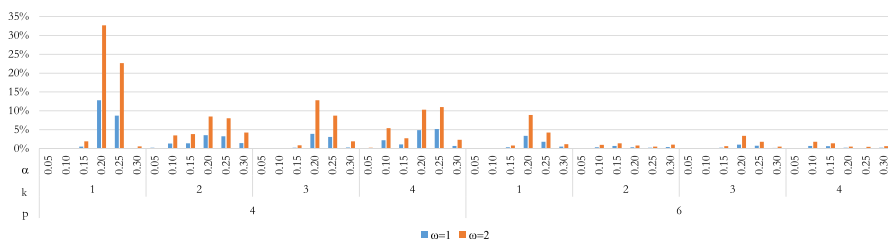


Fig. 6 The relative value of VSS w.r.t. SP (%)

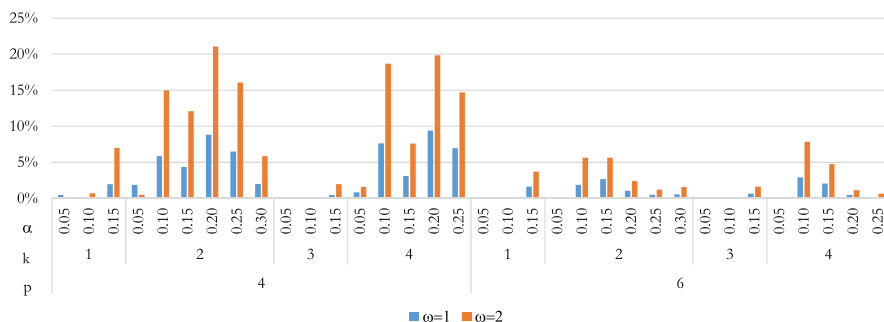


Fig. 7 The compactness value of the stochastic solution (%)

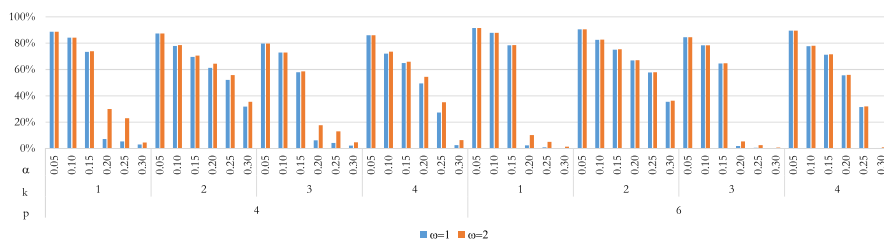


Fig. 8 The relative value of EVPI w.r.t. SP (%)

loose ($\alpha = 0.30$). Not by chance, all the instances characterized by a null VSS are associated with these values of α .

The above comments suggest a deeper look into the solutions we are obtaining. In our objective function we are considering the (re)assignment costs as well as the penalty costs for shortage/surplus demands w.r.t. the reference values. Since the penalty costs are high, they may blur a solution feature of great relevance to us: compactness. Therefore, we decided to take the instances for which penalties are observed and focus only on the (re)assignment costs which, as we have explained before, are a reliable measure of the solution compactness. We computed a measure that we call the Compactness Value of the Stochastic Solution (CVSS). It is computed like the VSS (i.e. for the same solutions) but ignoring the penalty costs (i.e. setting them to 0). The results can be observed in Fig. 7. Overall, these results show that although the values VSS are rather small, in terms of compactness, the expected value solution provides a poor approximation to the stochastic problem.

In Fig. 8 we present the results obtained for the relative values of EVPI w.r.t. the optimal value of the stochastic problem. The detailed results can be found in the Appendix—Tables 7 and 8. Our computational experience reveals quite significant values for $100 \times EVPI/SP$. On average, it is equal to 50.00% for $\omega = 1$ and 52.36% for $\omega = 2$. These high values give an indication that capturing uncertainty in our districting problem is of great relevance. Moreover, the behavior of the EVPI is clear: it is rather insensitive to the adopted value of p and it shows a decreasing trend with α . Not surprisingly, the lower the parameter α the higher the risk of observing demand surplus or shortages. Therefore, a decision maker would be willing to pay a higher price to know perfect information about the future.

Finally we report on the CPU time required to solve our stochastic programming model (M_2). This information is depicted in Fig. 9 with the details provided in the Appendix—Tables 7 and 8.

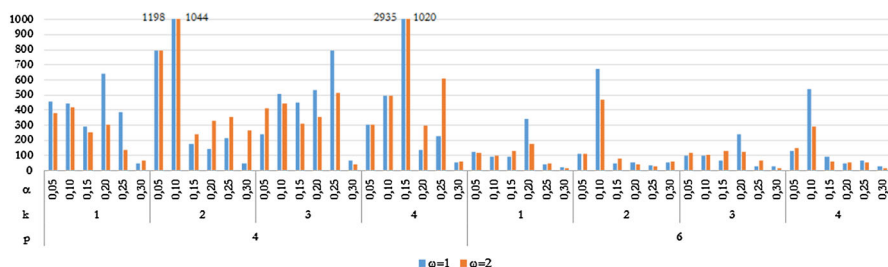


Fig. 9 CPU time (seconds)

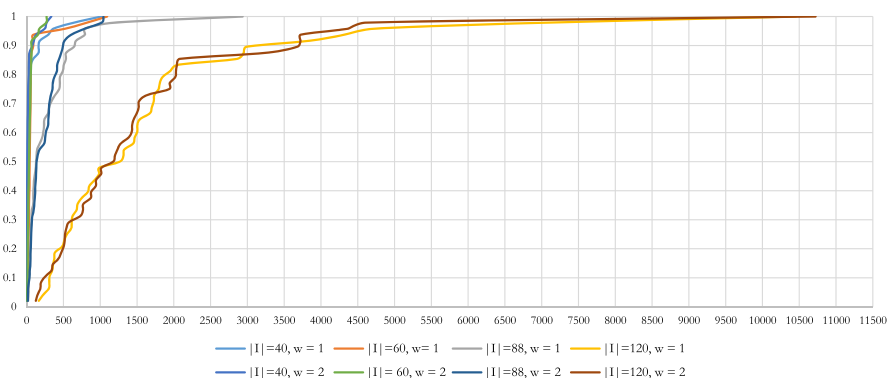


Fig. 10 Performance evaluation regarding the CPU time (seconds) required to solve the instances

Overall, we conclude that the model could be solved for all the instances within an acceptable CPU time. Most of the instances were solved in less than 1000 s and often much below that. The extreme cases can be devised in Fig. 9.

The CPU times observed show that a stochastic districting problem such as the one that we are investigating in this paper can be solved to optimality using tools that are available to most practitioners. We also note that the instances with $p = 6$ seem “easier” to solve than those with $p = 4$. We can also observe a tendency for higher CPU times with $\omega = 1$ than with $\omega = 2$. For the former we observed an average of 306 s while for the latter we observed 249 s. The averages per each value of p can be found in the Appendix.

6.5 Extended results: computational performance of the model

So far in terms of computational results we have focused on a single geographical data corresponding to the 88 municipalities of the province of Novara in Northern Italy. We extend now this analysis by considering the 120-, 60-, and 40-TU instances that were also built. In terms of the tractability of the model (M_2) we focused in the CPU time required to solve it to proven optimality. The detailed results can be found in the Appendix—Tables 9 and 10. In Fig. 10 we depict the performance analysis.

The most significant aspect to highlight in Fig. 10 is that the stochastic model (M_2) could be solved to proven optimality for all instances: approximately 95% of the largest instances could be solved in less than 1 hour of CPU. Apart from this, we see that changing the value of the relocation costs ω does not have a significant impact in the performance of the model.

7 Conclusions

In this paper we investigated a stochastic districting problem triggered by uncertainty in the demand vector. Assuming that uncertainty can be captured by a finite set of scenarios each of which occurring with a given probability, a mixed-integer two-stage stochastic programming framework with some variants and extensions was developed. Computational tests were performed with instances built from real geographical data—and thus instances of a realistic size.

The results show that all instances could be solved to proven optimality using a general purpose solver. This aspect is of particular relevance for a practitioner who may be interested in this type of problems but who does not master sophisticated stochastic programming tools. Moreover, by making use of appropriate measures, the computational results also show that capturing uncertainty may be of great relevance in the districting problems studied.

The work presented suggests several future research directions. First, some discussion provided in this work shows the multicriteria nature of some districting problems. This calls for investigating multicriteria stochastic programming variants of the problems we have investigated in this work. Another aspect that may be interesting to investigate concerns the possibility of capturing uncertainty using chance constraints. This means that instead of modeling the balancing requirements as hard constraints, a (high) probability can be imposed to the satisfaction of these constraints. This possibility leads to a totally different modeling framework which, nonetheless, is worth investigating.

Finally, we note that we have dealt with uncertainty by adopting stochastic programming models. This implies that uncertainty can be captured by some joint cumulative distribution function. If this is not the case, then we may need to resort to other possibilities such as robust optimization. This may also be a promising research direction.

Acknowledgements This work was partially supported by National Funding from FCT - Fundação para a Ciência e a Tecnologia, under the project: UIDB/04561/2020. The authors would like to thank the anonymous reviewer for his/her detailed comments on our work, which helped us improving the manuscript.

Appendix

In this Appendix we details the information concerning the test data used to realize our computational experiments and we present the detailed results that were reported in Sect. 6.

For each TU, Table 5 reports the generated demand vector and the coordinates of the respective centroid. The original code assigned by ISTAT to uniquely identify TUs (i.e. PRO_COM) is also reported. TUs constituting each of the considered instances are indicated in Table 6.

For each tested instance, Tables 7 and 8 contain the relative values of VSS and EVPI w.r.t. SP as well as the CPU time in seconds required by the general purpose solver to solve the instance to optimality. Tables 7 refers to the instances with 4 districts and Tables 8 to instances with 6 districts.

Finally, Tables 9 and 10 contain the CPU time (seconds) required to solve the instances to proven optimality for different values of $|I|$ (40, 60, 88, and 120).

Table 5 Demands and coordinates of the centroids (Coordinate Reference System: ED50 / UTM Zone 32N EPSG:23032)

ID	PRO_COM	d_{is}			Coordinates	
		$s = 1$	$s = 2$	$s = 3$	East	North
1	3001	6.75	8.44	10.13	466469.14504	5057067.1243
2	3002	5.89	7.36	8.84	457360.80298	5070654.0651
3	3006	1.46	1.82	2.18	458367.29932	5075525.7021
4	3008	3.36	4.20	5.04	464446.58482	5067475.4455
5	3012	2.49	3.11	3.73	461875.07870	5046430.6506
6	3016	7.89	9.87	11.84	472099.39165	5044532.8846
7	3018	7.69	9.62	11.54	458985.68591	5033119.2462
8	3019	7.19	8.99	10.79	453880.03626	5058773.2059
9	3021	1.06	1.33	1.60	463746.59461	5056614.7870
10	3022	2.17	2.71	3.25	457024.24652	5067694.9180
11	3023	5.52	6.91	8.29	475867.80907	5018179.9833
12	3024	7.57	9.47	11.36	458699.59655	5060623.5753
13	3025	7.86	9.82	11.78	468748.77429	5059930.7118
14	3026	2.48	3.10	3.72	457842.88085	5064441.1431
15	3027	7.04	8.80	10.56	461282.20632	5042304.8950
16	3030	1.76	2.21	2.65	466912.71807	5041541.5946
17	3032	4.50	5.62	6.75	473850.03066	5040149.0316
18	3036	7.25	9.06	10.87	454699.16093	5043248.9589
19	3037	4.98	6.22	7.46	458991.70292	5030668.8422
20	3039	7.10	8.87	10.64	461166.84122	5037284.9291
21	3040	6.94	8.67	10.41	463005.20972	5025693.9844
22	3041	1.63	2.04	2.44	458378.23864	5027594.6762
23	3042	6.77	8.46	10.15	460503.39529	5039893.8916
24	3043	2.56	3.21	3.85	470498.76541	5062038.8052
25	3044	7.03	8.79	10.55	460868.97196	5049643.6911
26	3045	7.87	9.83	11.80	458950.25351	5050881.8889
27	3047	2.61	3.26	3.91	453526.82528	5056717.3786
28	3049	5.17	6.46	7.75	483575.37782	5027888.5133
29	3051	5.45	6.81	8.17	460382.95820	5071036.2688
30	3052	3.91	4.89	5.87	465786.31118	5062579.1039
31	3055	6.07	7.58	9.10	461747.81832	5054842.8247
32	3058	5.42	6.77	8.13	457533.19728	5057309.6913
33	3060	6.13	7.67	9.20	468713.20787	5055875.8227
34	3062	0.86	1.08	1.29	466864.84219	5064613.2523
35	3065	1.94	2.43	2.92	457701.83486	5045208.0775
36	3066	7.86	9.83	11.79	458575.19316	5054272.8637
37	3068	1.51	1.89	2.27	477534.51025	5036924.0803

Table 5 continued

ID	PRO_COM	d_{is}			Coordinates	
		$s = 1$	$s = 2$	$s = 3$	East	North
38	3069	3.11	3.89	4.67	473442.27320	5026539.8587
39	3070	5.22	6.52	7.83	454155.87151	5063674.1402
40	3071	2.06	2.57	3.09	462422.41720	5062293.6956
41	3073	1.68	2.10	2.52	455155.14409	5049449.1661
42	3076	2.00	2.50	3.00	455550.01313	5066322.1538
43	3077	3.51	4.38	5.26	467749.87556	5024050.9054
44	3079	6.21	7.76	9.31	448971.22997	5059702.1691
45	3082	1.68	2.10	2.52	459582.26479	5067240.2467
46	3083	1.03	1.28	1.54	454760.04052	5038118.2791
47	3084	5.33	6.67	8.00	465893.29552	5074607.1588
48	3088	7.05	8.82	10.58	454327.56414	5060819.1108
49	3090	3.36	4.20	5.03	457952.80278	5038661.6660
50	3091	2.26	2.83	3.39	471150.33893	5053044.6259
51	3093	3.41	4.27	5.12	463470.02560	5074904.4577
52	3095	2.51	3.13	3.76	463306.10194	5070284.7056
53	3097	3.14	3.92	4.71	468822.98580	5051981.1486
54	3098	2.43	3.04	3.65	456024.65705	5072731.6257
55	3100	7.23	9.03	10.84	466041.49339	5046326.8202
56	3103	4.12	5.15	6.18	462321.93172	5072918.0747
57	3104	7.79	9.74	11.69	472386.06706	5024115.7316
58	3106	0.96	1.20	1.44	469940.80509	5031855.8341
59	3108	5.80	7.25	8.70	472407.19707	5049331.7995
60	3109	4.14	5.17	6.20	463432.25366	5065757.3825
61	3112	4.73	5.92	7.10	454556.77546	5070880.6136
62	3114	4.06	5.07	6.09	461944.95197	5066077.7231
63	3115	5.30	6.63	7.95	452274.44434	5073187.9210
64	3116	6.57	8.22	9.86	453719.36938	5075116.2267
65	3119	7.73	9.66	11.59	461721.07286	5071260.5003
66	3120	1.65	2.06	2.47	451532.46926	5067352.7839
67	3121	3.80	4.75	5.70	472451.22094	5054797.6926
68	3122	0.97	1.21	1.46	450911.25600	5056848.3821
69	3129	7.82	9.78	11.73	455062.48449	5034226.2727
70	3130	3.78	4.72	5.67	453178.69320	5053187.4802
71	3131	5.52	6.90	8.28	480191.40095	5034504.9385
72	3133	1.10	1.38	1.65	452854.75017	5069497.8989
73	3134	6.25	7.82	9.38	454650.75588	5030566.1747
74	3135	2.38	2.98	3.57	464109.73496	5035165.9598
75	3138	4.76	5.95	7.14	456097.21856	5040500.8095
76	3139	4.81	6.01	7.22	456553.86665	5047228.1749
77	3140	1.88	2.35	2.82	453174.03926	5064999.2855

Table 5 continued

ID	PRO_COM	d_{is}			Coordinates	
		$s = 1$	$s = 2$	$s = 3$	East	North
78	3141	5.26	6.58	7.90	478459.44218	5026931.1238
79	3143	2.84	3.55	4.26	464592.96713	5052731.0257
80	3144	4.83	6.04	7.25	476071.80757	5024724.1517
81	3146	6.40	8.00	9.60	478582.56269	5022122.4477
82	3149	7.53	9.42	11.30	479715.00701	5031009.2853
83	3153	1.93	2.42	2.90	465341.01996	5050075.1151
84	3154	5.20	6.51	7.81	472051.37971	5057268.5926
85	3157	6.68	8.35	10.02	464398.05591	5059931.6808
86	3158	7.76	9.70	11.64	473466.51007	5021464.4636
87	3159	1.06	1.33	1.59	457661.23491	5035969.1381
88	3164	1.28	1.60	1.92	460929.96937	5019673.6840
89	2003	2.75	3.44	4.13	451587.72756	5030408.5340
90	2006	4.55	5.68	6.82	451189.24473	5037731.9913
91	2016	1.39	1.74	2.08	442402.90635	5064408.0814
92	2061	2.10	2.62	3.15	449891.01426	5050281.1567
93	2062	7.42	9.28	11.13	451158.19037	5041667.9385
94	2065	7.68	9.60	11.53	451590.18473	5033778.3472
95	2068	4.44	5.55	6.66	450614.65991	5045115.2207
96	2089	1.58	1.98	2.37	451961.72239	5027182.4993
97	2137	3.36	4.20	5.04	446192.63681	5057878.2684
98	2152	4.54	5.67	6.81	449317.83691	5064083.3625
99	12003	5.15	6.44	7.73	468430.89537	5069222.6098
100	103040	2.58	3.22	3.87	449061.30687	5070968.4963
101	103048	2.53	3.16	3.80	450775.35735	5077946.2349
102	103050	1.88	2.35	2.82	454800.43809	5079842.5669
103	103064	2.84	3.56	4.27	463195.29007	5081605.6265
104	12077	2.20	2.75	3.30	473815.55655	5060439.6383
105	12084	3.13	3.91	4.69	469880.75965	5073430.2163
106	103004	4.20	5.25	6.31	449056.65847	5073774.0274
107	103010	4.87	6.09	7.30	467316.57267	5076966.6717
108	103013	5.96	7.45	8.94	463278.73221	5077379.6383
109	103022	3.14	3.92	4.71	449945.46283	5075647.1474
110	103034	6.31	7.89	9.47	461098.38868	5079134.5808
111	12090	6.49	8.11	9.74	479306.44706	5048536.7730
112	12116	6.17	7.71	9.25	466978.51711	5071665.1749
113	12120	1.32	1.65	1.98	471629.51782	5065263.6445
114	12123	2.10	2.62	3.14	476531.35641	5057331.5110
115	12140	1.54	1.92	2.31	475512.05575	5052567.0634
116	15062	4.90	6.13	7.35	481758.26552	5044110.0784

Table 5 continued

ID	PRO_COM	d_{is}			Coordinates	
		$s = 1$	$s = 2$	$s = 3$	East	North
117	15096	1.14	1.42	1.71	484316.24048	5039026.0351
118	15155	6.04	7.56	9.07	478227.02345	5044640.2035
119	15183	5.09	6.36	7.63	481459.00295	5040243.0124
120	15226	0.86	1.07	1.29	479315.12052	5041507.7121

Table 6 TUs that are part of each instance

ID	$ I = 40$	$ I = 60$	$ I = 88$	$ I = 120$
1	✓	✓	✓	✓
2		✓	✓	✓
3			✓	✓
4	✓	✓	✓	✓
5	✓	✓	✓	✓
6		✓	✓	✓
7			✓	✓
8	✓	✓	✓	✓
9	✓	✓	✓	✓
10		✓	✓	✓
11			✓	✓
12	✓	✓	✓	✓
13	✓	✓	✓	✓
14	✓	✓	✓	✓
15	✓	✓	✓	✓
16		✓	✓	✓
17			✓	✓
18		✓	✓	✓
19			✓	✓
20		✓	✓	✓
21			✓	✓
22			✓	✓
23		✓	✓	✓
24	✓	✓	✓	✓
25	✓	✓	✓	✓
26	✓	✓	✓	✓
27	✓	✓	✓	✓
28			✓	✓
29		✓	✓	✓
30	✓	✓	✓	✓
31	✓	✓	✓	✓
32	✓	✓	✓	✓
33	✓	✓	✓	✓

Table 6 continued

ID	$ I = 40$	$ I = 60$	$ I = 88$	$ I = 120$
34	✓	✓	✓	✓
35	✓	✓	✓	✓
36	✓	✓	✓	✓
37			✓	✓
38			✓	✓
39	✓	✓	✓	✓
40	✓	✓	✓	✓
41	✓	✓	✓	✓
42	✓	✓	✓	✓
43			✓	✓
44		✓	✓	✓
45	✓	✓	✓	✓
46		✓	✓	✓
47			✓	✓
48	✓	✓	✓	✓
49		✓	✓	✓
50	✓	✓	✓	✓
51			✓	✓
52		✓	✓	✓
53	✓	✓	✓	✓
54		✓	✓	✓
55	✓	✓	✓	✓
56		✓	✓	✓
57			✓	✓
58			✓	✓
59	✓	✓	✓	✓
60	✓	✓	✓	✓
61		✓	✓	✓
62	✓	✓	✓	✓
63			✓	✓
64			✓	✓
65		✓	✓	✓
66		✓	✓	✓
67	✓	✓	✓	✓
68			✓	✓
69	✓	✓	✓	✓
70			✓	✓
71		✓	✓	✓
72			✓	✓
73			✓	✓
74		✓	✓	✓
75	✓	✓	✓	✓
76		✓	✓	✓

Table 6 continued

ID	$ I = 40$	$ I = 60$	$ I = 88$	$ I = 120$
77			✓	✓
78	✓	✓	✓	✓
79			✓	✓
80			✓	✓
81			✓	✓
82	✓	✓	✓	✓
83	✓	✓	✓	✓
84	✓	✓	✓	✓
85			✓	✓
86			✓	✓
87			✓	✓
88	✓	✓	✓	✓
89				✓
90				✓
91				✓
92				✓
93				✓
94				✓
95				✓
96				✓
97				✓
98				✓
99				✓
100				✓
101				✓
102				✓
103				✓
104				✓
105				✓
106				✓
107				✓
108				✓
109				✓
110				✓
111				✓
112				✓
113				✓
114				✓
115				✓
116				✓

Table 6 continued

ID	$ I = 40$	$ I = 60$	$ I = 88$	$ I = 120$
117				✓
118				✓
119				✓
120				✓

Table 7 Computational results for the 88-TU instances, $p = 4$

k	α	$100 \times VSS/SP$		$100 \times EVPI/SP$		CPU (seconds)	
		$\omega = 1$	$\omega = 2$	$\omega = 1$	$\omega = 2$	$\omega = 1$	$\omega = 2$
1	0.05	0.05	0.00	88.72	88.73	452.72	376.95
	0.10	0.01	0.12	84.28	84.30	441.78	414.91
	0.15	0.54	1.94	73.33	73.85	291.45	252.05
	0.20	12.77	32.66	7.13	29.99	643.20	303.42
	0.25	8.73	22.62	5.38	22.98	386.17	134.56
	0.30	0.00	0.56	2.95	4.39	50.08	66.30
2	0.05	0.23	0.06	87.40	87.42	791.45	791.92
	0.10	1.34	3.43	77.85	78.59	1198.25	1043.63
	0.15	1.38	3.83	69.49	70.63	176.98	238.45
	0.20	3.54	8.48	61.31	64.38	141.77	328.81
	0.25	3.24	8.06	52.07	55.82	211.97	354.64
	0.30	1.43	4.22	31.68	35.51	44.91	262.92
3	0.05	0.01	0.00	79.74	79.74	241.64	412.03
	0.10	0.00	0.02	72.83	72.86	504.72	441.26
	0.15	0.21	0.85	57.99	58.54	451.30	311.66
	0.20	3.87	12.77	6.18	17.64	534.33	351.13
	0.25	3.05	8.73	4.10	12.98	795.24	514.28
	0.30	0.31	1.93	2.15	4.79	69.69	40.03
4	0.05	0.12	0.22	86.04	86.08	305.08	301.78
	0.10	2.21	5.40	72.06	73.49	492.23	491.58
	0.15	1.11	2.73	64.97	65.98	2935.17	1020.39
	0.20	4.85	10.26	49.52	54.41	135.13	294.61
	0.25	5.20	11.01	27.23	35.19	226.77	607.14
	0.30	0.70	2.32	2.58	6.30	54.39	59.75
Average		2.29	5.93	48.62	52.69	482.35	392.26

Table 8 Computational results for the 88-TU instances, $p = 6$

k	α	$100 \times VSS/SP$		$100 \times EVPI/SP$		CPU (seconds)	
		$\omega = 1$	$\omega = 2$	$\omega = 1$	$\omega = 2$	$\omega = 1$	$\omega = 2$
1	0.05	0.00	0.00	91.56	91.56	124.09	116.30
	0.10	0.00	0.00	87.90	87.90	90.23	97.03
	0.15	0.36	0.81	78.46	78.64	90.27	131.25
	0.20	3.36	8.91	2.20	10.21	338.22	174.81
	0.25	1.77	4.24	0.79	4.92	41.31	50.95
	0.30	0.50	1.19	0.19	1.37	19.30	16.77
2	0.05	0.01	0.01	90.55	90.55	111.53	111.08
	0.10	0.34	1.00	82.58	82.75	671.36	471.08
	0.15	0.67	1.42	75.05	75.41	46.11	76.63
	0.20	0.35	0.80	66.91	67.18	55.89	41.41
	0.25	0.22	0.51	57.74	57.96	36.43	29.11
	0.30	0.36	1.02	35.56	36.26	51.05	61.75
3	0.05	0.00	0.00	84.44	84.44	100.78	116.73
	0.10	0.00	0.00	78.41	78.41	95.83	102.45
	0.15	0.24	0.59	64.58	64.80	66.50	127.88
	0.20	1.02	3.36	1.77	5.38	237.00	125.00
	0.25	0.76	1.77	0.53	2.52	25.75	70.08
	0.30	0.15	0.50	0.19	0.69	27.59	16.91
4	0.05	0.02	0.00	89.47	89.48	127.89	149.48
	0.10	0.66	1.79	77.63	78.03	535.90	291.39
	0.15	0.61	1.38	71.11	71.51	90.73	57.45
	0.20	0.22	0.52	55.70	55.93	48.27	51.61
	0.25	0.13	0.45	31.49	31.88	64.66	52.39
	0.30	0.20	0.60	0.25	0.88	27.72	19.05
Average		0.50	1.29	51.04	52.02	130.18	106.61

Table 9 CPU time (seconds)—for all the instances with $p = 4$

k	α	$\omega = 1$				$\omega = 2$			
		$ I = 40$	$ I = 60$	$ I = 88$	$ I = 120$	$ I = 40$	$ I = 60$	$ I = 88$	$ I = 120$
1	0.05	5.78	56.80	452.72	1504.39	4.13	67.83	376.95	1430.00
	0.10	13.45	40.73	441.78	2048.06	5.75	37.66	414.91	2021.13
	0.15	5.59	48.47	291.45	1809.41	4.56	44.05	252.05	1527.80
	0.20	6.17	43.04	643.20	4739.06	2.72	30.28	303.42	4621.57
	0.25	3.20	31.36	386.17	1700.81	1.50	10.56	134.56	1632.21
2	0.30	4.08	16.25	50.08	678.05	1.97	7.17	66.30	728.00
	0.05	6.69	36.67	791.45	1847.67	6.78	35.86	791.92	1454.39
	0.10	14.97	51.17	1198.25	7100.00	7.89	50.19	1043.63	3676.97
	0.15	2.91	39.38	176.98	1728.00	2.06	22.97	238.45	1199.66
	0.20	3.97	32.59	141.77	1326.20	2.95	13.94	328.81	1175.73
3	0.25	3.92	47.88	211.97	607.25	2.53	23.72	354.64	1496.81
	0.30	2.97	32.30	44.91	1953.08	1.84	22.97	262.92	876.34
	0.05	4.58	33.22	241.64	1793.00	5.11	34.66	412.03	2043.00
	0.10	9.11	39.63	504.72	1468.08	6.73	36.08	441.26	1944.70
	0.15	4.05	62.28	451.30	2988.20	3.36	39.70	311.66	2029.80
4	0.20	3.98	50.39	534.33	2950.28	4.50	26.75	351.13	3254.94
	0.25	3.97	31.16	795.24	1734.61	1.64	13.88	514.28	1017.47
	0.30	4.58	30.51	69.69	985.70	1.94	14.99	40.03	540.09
	0.05	11.22	54.67	305.08	1266.38	7.02	47.73	301.78	1936.88
	0.10	16.27	87.75	492.23	4323.11	6.97	267.69	491.58	3721.40
	0.15	4.27	45.44	2935.17	3849.17	2.72	31.00	1020.39	2081.75
	0.20	3.41	48.88	135.13	2860.88	2.39	30.17	294.61	1522.99
	0.25	4.94	49.42	226.77	926.51	2.63	49.00	607.14	939.97
	0.30	3.59	28.78	54.39	346.50	1.52	20.61	59.75	435.06
	Average	6.15	43.28	482.35	2188.93	3.80	40.81	392.26	1804.56

Table 10 CPU time (seconds)—for all the instances with $p = 6$

k	α	$\omega = 1$				$\omega = 2$			
		$ I = 40$	$ I = 60$	$ I = 88$	$ I = 120$	$ I = 40$	$ I = 60$	$ I = 88$	$ I = 120$
1	0.05	36.53	55.70	124.09	749.31	13.25	54.19	116.30	1234.09
	0.10	40.50	23.38	90.23	624.00	22.41	20.19	97.03	769.69
	0.15	41.41	44.06	90.27	611.61	29.53	58.08	131.25	944.88
	0.20	179.38	1093.78	338.22	1675.09	71.91	173.48	174.81	4372.52
	0.25	161.65	39.63	41.31	385.58	98.09	23.48	50.95	250.59
2	0.30	89.66	12.19	19.30	225.16	15.78	5.45	16.77	124.86
	0.05	294.34	91.05	111.53	484.88	36.66	66.47	111.08	517.17
	0.10	645.94	528.83	671.36	1504.47	270.36	271.17	471.08	3727.34
	0.15	14.89	25.67	46.11	376.03	11.31	52.30	76.63	521.77
	0.20	16.47	12.64	55.89	308.31	10.97	6.52	41.41	355.19
3	0.25	36.20	19.80	36.43	165.05	18.64	18.91	29.11	147.90
	0.30	31.22	21.67	51.05	310.53	29.19	20.00	61.75	337.55
	0.05	11.50	55.88	100.78	1537.36	13.55	47.13	116.73	1277.75
	0.10	14.58	19.36	95.83	692.69	16.08	15.28	102.45	1013.52
	0.15	55.42	39.47	66.50	988.19	16.42	39.08	127.88	871.19
4	0.20	1016.36	93.67	237.00	1315.91	334.66	153.19	125.00	1385.94
	0.25	38.20	43.45	25.75	547.09	128.86	29.02	70.08	471.08
	0.30	13.25	20.67	27.59	358.36	18.69	12.67	16.91	193.92
	0.05	14.78	56.39	127.89	1447.39	15.33	55.06	149.48	1433.27
	0.10	368.34	822.94	535.90	10,452.94	246.78	68.84	291.39	10,721.34
	0.15	52.20	57.05	90.73	852.51	32.39	61.75	57.45	503.80
	0.20	22.64	20.19	48.27	511.06	20.98	13.28	51.61	576.40
	0.25	162.17	53.23	64.66	831.61	28.70	63.17	52.39	765.39
	0.30	5.58	8.75	27.72	300.52	8.80	13.09	19.05	186.28
	Average	140.13	135.81	130.18	1135.65	62.89	55.91	106.61	1362.64

References

- Akdoğan, M. A., Bayındır, Z. P., & Iyigun, C. (2018). Locating emergency vehicles with an approximate queuing model and a meta-heuristic solution approach. *Transportation Research Part C: Emerging Technologies*, 90, 134–155.
- Bard, J. F., & Jarrah, A. I. (2009). Large-scale constrained clustering for rationalizing pickup and delivery operations. *Transportation Research Part B: Methodological*, 43(5), 542–561.
- Bender, M., Kalcsics, J., Nickel, S., & Poulos, M. (2018). A branch-and-price algorithm for the scheduling of customer visits in the context of multi-period service territory design. *European Journal of Operational Research*, 269(1), 382–396.
- Bender, M., Meyer, A., Kalcsics, J., & Nickel, S. (2016). The multi-period service territory design problem—an introduction, a model and a heuristic approach. *Transportation Research Part E: Logistics and Transportation Review*, 96, 135–157.
- 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.
- Bergey, B. K., Ragsdale, C. T., & Hoskote, M. (2003). A simulated annealing genetic algorithm for the electrical power districting problem. *Annals of Operations Research*, 121(1–4), 33–55.
- Bianchi, G., Bruni, R., Reale, A., & Sforzi, F. (2016). A min-cut approach to functional regionalization, with a case study of the Italian local labour market areas. *Optimization Letters*, 10(5), 955–973.
- Birge, J. R., & Louveaux, F. V. (2011). *Introduction to Stochastic Programming* (2nd ed.). New York: Springer.
- Blais, M., Lapierre, S. D., & Laporte, G. (2003). Solving a home-care districting problem in an urban setting. *Journal of the Operational Research Society*, 54(11), 1141–1147.
- Bozkaya, B., Erkut, E., & Laporte, G. (2003). A tabu search heuristic and adaptive memory procedure for political districting. *European Journal of Operational Research*, 144(1), 12–26.
- 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). Cham: Springer.
- Bruno, G., Esposito, E., Genovese, A., & Piccolo, C. (2016a). Institutions and facility mergers in the Italian education system: Models and case studies. *Socio-Economic Planning Sciences*, 53, 23–32.
- Bruno, G., Genovese, A., & Piccolo, C. (2016b). Capacity management in public service facility networks: A model, computational tests and a case study. *Optimization Letters*, 10(5), 975–995.
- 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.
- Bruno, G., & Laporte, G. (2002). An interactive decision support system for the design of rapid public transit networks. *INFOR: Information Systems and Operational Research*, 40(2), 111–118.
- 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.
- Carlsson, J. G., & Delage, E. (2013). Robust partitioning for stochastic multivehicle routing. *Operations Research*, 61(3), 727–744.
- 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.
- Correia, I., & Saldanha-da-Gama, F. (2019). Facility location under uncertainty. In: Laporte G, Nickel S, Saldanha-da-Gama F (Eds.) *Location Science*, 2nd edn. Cham: Springer, Chapter 8, pp. 185–213.
- D’Amico, S. J., Wang, S. J., Batta, R., & Rump, C. M. (2002). A simulated annealing approach to police district design. *Computers & Operations Research*, 29(6), 667–684.
- 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.
- De Fréminville, P. D. L. P., Desaulniers, G., Rousseau, L. M., & Perron, S. (2015). A column generation heuristic for districting the price of a financial product. *Journal of the Operational Research Society*, 66(6), 965–978.
- ESPAS. (2015). *Global Trends to 2030: Can the EU meet the challenges ahead?* www.ec.europa.eu. Retrieved September 2018.
- Ferland, J., & Guénette, G. (1990). Decision support system for the school districting problem. *Operations Research*, 38(1), 15–21.

- García-Ayala, G., González-Velarde, J. L., Ríos-Mercado, R. Z., & Fernández, E. (2016). A novel model for arc territory design: Promoting Eulerian districts. *International Transactions in Operational Research*, 23(3), 433–458.
- Haugland, D., Ho, S. C., & Laporte, G. G. (2007). Designing delivery districts for the vehicle routing problem with stochastic demands. *European Journal of Operational Research*, 180(3), 997–1010.
- 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.
- ISTAT. (2011). *Dati del censimento generale della popolazione italiana (italian national census data)*. www.istat.it. Retrieved September 2018.
- Kalcsics, J., Nickel, S., & Schröder, M. (2005). Towards a unified territorial design approach—Applications, algorithms and GIS integration. *TOP*, 13, 1–56.
- Kalcsics, J., & Ríos-Mercado, R. (2019). *Districting problems*. In: G. Laporte, S. Nickel, F. Saldanha-da-Gama (Eds.), *Location science*, 2nd edn., Chapter 25. Cham: Springer, pp. 705–743.
- Kim, K., Chun, Y., & Kim, H. (2017). p -Functional clusters location problem for detecting spatial clusters with covering approach. *Geographical Analysis*, 49(1), 101–121.
- Lei, H., Laporte, G., & Guo, B. (2012). Districting for routing with stochastic customers. *EURO Journal on Transportation and Logistics*, 1(1–2), 67–85.
- Lei, H., Laporte, G., Liu, Y., & Zhang, T. (2015). Dynamic design of sales territories. *Computers & Operations Research*, 56, 84–92.
- Lei, H., Wang, R., & Laporte, G. (2016). Solving a multi-objective dynamic stochastic districting and routing problem with a co-evolutionary algorithm. *Computers & Operations Research*, 67, 12–24.
- Lin, M., Chin, K. S., Fu, C., & Tsui, K. L. (2017). An effective greedy method for the meals-on-wheels service districting problem. *Computers & Industrial Engineering*, 106, 1–19.
- Mourão, M. C., Nunes, A. C., & Prins, C. (2009). Heuristic methods for the sectoring arc routing problem. *European Journal of Operational Research*, 196(3), 856–868.
- Ricca, F., & Simeone, B. (2008). Local search algorithms for political districting. *European Journal of Operational Research*, 189(3), 1049–1426.
- Ricca, F., Scozzari, A., & Simeone, B. (2013). Political districting: From classical models to recent approaches. *Annals of Operations Research*, 204(1), 271–299.
- Ríos-Mercado, R. Z. (2016). Assessing a metaheuristic for large-scale commercial districting. *Cybernetics and Systems*, 47(4), 321–338.
- 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., & López-Pérez, J. F. (2013). Commercial territory design planning with realignment and disjoint assignment requirements. *Omega*, 41(3), 525–535.
- Schoepfle, O., & Church, R. (1991). A new network representation of a classic school districting problem. *Socio-Economic Planning Sciences*, 25(3), 189–197.
- Tavares, P., Figueira, F., Mousseau, J., & Roy, V. (2007). Multiple criteria districting problems. The public transportation network pricing system of the Paris region. *Annals of Operations Research*, 154, 69–9.
- Xie, S., & Ouyang, Y. (2016). Railroad caller districting with reliability, contiguity, balance, and compactness considerations. *Transportation Research Part C: Emerging Technologies*, 63, 65–76.
- Yanık, S., Sürer, Ö., & Öztayşi, B. (2016). Designing sustainable energy regions using genetic algorithms and location-allocation approach. *Energy*, 97, 161–172.
- Zhong, H., Hall, R. W., & Dessouky, M. (2007). Territory planning and vehicle dispatching with driver learning. *Transportation Science*, 41(1), 74–89.
- Zoltners, A. A., & Sinha, P. (2005). Sales territory design: Thirty years of modeling and implementation. *Marketing Science*, 24(3), 313–331.