Annals of Operations Research
https://doi.org/10.1007/s10479-024-06459-7

ORIGINAL RESEARCH

Multi-stage stochastic districting: optimization models and solution algorithms

Anika Pomes[1] · Antonio Diglio[2] · Stefan Nickel[1,3] · Francisco Saldanha-da-Gama[4]

Received: 20 October 2023 / Accepted: 23 December 2024

Abstract

This paper investigates a Multi-Stage Stochastic Districting Problem (MSSDP). The goal is to devise a districting plan (i.e., clusters of Territorial Units—TUs) accounting for uncertain parameters changing over a discrete multi-period planning horizon. The problem is cast as a multi-stage stochastic programming problem. It is assumed that uncertainty can be captured by a finite set of scenarios, which induces a scenario tree. Each node in the tree corresponds to the realization of all the stochastic parameters from the root node—the state of nature—up to that node. A mathematical programming model is proposed that embeds redistricting recourse decisions and other recourse actions to ensure that the districts are balanced regarding their activity. The model is tested on instances generated using literature data containing real geographical data. The results demonstrate the relevance of hedging against uncertainty in multi-period districting. Since the model is challenging to tackle using a general-purpose solver, a heuristic algorithm is proposed based on a restricted model. The computational results obtained give evidence that the approximate algorithm can produce high-quality feasible solutions within acceptable computation times.

Keywords Districting · Multi-stage stochastic programming · Heuristics

1 Introduction

A Districting Problem (DP) consists of partitioning a set of basic *Territorial Units* (TUs) into a predefined number of clusters, called *districts*, according to some *planning criteria*. Indeed, several features are usually desired in the solutions devised for these problems, such as: (i) *integrity*, i.e., each TU has to be assigned exactly to one district; (ii) *balancing*, expressing the need for districts of similar size w.r.t. to some *activity measure(s)*—hereafter, the *demand*—

✉ Anika Pomes
anika.pomes@kit.edu

[1] Institute of Operations Research, Karlsruhe Institute of Technology (KIT), Kaiserstr.12, Karlsruhe 76131, Germany

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

[3] Research Center for Information Technology (FZI), Haid-und-Neu-Str. 10-14, Karlsruhe 76131, Germany

[4] Sheffield University Management School, Sheffield S10 1FL, UK

Published online: 17 January 2025

linked with the TUs (e.g., territorial extension, population, demand for goods or services, etc.); (iii) *compactness*, indicating that the shape of a district should be mostly “rounded”, that is, not elongated and undistorted; (iv) *contiguity*, meaning that the districting plan does not contain enclaves. The latter also implies that there is always a path connecting two TUs belonging to the same district that does not cross any other districts.

In the literature, DPs have been successfully exploited to deal with various practical applications, including the definition of administrative boundaries and electoral zones (i.e., *political districting*, see Ricca et al., 2013), strategic service planning (e.g., health-care or school systems, see Bruno et al., 2016; Farughi et al., 2019), sales territory design (Ríos-Mercado and López-Pérez, 2013) and distribution logistics (Bender et al., 2020; Sandoval et al., 2022). For an overview of the topic, the interested reader can refer to Kalcsics and Ríos-Mercado (2019) and Ríos-Mercado (2020). As the latter sources reveal, most of the body of knowledge on DPs focuses on deterministic and single-period settings. However, when looking at the above-discussed applications, it is clear that demands are not always known beforehand. In many cases, in fact, demand is uncertain and has to be dealt with or anticipated accordingly. Furthermore, it is often relevant to also consider a planning horizon during which the underlying parameters change. These two aspects lead to a more realistic and comprehensive representation of practical DPs. In this paper, solution techniques are developed with the goal of hedging against uncertainty, anticipating future trends, and adapting decisions to the changing setting.

Notably, time-dependent decisions and stochastic models have already been covered by some of the existing literature on DPs, although scarcely in an integrated way. Concerning *multi-period districting*, Lei et al. (2015) investigate dynamic travelling salesmen with multiple depots, where customers are grouped in districts, and the travel cost in each district is approximated by the Beardwood-Halton-Hammersley formula (BHH, see Beardwood et al., 1959). Bender et al. (2016, 2018) introduce a multi-period service design problem divided into two subproblems: (i) a partitioning subproblem, in which customers must be grouped into service territories, and (ii) a scheduling subproblem, to determine customer visits over the multi-period planning horizon. Finally, Yanık et al. (2019) consider a multi-period multi-criteria districting model for primary health services, specifically, patient admissions to general practitioners. Regarding *stochastic districting*, the existing literature mostly tackles problems in the context of vehicle routing. Haugland et al. (2007) propose a tabu search and a multi-start heuristic to design delivery districts for vehicle routing problems with stochastic demand. Carlsson and Delage (2013) investigate a distributionally robust optimization framework for route districting with uncertain client locations. Moment-based ambiguity sets are considered for capturing the distribution underlying those locations. Lei et al. (2012) introduce the vehicle routing and districting problem assuming customers with an uncertain location and presence. The problem is modelled as a two-stage stochastic program with a here-and-now districting decision and wait-and-see routes. The BHH formula is used to approximate the (uncertain) driving time. This work has been extended to a multi-period framework in Lei et al. (2016), which is, to the best of the authors’ knowledge, the only article investigating the inclusion of time-dependent decisions in a stochastic districting problem.

Focusing on basic DPs (i.e., with no routing decisions), various modeling perspectives have been investigated in the literature: chance-constrained programming, robust optimization, and stochastic programming. In terms of *chance-constrained programming*, Diglio et al. (2021) develop a sim-heuristic (Juan et al., 2015) to find feasible solutions for a districting problem with balancing requirements treated as probabilistic constraints (the decision maker is satisfied with a solution that satisfies those constraints with at least some prescribed probability). In a follow-up paper, the same authors (Diglio et al., 2023) develop a sampling procedure, which is further enhanced in Baldassarre et al. (2023). Mostafayi Darmian et al. (2021) consider *robust optimization* for dealing with uncertainty in DPs. Demand is represented either by box uncertainty or by an uncertainty set induced by a maximum total scaled variation of the demand with respect to their nominal values (Baron et al., 2011). Finally, the work by Diglio et al. (2020) is a relatively recent study adopting a *stochastic programming* approach for the problem. Specifically, the authors introduce a two-stage stochastic programming model for districting in that reference. In the first stage, a districting plan is devised to seek a set of compact districts; in the second stage, different recourse actions are considered for ensuring balancing, such as reassigning TUs or implementing other specific actions to overcome shortage or surplus at the districts. The authors consider other aspects of practical relevance, such as a maximum dispersion allowed and similarity of a redistricting plan w.r.t. the original (here-and-now) districting solution. The reader is also addressed to Saldanha-da-Gama and Wang (2024) for a comprehensive discussion on the topic.

The work by Diglio et al. (2020) is set at the core of the developments proposed in this manuscript. In fact, the present paper extends the modeling framework introduced therein into a multi-period setting. The aim is to devise a multi-stage decision-making process in stochastic districting. The formulation starts with the deterministic version of the problem, which is then extended to embed stochasticity in demands and costs. The recourse actions in different periods include redistricting (i.e., reassignments of TUs) and other recourse actions to adapt the decisions to the observed parameters, thus ensuring that the districts are balanced. To the best of the authors' knowledge, no previous works have dealt with such a challenge in a classical DP, and an explicit mathematical formulation for a multi-period DP under uncertainty is missing. As such, the major contributions provided by this work are the following: (i) proposing a novel and general modelling framework for Multi-Stage Stochastic Districting Problems (MSSDPs); (ii) demonstrating, through extensive experiments on instances built on real geographical data, that the problem leads to meaningful solutions and that capturing stochasticity is relevant in the investigated problem. Besides, a heuristic approach is proposed to obtain approximate solutions to the problem. Indeed, as experienced, the stochastic programming resulting from the proposed extension can hardly be tackled efficiently using a general-purpose solver or even specially tailored exact algorithms (see, e.g., Aldasoro et al., 2013; Escudero et al., 2010, 2012). The devised algorithm seeks to solve a reduced model obtained by considering a restricted set of "promising" candidate representatives for the districts. The latter is obtained using information from the solution of the linear relaxation of the model. The algorithm proves effective in producing high-quality solutions at a reasonable computational effort, thus representing a first successful idea for the problem at hand that paves the way for future algorithmic developments.

The remainder of this paper is organized as follows. In the next section, the deterministic multi-period version of the problem is introduced. In Sect. 3, uncertainty is embedded in that model. The heuristic proposed is described in Sect. 4. In Sect. 5, computational results are reported. The article ends with an overview of the work and future research perspectives.

2 An optimization model for multi-period districting

The districting problem comprises a set of TUs whose demand for some commodity or service varies throughout a planning horizon divided into a finite number of time periods. For now, it is assumed that such variation can be accurately predicted. The objective is to partition those TUs into a given number of districts in each period. The goal is to obtain compact districts. As customary in the districting literature, a surrogate measure for compactness is used, namely the total cost (e.g., road distances) for assigning the TUs to the defined districts. Due to the time-dependent demand, in each time period, districts may be reorganized, i.e., some TUs may be reassigned, which, nonetheless, is assumed to incur some cost. Such a cost is assumed to result from assigning a TU to a district it was not previously assigned to, minus the saving for removing the TU from its previous district. Districts must be balanced when their activity does not deviate from a reference value more than the prescribed amount. Accordingly, as an alternative to reassigning some TUs, it may be preferable to incur an extra cost for corrective measures that handle shortage or surplus at a district depending on the case (e.g., outsourcing or downsizing the workforce). Each district has a representative TU that may change from one period to another. Hence, when a TU is assigned to a district, the language is abused by saying that a TU is being assigned to the district's representative. It is noted that single-assignment is assumed for the TUs as customary in districting problems, thus ensuring integrity — a common feature relevant in practice.

Aiming at developing a mathematical formulation for the problem, the following sets are introduced:

I, set of TUs.
TP, set of time periods, $TP = \{1, \ldots, T\}$.

Additionally, the parameters that define the problem are introduced:

p_t, number of districts to operate in period $t \in TP$.
d_{it}, demand of TU $i \in I$ in period $t \in TP$.
μ_t, reference value for the demand assigned to each district in period $t \in TP$. This value is defined as $\mu_t = \frac{1}{p_t} \sum_{i \in I} d_{it}$ $(t \in TP)$, i.e., the exact demand assigned to each district if the districts are perfectly balanced—which may not be possible due to the single assignment assumption.
α, allowed deviation for the demand assigned to each district in each period w.r.t. the reference value in that period.
c_{ij}, initial cost for assigning TU $i \in I$ to TU $j \in I$ (period 1).
r_{ijt}, cost for reassigning TU $i \in I$ to TU $j \in I$ in (the beginning of) period $t \in TP \setminus \{1\}$.
s_{ijt}, saving for removing TU $i \in I$ from district $j \in I$ in (the beginning of) period $t \in TP \setminus \{1\}$.
g_{jt}, unit penalty for surplus at district $j \in I$ in period $t \in TP$ w.r.t. the maximum deviation stated by α.
h_{jt}, unit penalty for shortage at district $j \in I$ in period $t \in TP$ w.r.t. the maximum deviation stated by α.

The multi-period districting problem underlying this work can be formulated using the following sets of variables:

$$x_{ijt} = \begin{cases} 1, & \text{if TU } i \text{ is assigned to TU } j \\ & \text{in (the beginning of) period } t; \\ 0, & \text{otherwise.} \end{cases} \qquad (i, j \in I, t \in TP)$$

$$v_{ijt} = \begin{cases} 1, & \text{if TU } i \text{ is reassigned to TU } j \text{ in (the beginning of) period } t; \\ 0, & \text{otherwise.} \end{cases}$$
$$(i, j \in I, t \in TP \setminus \{1\})$$

$$w_{ijt} = \begin{cases} 1, & \text{if TU } i \text{ is removed from district } j \text{ in (the beginning of) period } t; \\ 0, & \text{otherwise.} \end{cases}$$
$$(i, j \in I, t \in TP \setminus \{1\})$$

ψ_{jt} = demand surplus in district j in period t (w.r.t. the upper threshold), $j \in I, t \in TP$.

φ_{jt} = demand shortage in district j in period t (w.r.t. the lower threshold), $j \in I, t \in TP$.

Noting that $x_{jjt} = 1$ for some $j \in I$ and $t \in TP$, indicates that TU j is assigned to itself in period t, which also means that it is selected as the "representative" TU of its district in that period.

Considering the above parameters and decision variables, an optimization model can be formulated for the multi-period districting problem being studied, which is called (M1):

$$\min \quad \sum_{i \in I}\sum_{j \in I} c_{ij} d_{i1} x_{ij1} + \sum_{t \in TP \setminus \{1\}} \sum_{i \in I}\sum_{j \in I} (r_{ijt} d_{it} v_{ijt} - s_{ijt} d_{it} w_{ijt}) + \sum_{t \in TP}\sum_{j \in I} \big(g_{jt}\psi_{jt} + h_{jt}\varphi_{jt}\big), \tag{1}$$

$$\text{s. t.} \quad \sum_{j \in I} x_{ijt} = 1 \qquad \forall i \in I,\ t \in TP, \tag{2}$$

$$\sum_{j \in I} x_{jjt} = p_t \qquad \forall t \in TP, \tag{3}$$

$$v_{ijt} \ge x_{ijt} - x_{ij,t-1} \qquad \forall i, j \in I,\ t \in TP \setminus \{1\}, \tag{4}$$

$$w_{ijt} \le x_{ij,t-1} \qquad \forall i, j \in I,\ t \in TP \setminus \{1\}, \tag{5}$$

$$w_{ijt} + x_{ijt} \le 1 \qquad \forall i, j \in I,\ t \in TP \setminus \{1\}, \tag{6}$$

$$(1-\alpha)\mu_t x_{jjt} \le \sum_{i \in I} d_{it} x_{ijt} - \psi_{jt} + \varphi_{jt} \le (1+\alpha)\mu_t x_{jjt} \qquad \forall j \in I,\ t \in TP, \tag{7}$$

$$x_{ijt} \in \{0, 1\} \qquad \forall i, j \in I,\ t \in TP, \tag{8}$$

$$v_{ijt} \in \{0, 1\} \qquad \forall i, j \in I,\ t \in TP \setminus \{1\}, \tag{9}$$

$$w_{ijt} \in \{0, 1\} \qquad \forall i, j \in I,\ t \in TP \setminus \{1\}, \tag{10}$$

$$\psi_{jt} \ge 0 \qquad \forall j \in I,\ t \in TP, \tag{11}$$

$$\varphi_{jt} \ge 0 \qquad \forall j \in I,\ t \in TP. \tag{12}$$

In the above model, the objective function (1) represents the total cost for the entire planning horizon, given by the sum of three terms: (i) the total assignment cost incurred for building districts at the beginning of the planning horizon; (ii) the total cost for redesigning districts, i.e., reassigning TUs over the planning horizon minus the corresponding savings (accounted for since period 2); (iii) costs for shortage or surplus. As often done in districting, demands are used as weights when computing the assignment costs (see Kalcsics and Ríos-Mercado, 2019). Constraints (2) ensure that every TU is assigned to exactly one district in every period. Constraints (4)–(6) quantify the reassignments/removals throughout the planning horizon so that the corresponding costs are paid. In particular, Constraint (4) states that there is an actual reassignment whenever a given TU i is assigned in time period t to a district j it was not previously assigned to (i.e., $x_{ijt} = 1$, and $x_{ij,t-1} = 0$). Constraint (5) ensures that the removal of some TU i from district j in time period t cannot be accounted for if that TU was not assigned to that district ($x_{ij,t-1} = 0$). Also, Constraint (6) guarantees that a TU i can be either assigned or reassigned to district j in each time period. Constraints (7) are the balancing constraints; they guarantee that the demand served by each district in every time period is within the maximum prescribed deviation α from the reference value. Note that surplus and shortages are also considered. Finally, constraints (8)–(12) define the domain of the decision variables.

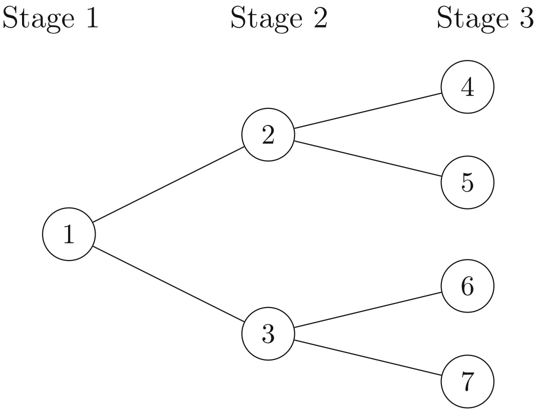

Fig. 1 A multi-stage scenario tree—three stages in the planning horizon

Remark 1 It should be noted that in the above model, contiguity is not explicitly considered. However, optimizing compactness plays in favor of contiguity. In other words, a very compact solution tends to be also contiguous. This evidence is at the base of successful exact solution approaches for districting (Salazar-Aguilar et al., 2011). Clearly, this is not always possible because of the balancing requirements. Nevertheless, many authors do not explicitly address contiguity in their models, especially when dealing with point-like basic TUs (Kalcsics and Ríos-Mercado, 2019). This is also the case, among others, of the work by Diglio et al. (2020), which is set at the core of the developments presented in the current paper.

3 Embedding stochasticity

It is now assumed that the future is uncertain. Uncertainty is most likely associated with demand, but it can also refer to the costs, including the reassignment of TUs and the surplus and shortage costs paid for recovering balancing.

Besides, uncertainty is assumed to be captured by a finite set of previously identified scenarios that induce a scenario tree. Each node in this tree—apart from the root—corresponds to the realization of all the (uncertain) parameters up to that node. The root node represents the initial setting (status quo), directly calling for a here-and-now decision to be made. In the second stage of the scenario tree, the shortage and surplus costs corresponding to period 1 can now be accounted for because the demand has eventually been revealed. Furthermore, planning for a redistricting can be done. Thus, the nodes in stage 2 are represented by child nodes of the root. The tree proceeds by enumerating all the possible moments in which the costs can be accounted for, and the territory design can be changed. In the last stage, the only thing to do is to account for the shortage and surplus costs using the information corresponding to the demand observed.

In Fig. 1, a multi-stage scenario tree with three stages is illustrated. In the second stage the uncertainty associated with that stage is revealed. In Stage 3 the uncertainty associated with that stage is revealed.

Remark 2 It is worth noting that the scenario tree in Fig. 1, along with the described problem setting, can also be interpreted in time-related terms. Stage 1 marks the beginning of the planning horizon. Stage 2 represents the end of period one and the start of period two. Stage 3 marks the end of period 2 and of the planning horizon. In other words, the example above depicts a scenario tree with three stages and two time periods. The demands are realized during a time period, i.e., between two consecutive stages. However, since the problem is not cast within a multi-horizon setting (Escudero and Monge, 2018; Kaut et al., 2014), there is no reason to make a difference between period and stages. Hence, only the term "stage" will be used hereafter.

In the given terminology, one scenario is a full sequence of events from the first stage to the last one. In other words, one scenario fully determines all the information for the entire planning horizon and thus induces a deterministic multi-period districting problem as defined in Sect. 2. In Fig. 1, four possible sequences can be observed, i.e., four scenarios. Each scenario is associated with one and only one leaf node in the scenario tree. For instance, the scenario culminating in node 5 consists of the sequence of events leading from the status quo (stage 1) to the possible "future" observed in stage 2, which is represented by node 2, and, finally, to the possible "future" in stage 3 which is represented by node 5.

Based upon the above representation adopted for the uncertainty, a multi-stage stochastic programming model emerges as a possibility for the problem. To develop such a model, some additional notation is introduced as follows:

$\mathcal{N}$, set of nodes in the scenario tree.

M, number of stages in the decision-making process.

$\mathcal{N}_m$, set of nodes in stage $m \in M$.

Ω, set of scenarios in the scenario tree. Since there is only one path from the root node to every leaf in the tree, a scenario is fully identified by the corresponding leaf node.

$\gamma(n)$, immediate predecessor of node $n \in \mathcal{N} \setminus \{1\}$.

$m(n)$, stage to which node n belongs to, $n \in \mathcal{N}$.

π^n, probability associated with node $n \in \mathcal{N} \setminus \{1\}$. It is unnecessary to consider the probability associated with node 1 since it is equal to 1 (the node corresponds to the current state of nature). Note also that in each stage, some node will occur for sure. Therefore, the probabilities of the nodes are such that, for every stage $m \in M \setminus \{1\}$, i.e., $\sum\limits_{n \in \mathcal{N}_m} \pi^n = 1$.

Example 1 In the case of the scenario tree depicted in Fig. 1, $\mathcal{N} = \{1, \ldots, 7\}$ and $\Omega = \{4, 5, 6, 7\}$. For instance, $\mathcal{N}_2 = \{2, 3\}$, $\gamma(5) = 2$, $m(6) = 3$.

In order to formulate the multi-stage stochastic programming problem, the notation already presented will be adopted, namely, c_{ij}, $i, j \in I$, and α, with the same meaning as before. It is important to note that a node $n \in \mathcal{N}\backslash\Omega$ in the scenario tree represents stage $m(n)$. For that stage, it is known that the number of districts should be $p_{m(n)}$. To simplify the notation, this number is represented simply by p^n, and it is associated with node n since no confusion emerges. In other words, the value p^n $(n \in \mathcal{N}\backslash\Omega)$ is the same for all nodes in the same stage.

To ensure that a general setting is investigated, it is assumed that all the other parameters are stochastic. This calls for them to be redefined as follows:

d_i^n, demand of TU $i \in I$ in stage $m(n)$ if node $n \in \mathcal{N} \setminus \{1\}$ in the scenario tree occurs. Note that the demand of each TU in each stage is unknown beforehand. In the scenario tree, node n belongs to stage $m(n)$, so it represents a possibility for the "future" that may be observed. Thus, for some node $n \in \mathcal{N}\backslash\{1\}$, d_j^n represents one possible observation of the random variable $d_{i,m(n)}$. The μ^n value is defined as $\frac{1}{p_{m(n)-1}} \sum_{i \in I} d_i^n$ $(n \in \mathcal{N} \setminus \{1\})$. This is a value associated with node n that represents the reference value for the demand that should be assigned to each district in stage $m(n)$, i.e., the stage of node n.

g_j^n, unit penalty cost for surplus at district j in node n ($j \in I$, $n \in \mathcal{N} \setminus \{1\}$). Due to uncertainty, the penalty for surplus can be assessed over time. Hence, in the stochastic model, these costs are associated with nodes in stages $2, \ldots, M$.

h_j^n, unit penalty cost for shortage at district j in node n ($j \in I, n \in \mathcal{N} \setminus \{1\}$). As for the surplus costs, in the stochastic model, these costs are associated with nodes in stages $2, \ldots, M$.

r_{ij}^n, unit cost for reassigning TU i to district j in node n ($i, j \in I, n \in \mathcal{N}\backslash(\Omega \cup \{1\})$). Note that reassignments are made neither at stage 1 (node 1) nor at stage M (nodes in Ω).

s_{ij}^n, unit saving for removing TU i from district j in node n ($i, j \in I, n \in \mathcal{N}\backslash(\Omega \cup \{1\})$). As for the reassignment costs, only stages $2, \ldots, M-1$ are considered (from stage 2 to stage $M-1$).

Regarding the decision variables, they are grouped into two sets:

(i) Variables accounting for the shortage and surplus in each stage. Note that such values can only be assessed after the demand in the stage is disclosed.

(ii) Variables corresponding to planning for a stage. These correspond to districting/redistricting decisions.

The decision variables can be formally defined as follows:

ψ_j^n = demand surplus in district j in node n, i.e., a possible value for the surplus occurring in district j in stage $m(n)$ ($j \in I, n \in \mathcal{N} \setminus \{1\}$).

φ_j^n = demand shortage in district j in node n, i.e., a possible value for the shortage occurring in district j in stage $m(n)$ ($j \in I, n \in \mathcal{N} \setminus \{1\}$).

$$x_{ij}^n = \begin{cases} 1, & \text{if TU } i \text{ is assigned to TU } j \text{ in node } n \text{ (stage } m(n)); \\ 0, & \text{otherwise.} \end{cases} \qquad (i, j \in I, n \in \mathcal{N}\backslash\Omega)$$

$$v_{ij}^n = \begin{cases} 1, & \text{if TU } i \text{ is reassigned to TU } j \text{ in node } n \text{ (stage } m(n)); \\ 0, & \text{otherwise.} \end{cases} \qquad (i, j \in I, n \in \mathcal{N}\backslash(\Omega \cup \{1\}))$$

$$w_{ij}^n = \begin{cases} 1, & \text{if TU } i \text{ is removed from district } j \text{ in node } n \text{ stage } m(n)); \\ 0, & \text{otherwise.} \end{cases} \qquad (i, j \in I, n \in \mathcal{N}\backslash(\Omega \cup \{1\}))$$

Using the node-indexed decision variables presented, one can formulate a multi-stage stochastic programming model for the multi-period stochastic districting problem being investigated. The objective function represents the total expected cost (initial districting, plus territory redesign, shortages, and surplus):

$$\begin{aligned}
&\sum_{i \in I} \sum_{j \in I} c_{ij} \left(\sum_{n \in \mathcal{N} | \gamma(n)=1} \pi^n d_i^n \right) x_{ij}^1 \\
&+ \sum_{n \in \mathcal{N}\backslash(\Omega \cup \{1\})} \pi^n \sum_{i \in I} \sum_{j \in I} \left(r_{ij}^n \left(\sum_{\nu \in \mathcal{N} | \gamma(\nu)=n} \frac{\pi^\nu}{\pi^n} d_i^\nu \right) v_{ij}^n - s_{ij}^n \left(\sum_{\nu \in \mathcal{N} | \gamma(\nu)=n} \frac{\pi^\nu}{\pi^n} d_i^\nu \right) w_{ij}^n \right) \\
&+ \sum_{n \in \mathcal{N}\backslash\{1\}} \pi^n \sum_{j \in I} \left(g_j^n \psi_j^n + h_j^n \varphi_j^n \right), \\
=&\sum_{i \in I} \sum_{j \in I} \left[c_{ij} x_{ij}^1 \sum_{n \in \mathcal{N} | \gamma(n)=1} \pi^n d_i^n \right]
\end{aligned}$$

$$+ \sum_{n \in \mathcal{N} \setminus (\Omega \cup \{1\})} \sum_{i \in I} \sum_{j \in I} \left[\left(r_{ij}^n v_{ij}^n - s_{ij}^n w_{ij}^n \right) \sum_{\nu \in \mathcal{N} | \gamma(\nu) = n} \pi^\nu d_i^\nu \right]$$
$$+ \sum_{n \in \mathcal{N} \setminus \{1\}} \pi^n \sum_{j \in I} \left(g_j^n \psi_j^n + h_j^n \varphi_j^n \right),$$

which can be rewritten as:

$$\sum_{i \in I} \sum_{j \in I} c_{ij} x_{ij}^1 \left(\sum_{n \in \mathcal{N} | \gamma(n) = 1} \pi^n d_i^n \right)$$
$$+ \sum_{n \in \mathcal{N} \setminus (\mathcal{N}_1 \cup \mathcal{N}_2)} \sum_{i \in I} \pi^n d_i^n \left(\sum_{j \in I} (r_{ij}^{\gamma(n)} v_{ij}^{\gamma(n)} - s_{ij}^{\gamma(n)} w_{ij}^{\gamma(n)}) \right)$$
$$+ \sum_{n \in \mathcal{N} \setminus \{1\}} \pi^n \sum_{j \in I} \left(g_j^n \psi_j^n + h_j^n \varphi_j^n \right). \tag{13}$$

So, the *Multi-Stage Stochastic Districting Problem (MSSDP)* can be finally formulated as the following model, which is called (M2):

$$\min \quad (13),$$
$$\text{s.t.} \quad \sum_{j \in I} x_{ij}^n = 1 \qquad \forall i \in I,\ n \in \mathcal{N} \setminus \Omega, \tag{14}$$
$$\sum_{j \in I} x_{jj}^n = p^n \qquad \forall n \in \mathcal{N} \setminus \Omega, \tag{15}$$
$$v_{ij}^n \geq x_{ij}^n - x_{ij}^{\gamma(n)} \qquad \forall i, j \in I,\ n \in \mathcal{N} \setminus (\Omega \cup \{1\}), \tag{16}$$
$$w_{ij}^n \leq x_{ij}^{\gamma(n)} \qquad \forall i, j \in I,\ n \in \mathcal{N} \setminus (\Omega \cup \{1\}), \tag{17}$$
$$w_{ij}^n + x_{ij}^n \leq 1 \qquad \forall i, j \in I,\ n \in \mathcal{N} \setminus (\Omega \cup \{1\}), \tag{18}$$
$$(1-\alpha)\mu^n x_{jj}^{\gamma(n)} \leq \sum_{i \in I} d_i^n x_{ij}^{\gamma(n)} - \psi_j^n + \varphi_j^n \leq (1+\alpha)\mu^n x_{jj}^{\gamma(n)} \qquad \forall j \in I,\ n \in \mathcal{N} \setminus \{1\}, \tag{19}$$
$$x_{ij}^n \in \{0, 1\} \qquad \forall i, j \in I,\ n \in \mathcal{N} \setminus \Omega, \tag{20}$$
$$v_{ij}^n \in \{0, 1\} \qquad \forall i, j \in I,\ n \in \mathcal{N} \setminus (\Omega \cup \{1\}), \tag{21}$$
$$w_{ij}^n \in \{0, 1\} \qquad \forall i, j \in I,\ n \in \mathcal{N} \setminus (\Omega \cup \{1\}), \tag{22}$$
$$\psi_j^n \geq 0 \qquad \forall j \in I,\ n \in \mathcal{N} \setminus \{1\}, \tag{23}$$
$$\varphi_j^n \geq 0 \qquad \forall j \in I,\ n \in \mathcal{N} \setminus \{1\}. \tag{24}$$

In the above model, the set of constraints that is not so straightforward is (19). Consider a node $n \in \mathcal{N} \setminus \{1\}$. This means that in stage $m(\gamma(n)) + 1 \equiv m(n)$, the demand observed is part of the "future" leading to node n, i.e., d_i^n, $i \in I$. The shortage and surplus at district j are represented by ψ_j^n and φ_j^n respectively. However, the shortage and surplus depend on the (re-)districting holding in stage $m(\gamma(n)) + 1 \equiv m(n)$, which is represented by variables $x_{ij}^{\gamma(n)}$. On the other hand, the reference value for this stage depends on the observed demand, which explains the use of value μ^n in (19). This explanation is illustrated in Fig. 2.

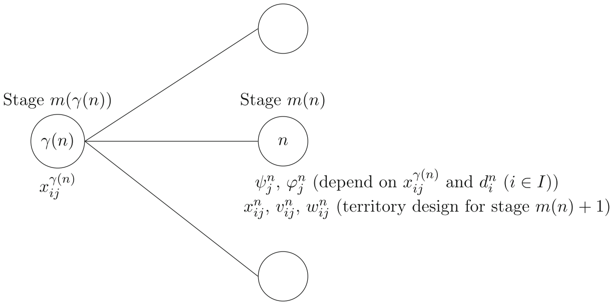
Stage $m(\gamma(n))$
Stage $m(n)$
$\gamma(n)$
n
$x_{ij}^{\gamma(n)}$
ψ_j^n, φ_j^n (depend on $x_{ij}^{\gamma(n)}$ and d_i^n ($i \in I$))
x_{ij}^n, v_{ij}^n, w_{ij}^n (territory design for stage $m(n)+1$)

Fig. 2 Illustration of the building blocks for Constraints (19)

Remark 3 In the objective function, different cost factors are considered: penalty costs and (re-)assignment costs/savings. Note that, due to the non-negativity of parameters g_j^n and h_j^n, surplus and shortages can be calculated as follows:

$$\psi_j^n = \max\{0, \sum_{i \in I} d_i^n x_{ij}^{\gamma(n)} - (1+\alpha)\mu^n x_{jj}^{\gamma(n)}\}, \; j \in I, n \in \mathcal{N} \setminus \{1\}$$

$$\varphi_j^n = \max\{0, (1-\alpha)\mu^n x_{jj}^{\gamma(n)} - \sum_{i \in I} d_i^n x_{ij}^{\gamma(n)}\}, \; j \in I, n \in \mathcal{N} \setminus \{1\}.$$

In practice, they account for the deviation from the upper and lower thresholds in the balancing constraints. These quantities turn out to be crucial for the trade-off between two different courses of action that the model is seeking throughout the planning horizon: Reassignment of TUs and performing extraordinary actions to compensate for shortages/surplus. If the costs associated with the latter are too high compared to those associated with the former, the problem reduces to a multi-period districting-redistricting problem. In the other extreme case, reassignments are forbidden. Hence, the problem has a bi-objective "flavour", although it is not handled explicitly in a multicriteria setting.

3.1 The relevance of considering a multi-stage stochastic model

One important aspect when considering a multi-stage stochastic model (as has just been done) concerns its effective need. In two-stage stochastic programming, this is often assessed by the so-called value of the stochastic solution. This value can be extended to the multi-stage case in different ways (see, e.g., Escudero et al., 2007; Ziegler, 2012). In any case, the underlying idea is always to compute an expected value solution as an approximation to the stochastic problem. In this work, a conditionally expected value approach is adopted, corresponding to a dynamic solution to the average scenario as proposed in the above papers.

The process can be briefly summarized as follows: compute the expected values for all random variables and compute the deterministic solution. Fix the first-stage decision. Compute conditional expected values for all random variables in the subtree for each node in the subsequent stage of the scenario tree and solve the (deterministic) problem thus induced from that node. Use the solutions to fix the decisions for the actual stage. Proceed to the next stage and iterate likewise until the leaf nodes of the scenario tree are reached.

According to this procedure, a model, say EV^n, is solved in every node of the scenario tree. We denote its optimal value by Z_{EV^n}. This is done sequentially for stages $1, 2, \ldots, M-1$. Every model EV^n "aggregates" the costs from stage $m(n)$ to the end of the planning horizon.

Following Escudero et al. (2007), the expected result in stage m of using the above dynamic solution of the average scenario is defined, denoted by EDEV$_m$ ($m = 1, \ldots, M-1$) as the expected value of the optimal values of problems EV^n problems, with $n \in \mathcal{N}$, such that $m(n) = m$. In other words, EDEV$_m$ is defined as:

$$\text{EDEV}_m = \sum_{n \in \mathcal{N}_m} \pi^n Z_{EV^n}, \quad m = 1, \ldots, M-1.$$

Finally, the dynamic value of the stochastic solution is computed as:

$$DVSS = EDEV_{M-1} - SP, \quad \text{or} \quad \%DVSS = \frac{EDEV_{M-1} - SP}{SP} \times 100,$$

where SP is the optimal value of the multi-stage stochastic model.

Although insightful regarding the possibility of simplifying the multi-stage stochastic model, the DVSS does not actually assess the relevance of capturing uncertainty in the problem. To do so, one relies on another well-known indicator: the Expected Value of Perfect Information (EVPI). The EVPI measures how much a decision maker would be willing to pay to get access to perfect information about future demand realization(s). Therefore, the higher the EVPI, the more important the role uncertainty plays in the problem (and, thus, the more relevant it is to capture it). This is a value computed easily: for each scenario $\omega \in \Omega$ one computes the optimal value of the corresponding multi-period single-scenario problem. Denote that value by Z_ω. The value of the wait-and-see solution value is computed as:

$$WS = \sum_{\omega \in \Omega} \pi^\omega Z_\omega,$$

and, finally:

$$EVPI = SP - WS, \quad \text{or} \quad \%EVPI = \frac{SP - WS}{WS} \times 100.$$

4 A heuristic algorithm

The mathematical model proposed for the MSSDP in the previous section easily becomes intractable due to the number of scenarios in the scenario tree that will increase fast with the number of stages. For this reason, in this section, a heuristic algorithm is proposed seeking to find hopefully good feasible solutions to the problem.

The procedure relies on the resolution of the problem by considering a restricted model in which the set of representative TUs that can be selected in each node $n \in N \setminus \Omega$ of the scenario tree is limited to a restricted set of candidates, say C_n. Recall that in the last stage, M, only reassignments, surplus, and shortages based on districting decisions made in stage $M-1$ are accounted for, which explains why no restricted sets are defined in stage M. Thus, the restricted model, which is called (M2R), results from (M2) with the following additional constraints:

$$\sum_{j \in C_n} x_{ij}^n = 1, \quad i \in I,\ n \in \mathcal{N} \setminus \Omega, \tag{25}$$

$$x_{ij}^n = 0, \quad i \in I,\ j \in I \setminus C_n,\ n \in \mathcal{N} \setminus \Omega. \tag{26}$$

Constraints (25) allow the assignment of TU i in node n only to a potential representative, while Constraints (26) forbid assignments to non-potential representatives. Note that it is possible to avoid the introduction of variables $x_{ij}^{n}, i \in I,\ j \in I \backslash C_n,\ n \in \mathcal{N} \backslash \Omega$ although they are kept for a clearer exposition.

The heuristic is formalized in Algorithm 1. Lines 1–8 seek to iteratively define the restricted sets $C_n, n \in N \backslash \Omega$. Initially, no restriction is imposed on the candidate sets C_n (line 1). Thus, all sets coincide with I, and models (M2) and (M2R) coincide (all the TUs are initially regarded as potential representatives).

A loop starts in phase 1 and ends in stage $M-1$ to define all the restricted sets to consider in each stage. In particular, in iteration m, the sets C_n for nodes n in stage m become fixed. This is accomplished as follows. First, the linear relaxation of (M2R) is solved. Note that, in the current iteration m, the restricted sets C_n for all stages before m have been fixed, which is already reflected in (M2R) and thus in its linear relaxation, via Constraints (25) and (26), that should be updated each time new restricted sets are found. Now, the solution of the linear relaxation is examined and the values of the self-assignment variables x_{jj}^{n} for n in stage m ($m(n) = m$) are retrieved. Those greater than zero provide a candidate for being a TU representative. Accordingly, every set C_n in stage m is built from scratch using only such TUs (lines 5–8). When the model (M2R) is solved next time, these restricted sets are updated accordingly.

When the sets C_n are fully determined, the model (M2R) (that now has embedded all the restricted sets) is solved. This is done in line 9 of Algorithm 1. The obtained solution is the approximation proposed for the optimal solution of MSSDP.

Algorithm 1 Heuristic.

```
1:  $C_n \leftarrow I, n \in N \setminus \Omega$;                       // set of candidate representatives for each node
2:  for $m \in \{1, .., M-1\}$ do                                  // for each stage, except the last one
3:      Solve the linear relaxation of the model (M2R). Denote the corresponding solution by $\bar{x}$;
4:      for $n \in N_m$ do                                         // examine the solution
5:          $C_n \leftarrow \emptyset$;
6:          for $j \in I$ do
7:              if $\bar{x}_{jj}^{n} > 0$ then
8:                  $C_n \leftarrow C_n \cup \{j\}$;                 // store TU $j$ as a potential representative for node $n$
9:  Solve the model (M2R);
10: return $x^{\star}$.                                            // a feasible solution for the MSSDP
```

5 Computational experiments

This section reports on the computational experiments performed to validate the proposed model (M2) for the MSSDP and the heuristic proposed for approximating its optimal solution. The test data used is described in Sect. 5.1, while the obtained results are presented in Sects. 5.2–5.3. Specifically, Sect. 5.2 overviews extensive results. Section 5.3 focuses on additional results for larger-sized instances. Finally, the reader is also addressed to the Electronic Supplement—Appendix A, where information about some selected instances is provided to understand the solutions obtained and illustrate the relevance of capturing uncertainty in the investigated problem.

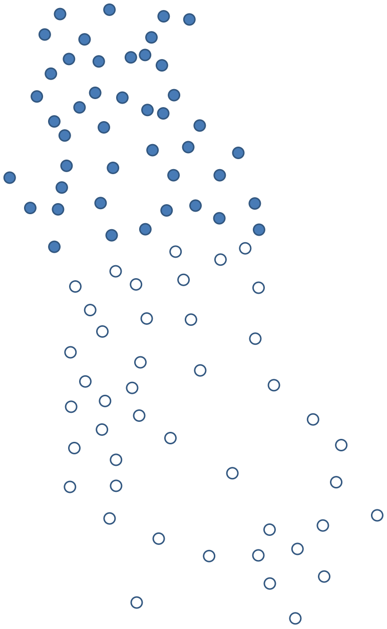

Fig. 3 Test instance - basic TUs corresponding to the centroids of the province of Novara (note: "southern" centroids displayed as empty dots)

5.1 Test data and implementation details

For the computational experiments, the set of instances from Diglio et al. (2020) is used as a starting point. These instances use real geographical data corresponding to the province of Novara, in northern Italy. Specifically, there are 88 point-like basic TUs ($|I| = 88$), which are determined as the centroids of the municipalities in the province (see Fig. 3).

To investigate the relevance of considering a multi-stage decision-making process under uncertainty, the focus is on a small case with three stages, namely, that depicted in Fig. 1. Hence, for illustrative purposes, four scenarios ($|\Omega| = 4$) and seven nodes ($|N| = 7$) are considered. The probabilities of reaching each node $n \in N$ from the immediate predecessor are the same for each node.

The following introduces a setting where the only stochastic parameter is represented by demands. The corresponding data was obtained as follows. The demands in node 1 were generated assuming they were represented by random variables following a Uniform Distribution with a fixed expected value equal to 50, and Relative Standard Deviation (RSD) equal to 0.1[1] To ensure similarity in demands, a choice is made at the start of the planning horizon. It is assumed that the demands in the lower branch of the scenario tree will remain unchanged. This means that for i in I, $d_i^3 = d_i^6 = d_i^7 = d_i^1$.

However, some variability is introduced in the upper branch of the scenario tree, specifically in nodes 2, 4, and 5. For node 2, it is assumed that there is a 50% probability that TUs (Transportation Units) will experience a 25% reduction in demand compared to that in node 1. To achieve this, a random number λ_i^2 is generated following a Bernoulli distribution with a parameter of 0.50. If the generated random number equals one, the demand in node 2 is

[1] For a random variable d, its RSD is given by the ratio between the square root of its variance ($\mathbb{V}ar[d]$) and its expected value ($\mathbb{E}[d]$), namely: $\frac{\sqrt{\mathbb{V}ar[d]}}{\mathbb{E}[d]}$. When considering such values, for a Uniform distribution in the range $[a, b]$, it is possible to calculate a as $\mathbb{E}[d](1 - \sqrt{3} \times RSD)$, and b as $\mathbb{E}[d](1 + \sqrt{3} \times RSD)$.

computed as 75% of the corresponding demand in node 1. Otherwise, the demand in node 2 will remain the same as that in node 1. The same mechanism applies to nodes 4 and 5, with a 50% probability of a 50% and 75% reduction in demand, respectively.

The generation process can replicate situations where demand originating from TUs may undergo unexpected severe reductions or increases in the future. In summary, there is:

$$d_i^n = \begin{cases} (1-\theta^n) \times d_i^{\gamma(n)}, & \text{if } \lambda_i^n = 1, \\ d_i^{\gamma(n)}, & \text{if } \lambda_i^n = 0, \end{cases} \quad i \in I, n \in \{2, 4, 5\},$$

with $\theta^2 = 0.25$, $\theta^4 = 0.50$, and $\theta^5 = 0.75$.

Using the above procedure, ten different instances were generated, referred to as "Instances_1". Additionally, a second set of instances, called "Instances_2", was created, in which the same demand-generation procedure was applied only to the "southern" TUs. These TUs have 50% of the centroids having the lowest y-coordinates. Doing so enforced a "local" trend in demand variation. This approach is expected to make balancing constraints more difficult to meet in the following stages, which may result in higher reassignment/penalty costs. In Fig. 3, the "southern" TUs are displayed as empty dots. All the instances used in the computational tests whose results are reported in this section are available to anyone interested upon request from the authors.

Concerning the other parameters underlying the given instances, they were set as follows:

- Assignment costs: $c_{ij} = \ell_{ij}$, $i, j \in I$, where ℓ_{ij} is the euclidean distance between TUs i and j;
- The probabilities of the nodes $n \in \mathcal{N}\backslash\{1\}$ in the scenario tree are set equal to $\frac{1}{2}$, $\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{4}$, $\frac{1}{4}$, and $\frac{1}{4}$ for nodes 2 to 7, respectively.
- The reassignment costs are defined as $r_{ij}^n = \ell_{ij}$, $i, j \in I$, $n \in \mathcal{N}\backslash\{\Omega \cup \{1\}\}$;
- The savings from removing a TU from a district are determined as $s_{ij}^n = \zeta \cdot l_{ij}$, $i, j \in I$, $n \in \mathcal{N}\backslash(\Omega \cup \{1\})$, with $\zeta \in [0, 1]$. Hence, the savings are defined as a fraction of the assignment costs. $\zeta = 0$ is set to zero, which means no savings are considered.
- Finally, for the penalty costs, the maximum distance between any pair of TUs is considered as $g_j^n = h_j^n = \max_{i,j \in I}\{\ell_{ij}\}$, $j \in I$, $n \in \mathcal{N}\backslash\{1\}$. In other words, it is set as the maximum distance between any pair of TUs.

Remark 4 The above-discussed settings lead to an interesting interpretation of the unit penalty costs. To explain it, let us focus on a single TU i. It is supposed that by reassigning this TU in node n to a new district k, a surplus in the district it currently belongs to, say j, can be avoided. The corresponding penalty and reassignment costs can be computed, respectively, as $g_j \times d_i^n$, and $\ell_{ik} \times d_i^n$. Therefore, a reassignment will be performed only if $\ell_{ik} < g_j$. Similar reasoning can be followed for shortages. In practice, penalty costs can be interpreted as the maximum distance within which reassignments are worth accepting. In this specific case, a TU is reassigned as long as its distance from the new representative is less than the maximum distance among the TUs. Otherwise, penalties are preferred. This observation is in line with the already mentioned bi-objective "flavour" of the problem (see Sect. 3).

Once the above parameters are fixed, various experiments are realized by varying the value of the tolerance $\alpha \in \{0.05, 0.10, 0.15, 0.20, 0.25\}$, and the number of districts $p \in \{4, 6\}$. In total, 200 experiments are performed: five values of α, two values of p, and ten different demand generations for both *Instances_1* and *Instances_2*.

All the experiments were performed on an Intel(R) Core(TM) i7-7700 with 3.60 GHz and 64 GB of RAM, running the Windows 10 Pro 64-bit operating system. The models and the heuristic were coded in Python 3.7, using IBM ILOG CPLEX 12.10 as a solver. A time limit of three hours was set for the solution of the *MSSDP*. The procedures for calculating the Dynamic Value of the Stochastic Solution (DVSS) and the Expected Value of Perfect Information (EVPI) were only run when the corresponding MSSDP was solved up to proven optimality within the prescribed time limit. For both procedures, an overall time limit of three hours—10800 s—was considered.

5.2 Extensive results

In the following, the results obtained by solving model (M2) for all the generated instances described in Sect. 5.1 will be discussed. The results are summarized in Table 1. For every instance, the presented information includes:

- The computational performance of the model, by showing: (i) the number of optimal solutions—out of 10—obtained within the imposed time limit (10,800 s); (ii) the minimum, maximum, and average optimality gap (in %) of the obtained solutions (clearly, the gap equals zero when solutions are optimal); (iii) the minimum, maximum, and average computing times (seconds). The table displays "t.l." under the maximum computing time column if at least one instance was not optimally solved within the time limit. Of course, if no optimal solutions were obtained, "t.l." would also occur for the minimum and average cases.
- The relevance of capturing uncertainty in the problem by means of the %EVPI and the quality of the expected value solution for approximating the optimal solution to the stochastic problem through the %DVSS. These indicators were only calculated when the corresponding model's solutions were optimal. Accordingly, "N/A" is displayed if none of the ten instances was solved to proven optimality.

Starting with a focus on *Instances_1*, it can be observed that the model could optimally solve all the tested instances for $p = 4$ within acceptable computing times, averaging 117 s. In fact, the number of optimal solutions was the same for all considered values of the tolerance α, equaling 10. It is also worth highlighting that computing times tend to increase as α decreases. This result is not unexpected, as lower values of α tighten the balancing constraints, thus making them harder to meet. This emphasizes the trade-off between the (re-)assignments and penalty costs when seeking the optimal solution to the problem. Interestingly, such a finding is reflected by the distributions of the %DVSS and %EVPI. In particular, as α increases, the %DVSS reduces to about 8% for $\alpha = 0.20$ and $\alpha = 0.25$. This indicates that as instances become relatively easier to solve, deterministic expected-value problems can produce better approximate solutions to the MSSDP. Nevertheless, the above values are not negligible. Furthermore, note that the %DVSS equals 15.50% on average, with a peak of more than 30% for $\alpha = 0.10$. Therefore, it can be argued that explicitly considering uncertainty in the model is crucial. The relevance of hedging against uncertainty is also supported by the obtained values for the %EVPI.

A similar behaviour is observed for $p = 6$. However, these instances proved harder to solve, especially for $\alpha = 0.05$. Indeed, under such settings, the model could not obtain the optimal solutions for two instances, exhibiting an optimality gap equal to 1.71% in the worst case. Recall that an increase in the value of p reduces the lower and upper threshold used in the balancing constraints. Clearly, this effect is amplified if jointly occurring with a decrease of α. Finally, no significant differences are devised in terms of the %DVSS and %EVPI.

The computational performance of the model drastically reduces when tackling *Instances_2*. In particular, optimal solutions were not obtained for $p = 6$ and $\alpha = 0.05$. Remarkably, the corresponding average optimality gap is about 7% in this case. Also, the running times increase significantly, being higher than 3000 and 5000 s for $p = 4$ and $p = 6$, respectively. These results highlight that resorting to heuristic procedures is necessary to reduce the computational effort required to obtain feasible (and, hopefully, higher-quality) solutions to the problem, even for small-sized instances, as shown in Sect. 5.2.2.

On a last note, it is observed that the uncertainty considerations turn out to be more relevant for these instances, as the higher values of %DVSS and %EVPI reveal, which suggests that capturing uncertainty becomes more relevant when it only affects a subset of locally-distributed TUs. For a further discussion on the latter, the reader is also referred to the Electronic Supplement—Appendix A. Moreover, a deeper look into these indicators is given in Sect. 5.2.1. Finally, for more detailed results about the performed experiments, the interested reader is also addressed to the Electronic Supplement—Appendix B, Tables B1—B4.

Next, the analysis proceeds in two different directions. Section 5.2.1 provides some additional insights and results from the solution of the model, while Sect. 5.2.2 overviews the performance of the proposed heuristic.

5.2.1 Additional insights

This section explores three further aspects of relevance to our problem: (i) a cost breakdown analysis, to assess the motivation for the values of the %DVSS and %EVPI discussed above; (ii) the effect of considering both decreasing and increasing demand scenarios; (iii) the effect of savings.

(i) Cost breakdown analysis

Hereafter, a cost breakdown analysis of the obtained solutions is presented to explain the values of the %DVSS and %EVPI that emerged from the performed experiments. To this end, a subset of solutions is considered—specifically, those obtained for *Instances_1*, $p = 4$, and $\alpha = 0.10, 0.15, 0.20$ (i.e., rows 2, 3, and 4, respectively, in the upper part of Table 1).

For these 30 solutions, Table 2 reports: (i) the objective function (OF) and its three components, i.e., (ii) the initial assignment costs (AssCosts), (iii) the reassignment costs (ReassCosts), and (iv) the penalty costs (PenCosts) for the MSSDP, its expected value approximation, i.e., the so-called $EDEV_{M-1}$, and the wait-and-see (WS) solution. Note that, for brevity, the above indicators are expressed as the average of the ten solutions obtained for each value of α.

The comparison between the MSSDP and EDEV$_{M-1}$ explains the high observed %DVSS. As the table shows, the objective function of the EDEV$_{M-1}$ is higher than the MSSDP regardless of the value of α. The reason for this outcome is clear. Indeed, approximating the stochastic program by its expected value leads to a more compact first-stage solution (i.e., the Expected Value solution rooted in node 1, i.e., the so-called EV^1). The latter, in fact, has lower initial assignment costs than the first-stage solution of the MSSDP (see the "AssCosts"-column). However, this turns out to be very weak when hedging against uncertainty, leading to more expensive reassignments and penalties for unmet balancing.

The comparison between the WS and MSSDP reveals that, regardless of α, accessing perfect information about uncertainty can foster the identification of more compact first-stage solutions (the initial assignment costs —"AssCosts"—are lower than the WS). Also, it helps hedge against uncertainty more effectively, as the lower reassignment costs and penalties reveal. However, the differences are somewhat limited when one looks at the values involved, which clarifies the low observed %EVPI values.

ii) Varying demand scenarios

Table 1 Summary of the results obtained by solving model (M2)

Instance	p	α	*#Optimal*	Optimality Gap (%)			Computing time (s)			%DVSS			%EVPI		
			(out of 10)	Min	*Max*	*Avg*	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg
Instances_1	4	0.05	10	0	0	0	26	3101	390	2.61	28.01	11.25	0.4	2.5	1.39
		0.1	10	0	0	0	19	165	59	6.77	49.65	30.45	0.1	1.76	0.86
		0.15	10	0	0	0	23	386	101	11.35	38.57	19.57	0.39	2.06	1.15
		0.2	10	0	0	0	13	29	19	0.96	28.33	8.25	0.26	1.53	0.95
		0.25	10	0	0	0	12	21	14	0.1	23.46	7.96	0.29	1.39	0.61
		Total	50	0	0	0	12	3101	117	0.1	49.65	15.50	0.1	2.5	0.99
	6	0.05	8	0	1.71	0.22	163	t.l	4738	6	37.12	21.55	1.45	4.4	2.73
		0.1	10	0	0	0	13	296	88	0.42	32.14	15.65	0.1	2.17	1.14
		0.15	10	0	0	0	14	70	29	5.61	43.71	17.25	0.32	1.59	0.86
		0.2	10	0	0	0	11	27	17	2.05	22.08	10.69	0.12	1.45	0.59
		0.25	10	0	0	0	10	21	14	0	16.3	7.72	0	1.24	0.38
		Total	48	0	1.71	0.04	10	t.l	977	0	43.71	14.57	0	4.4	1.14
Instances_2	4	0.05	2	0	1.95	0.76	1307	t.l	9775	44.58	50.23	47.4	5.43	7.05	6.24
		0.1	8	0	0.88	0.13	247	t.l	3928	43.56	55.11	48.58	2.96	8.52	6.42
		0.15	10	0	0	0	46	8362	1525	42.44	65.79	55.69	2.62	8.79	5.77
		0.2	10	0	0	0	37	1594	468	41.48	59.19	50.5	3.13	8.2	5.72
		0.25	10	0	0	0	71	588	297	24.34	60.68	43.91	2.51	7.26	5.09
		Total	40	0	1.95	0.18	37	t.l	3199	24.34	65.79	49.22	2.51	8.79	5.85
	6	0.05	0	2.7	11.56	7.13	t.l	t.l	t.l	N/A	N/A	N/A	N/A	N/A	N/A
		0.1	1	0	2.5	1.28	8417	t.l	10,568	32.27	32.27	32.27	2.7	2.7	2.7
		0.15	10	0	0	0	140	10,385	3640	42.21	52.57	48.35	1.86	6.53	4.15
		0.2	10	0	0	0	30	453	127	33.63	54.15	47.33	1.2	4.92	2.57
		0.25	10	0	0	0	17	238	48	18.15	56.01	37.73	1.07	3.19	1.73
		Total	31	0	11.56	1.68	17	t.l	5037	18.15	56.01	41.42	1.07	6.53	2.79

Table 2 Cost breakdown analysis for a subset of obtained optimal solutions (88-TUs, $p = 4$, *Instances_1*

α		OF	AssCosts	ReassCosts	PenCosts
0.10	MSSDP	2.93E+07	2.92E+07	1.10E+05	4.60E+04
	EDEV$_{M-1}$	3.83E+07	2.90E+07	8.29E+06	9.15E+05
	WS	2.91E+07	2.91E+07	0.00E+00	4.66E+03
0.15	MSSDP	2.88E+07	2.86E+07	8.51E+04	5.39E+04
	EDEV$_{M-1}$	3.44E+07	2.84E+07	4.87E+06	1.12E+06
	WS	2.85E+07	2.85E+07	3.93E+03	3.14E+03
0.20	MSSDP	2.82E+07	2.80E+07	1.64E+05	2.40E+04
	EDEV$_{M-1}$	3.05E+07	2.79E+07	1.78E+06	8.42E+05
	WS	2.79E+07	2.79E+07	5.45E+03	1.27E+04

Table 3 Results for the 88-TUs instances without (w/o) and with (w) increasing demand scenarios

α	%DVSS		%EVPI		CPU times	
	w/o	w	w/o	w	w/o	w
0.10	30.45	31.85	0.87	1.05	59	154
0.15	19.57	33.18	1.15	1.13	101	82
0.20	8.25	18.03	0.95	1.28	19	95

So far, instances with decreasing demand scenarios have been considered. In this section, additional tests are performed where changes in the demand can occur also in the "lower branch" of the tree, i.e., in nodes 3, 6, and 7. Specifically, it is assumed that demands can increase following the procedure for their generation described in 5.1 (and using the same values of θ therein defined). In practice, demands in node 3 can increase by 25% compared to node 1, while in nodes 6 and 7 they can increase by 50% and 75% w.r.t. node 3, respectively. These increases have a probability (λ_i) of 0.50 to occur. Mathematically, the following holds:

$$d_i^n = \begin{cases} (1-\theta^n) \times d_i^{\gamma(n)}, & n \in \{2,4,5\}, \text{ if } \lambda_i^n = 1, \\ (1+\theta^n) \times d_i^{\gamma(n)}, & n \in \{3,6,7\}, \text{ if } \lambda_i^n = 1, \\ d_i^{\gamma(n)}, & n \in \mathcal{N}\backslash\{1\}, \text{ if } \lambda_i^n = 0, \end{cases}$$

with $\theta^2 = \theta^3 = 0.25, \theta^4 = \theta^6 = 0.50$, and $\theta^5 = \theta^7 = 0.75$.

For these tests, the 88-TUs of the type of *Instances_1* were considered and solved for $p = 4$ and $\alpha \in \{0.10, 0.15, 0.20\}$, thus resulting in 30 new experiments (10 demand generation for each combination of p and α). Table 3 summarizes the main obtained findings, by showing, for each value of α the average (i) %DVSS, (ii) % EVPI, and (iii) CPU Time (in sec.). Specifically, results are given for both cases, i.e., without and with increasing demands (denoted by "w/o" and "w", respectively). Note that results for the "with"-cases are the same as in Table 1.

The table reveals several key findings. Firstly, the values of %DVSS are consistently higher for the instances including increasing demand scenarios. This indicates that deterministic approximations are less accurate when considering mixed-demand scenarios. Secondly, slightly higher values of the %EVPI are observed, meaning that accessing perfect information has higher relevance in such a setting. Thirdly, the instances appear to be relatively more challenging to solve—except for $\alpha = 0.15$—as the corresponding running times underscore.

Table 4 Assessing the impact of savings (88-TUs instances, $p = 4$, $\alpha = 0.25$, *Instances_2*)

ζ	OF	AssCosts	ReassCosts	PenCosts	CPU time
0	3.16E+07	3.00E+07	1.57E+06	3.01E+04	588
0.2	3.15E+07	3.00E+07	1.45E+06	3.01E+04	3105
0.4	3.14E+07	3.00E+07	1.34E+06	3.01E+04	4677
0.6	3.13E+07	3.00E+07	1.22E+06	3.01E+04	21,941
0.8	3.05E+07	2.88E+07	1.74E+06	0.00E+00	17,289
1	2.81E+07	9.83E+07	-7.02E+07	0.00E+00	5272

Overall, these initial findings suggest that this line of investigation holds promise. However, further and more extensive analysis is needed to confirm their generalizability.

(iii) The role of savings

This section aims to show how savings can impact the solutions produced by the proposed model. To this end, some further computational tests were performed. As an illustrative example, a specific instance was considered (out of the ten generated for *Instances_2*) and tested for $p = 4$, $\alpha = 0.25$ and by varying the savings s^n_{ij}. Recall that savings obtained by removing a TU from a district are defined as follows: $s^n_{ij} = \zeta \cdot l_{ij}, i, j \in I, n \in \mathcal{N} \backslash (\Omega \cup \{1\})$, with $\zeta \in [0, 1]$. Hence, the savings are a fraction of the assignment costs. For these new experiments, ζ was varied from 0 (i.e., no savings are considered) to 1 (i.e., the saving equals the initial assignment cost) with a pace equal to 0.2. The obtained results are summarized in Table 4, which reports, for each tested value of ζ: (i) the objective function (OF) and its three components, i.e., (ii) the initial assignment costs (AssCosts), (iii) the reassignment costs (ReassCosts, net of the savings), (iv) the penalty costs (PenCosts), and (v) the CPU Times needed to solve these instances up to optimality (in seconds).

As the Table shows, the model produces the same optimal solutions for ζ values up to 0.6. Indeed, the initial assignment costs and the penalty costs are the same, while the reassignment costs are obviously different because of the different savings being considered.

When looking at the solution obtained for $\zeta = 0.8$, it can be observed that the first stage solution has lower initial assignment costs, but higher reassignment costs. However, such reassignments allow the model to identify a fully balanced solution, i.e., with penalty costs equal to 0. This fact underscores the impact of savings. Indeed, if they are in play, the model can seek more compact first-stage solutions and has an incentive for numerous (cheaper) reassignments in the second stage. In turn, these reassignments can help avoid penalty costs.

For lower values of ζ, instead, it can noticed penalty costs are paid. Indeed, to avoid them and meet balancing, numerous (highly expensive) reassignments would be needed. This explains why, in order to reduce reassignments and penalties, a less compact first-stage is observed. It is worth noticing that these findings further underline the inherent multi-objective flavor of our model.

Finally, for $\zeta = 1$, the initial assignment costs are high, suggesting a poorly compact first-stage solution. However, the value of reassignment costs is negative, which indicates that savings surpass reassignment costs. This means that numerous and cheaper reassignments are made to evolve the initial solution toward a more compact and fully balanced districting plan (penalty costs are again null).

On a final note, it is also observed that the impact of savings is not negligible in terms of computational times, which are at least five times higher w.r.t. to the base-case (i.e., for $\zeta = 0$).

5.2.2 Results for the proposed heuristic

This step of the analysis consists of assessing the performance of the heuristic introduced in Sect. 4. To this end, Algorithm 1 was applied to the whole set of generated testbed instances, and its results were benchmarked against those obtained from the implementation of the model (M2).

A brief comparative assessment is reported in Table 5, which displays, for each Instance: (i) the minimum, maximum, and average gap between the objective functions (Δ_{OF}); (ii) the number of optimal solutions yielded by the model ("Model") and the heuristic ("Heur"); (iii) the minimum, maximum, and average computing times required by the model and the heuristic. Note that the above-mentioned gap is computed as: $\Delta_{OF} = 100 \times (Z_{Heur} - Z_{Model})(Z_{Model})$, with Z_{Model} and Z_{Heur} denoting the objective function values for the solutions obtained by the model and heuristic, respectively. Thus, a negative deviation indicates cases in which the heuristic solution outperforms the best feasible solution found by the model. Clearly, such a circumstance may occur only when the model does not achieve an optimal solution within the imposed time limit.

As the table shows, the heuristic produces very good solutions to the problem. This is particularly evident when looking at *Instances_1*. For $p = 4$, the heuristic attains the optimal solution in 42 out of 50 cases. In the remaining ones, the corresponding gaps are minimal, being equal—at most—to 0.83% (for $\alpha = 0.05$) and 0.03% on average. The number of optimal solutions achieved by the heuristic increases to 46 for $p = 6$. Again, the few non-zero gaps between the objective functions are very limited and equal to 0.36% in the worst case, i.e., for $\alpha = 0.10$. Also, these results highlight something that is wished to be seen. Specifically, if the focus is on $\alpha = 0.05$, it can be found that the minimum and maximum gaps equal -0.12% and 0%, respectively. Therefore, it can be concluded that the heuristic either attains the optimal solution to the problem or outperforms the model when its resolution reaches the time limit prescribed by producing higher-quality solutions. These (near-)optimal solutions are obtained at a very acceptable computational effort. It should be observed that while computing times are comparable for higher values of α, savings become significant as α decreases. On average, it is noticed that the heuristic runs for 28 and 61 s for $p = 4$ and $p = 6$, respectively. In particular, in the latter case, it is worth underlining that the algorithm produces (almost) the same number of optimal solutions by lowering the computing times to about one-fifteenth (from 977 to 61 s).

These findings are confirmed when focusing on the more challenging *Instances_2*. The number of optimal solutions reached by the heuristic reduces (19 out of 31). However, the gaps are relatively small and equal to 0.85% in the worst case ($p = 6, \alpha = 0.15$). Interestingly, it must be highlighted that the heuristic can produce significantly better solutions whenever the model fails to achieve the optimal solutions to the problem within the imposed time limit. In fact, the average gap equals -1.66% for $p = 6, \alpha = 0.05$, with an improvement of 3.55% in the best case. Although increased w.r.t. to the first set of instances, running times remain acceptable and, above all, they are one order of magnitude lower than the corresponding solution times of the model (on average, 111 vs. 3199 s for $p = 4$; 478 vs. 5307 s for $p = 6$). More detailed results about are given into the Electronic Supplement—Appendix B, Tables B1—B4.

Overall, these findings validate the proposed heuristic and classify it as effective for the investigated problem.

Table 5 Assessing the performance of the proposed heuristic

Instance	p	α	Δ_{OF} (%)			# Optimal (out of 10)		Computing time (s)					
								Model			Heur		
			Min	Max	Avg	Model	Heur	Min	Max	Avg	Min	Max	Avg
Instances_1	4	0.05	0	0.83	0.08	10	9	26	3101	390	26	40	30
		0.1	0	0.21	0.03	10	8	19	165	59	27	28	28
		0.15	0	0.22	0.06	10	5	23	386	101	27	31	28
		0.2	0	0	0	10	10	13	29	19	26	27	27
		0.25	0	0	0	10	10	12	21	14	25	27	26
		Total	0	0.83	0.03	50	42	12	3101	117	25	40	28
	6	0.05	−0.12	0	−0.02	8	8	162	t.l	4738	29	754	192
		0.1	0	0.36	0.04	10	9	13	296	88	26	39	30
		0.15	0	0	0	10	10	14	70	29	26	42	30
		0.2	0	0.02	0	10	9	11	27	16	26	30	27
		0.25	0	0	0	10	10	10	21	14	25	28	26
		Total	−0.12	0.36	0	48	46	10	t.l	977	25	754	61
Instances_2	4	0.05	−0.1	0.6	0.26	2	2	1307	t.l	9775	56	1388	396
		0.1	−0.14	0.59	0.14	8	4	246	t.l	3928	31	147	64
		0.15	0	0.36	0.07	10	8	45	8362	1525	29	58	36
		0.2	0	0.31	0.05	10	7	37	1594	468	27	35	31
		0.25	0	0.67	0.19	10	5	71	588	297	27	44	30
		Total	−0.14	0.67	0.14	40	26	37	t.l	3199	27	1388	111
	6	0.05	−3.55	0.1	−1.66	0	0	t.l	t.l	t.l	1019	3502	1588
		0.1	−0.26	0.32	−0.03	1	1	8417	t.l	10,567	67	1693	657
		0.15	0	0.85	0.14	10	5	140	10,385	3640	36	161	84
		0.2	0	0.31	0.03	10	8	29	453	126	29	37	32
		0.25	0	0.18	0.06	10	5	17	238	48	27	32	29
		Total	−3.55	0.85	−0.29	31	19	17	t.l	5037	27	3502	478

5.3 Further results: larger-sized instances

The last step of the empirical analysis consists of additional computational tests to assess the capability of the model and the heuristic to solve larger-sized instances for the investigated problem in terms of (i) an increased number of TUs and (ii) a higher number of stages (and scenarios). The related discussions follow in Sects. 5.3.1 and 5.3.2, respectively.

5.3.1 The impact of the number of TUs

For these computations, the 120-TUs instances (i.e., $|I| = 120$) used by Diglio et al. (2020) were considered, and the same experimental setting as above was replicated. Thus, 200 additional experiments were performed, summarized in Table 6. For the sake of brevity, only the average values of some relevant indicators are reported in the latter, while a more extensive overview is provided into the Electronic Supplement—Appendix B, Tables B5–B8.

Three main elements seem to emerge from these extended experiments. First, capturing uncertainty in the problem remains crucial. In particular, the (average) values of %DVSS are often higher than those obtained for the 88-TUs instances (see Table 1), thus revealing that the size of the problem seems to affect this aspect. Second, and as expected, the latter has a clear effect on the computational performance of the model. Indeed, although most of the tested instances are optimally solved within the imposed time limit, the average running times are significantly increased w.r.t. to the former tests. Finally, the heuristic confirms its effectiveness, being capable of attaining (near-)optimal or even improved solutions at an acceptable computational effort.

5.3.2 The impact of the number of stages

For these experiments, the 88-TUs instances were considered. To this end, demands were generated by using the method described in Sect. 5.1. Recall that the following values of θ were assumed for three stages: 0.25 for node 2, 0.5 for node 4, and 0.75 for node 5. Recall also that, for a given node n, θ^n expresses the percentage by which the demand for a generic TU i may reduce w.r.t. to its predecessor.

The scenario tree shown in Fig. 1 was used as a reference and extended to include four and five stages, resulting in eight and 16 scenarios, respectively. In the case of four stages, nodes 8 and 9 would be successors of node 4, with associated values of $\theta = 0.5$ and $\theta = 0.75$. The same applied to nodes 10 and 11, successors of node 5. No variations in the demand occurred for nodes 3, 6, 7, and their successors. The same reasoning was assumed for a five-stage scenario tree. It is worth underlining that in this generation process, no differentiation was made between the "Northern" and "Southern" TUs, which means that demand changes could apply to all the TUs (as for *Instances_1*). Finally, the equiprobability of the scenarios was still assumed, i.e., with each scenario having a probability equal to $\frac{1}{|\Omega|}$ to occurr.

The initial focus is on four stages. Various tests were performed by setting $p = 4$ and varying $\alpha \in \{0.10, 0.15, 0.20\}$. Again, for each combination of p and α, ten instances were generated by varying the demands. This resulted in 30 additional experiments, whose results are summarized in Table 7. The structure of the table and the information therein reported are the same as in Table 6.

It is important to underline that no time limit was imposed for these runs. Hence, all the tested instances were solved up to proven optimality, and the corresponding %DVSS and %EVPI were calculated. Interestingly, it can be observed that both indicators are higher

Table 6 Results for the 120-TUs instances

Instance	p	α	Avg %DVSS	Avg %EVPI	# Optimal (out of 10)		Avg Δ_{OF} (%)	Avg computing times (s)	
					Model	Heur		Model	Heur
Instances_1	4	0.05	42.43	1.17	10	9	0.02	738	91
		0.1	12.1	1.26	10	9	0.01	1030	59
		0.15	33.9	1.23	10	9	0.01	224	58
		0.2	23.05	1.24	10	10	0	325	57
		0.25	27.04	1	10	10	0	125	54
		Total	27.7	1.18	50	47	0.01	488	64
	6	0.05	30.23	2.14	8	7	0.03	5531	228
		0.1	15.61	1.11	10	7	0.02	373	58
		0.15	19.04	0.54	10	10	0	63	55
		0.2	17.2	0.34	10	10	0	39	54
		0.25	14.72	0.3	10	10	0	29	53
		Total	19.36	0.88	48	44	0.01	1207	89
Instances_2	4	0.05	N/A	N/A	0	0	−0.33	$t.l$	4011
		0.1	67.62	4.33	6	3	0.07	6468	669
		0.15	58.4	3.96	10	1	0.17	1232	114
		0.2	56.07	4.12	10	8	0.05	2035	186
		0.25	49.77	3.63	10	9	0	1247	77
		Total	57.96	4.01	36	21	−0.01	4358	1012
	6	0.05	N/A	N/A	0	0	−0.56	$t.l$	2748
		0.1	26.01	3.25	1	1	−0.07	9747	355
		0.15	41.67	2.38	10	9	0.01	1856	66
		0.2	35.65	1.33	10	9	0	88	56
		0.25	44.4	0.72	10	10	0	42	54
		Total	36.93	1.92	31	29	−0.12	4509	656

Table 7 Results for the 88-TUs instances with four stages

α	Avg %DVSS	Avg %EVPI	# Optimal (out of 10)		Avg Δ_{OF} (%)	Avg Computing times (s)	
			Model	Heur		Model	Heur
0.10	34.34	2.01	10	9	0.02	13,735	231
0.15	27.15	2.06	10	9	0.00	10,690	167
0.20	18.97	1.93	10	8	0.03	8484	136
Total	26.82	2.00	30	27	0.02	10,970	178

Table 8 Results for the 88-TUs instances with five stages

# Optimal (out of 10)		Avg Δ_{OF} (%)	Avg OPTgap (%)	Avg GAP_{HLB} (%)	Avg computing times (s)	
Model	Heur				Model	Heur
2	0	−4.76	5.58	0.89	10,185	2803

(even significantly) when compared to those reported in Table 1. This indicates that capturing uncertainty becomes more and more relevant as the number of stages (and scenarios) increases in the investigated problem.

Moreover, it can noticed that the heuristic still classifies as effective, being able to achieve 27 optimal solutions out of 30, with negligible optimality gaps in the remanding cases (on average equal to 0.02%—see the "Avg Δ_{OF}"-column). Finally, the computing times are still very reasonable, as the algorithm converges, on average, in 178 s (231 in the worst case, for $\alpha = 0.10$), that is, about 60 times faster than the solver. Further details on the performed experiments can be found in the Electronic Supplement—Appendix B, Table B9.

Ten additional experiments were performed for five stages, corresponding to ten different demand generations for the 88-TUs instances, with $p = 4$ and $\alpha = 0.20$. In this case, a time limit of 3 h was again imposed. The detailed results for these runs are reported in the Electronic Supplement—Appendix B, Table B10, while some summary statistics are reported in Table 8. In the latter table, the following indicators are also displayed in addition to already presented information for previous tests:

- The average optimality gap—across the ten experiments—of the solutions obtained by CPLEX (Avg. OPTgap);
- The average relative gap—across the ten experiments—between the objective function of the solutions yielded by the heuristic and the lower bound provided by CPLEX (Avg. GAP_{HLB}). It is measured as $GAP_{HLB} = \frac{Z_H - Z_{LB}}{Z_{LB}}$, with Z_H and $Z_L B$ denoting the objective functions of the heuristic solution and the lower bound, respectively.

As the table shows, the solver attains only two optimal solutions within the given time limit, with an average running time of more than 10,000 s. The optimality gap reported by CPLEX was not negligible for the remaining cases, averaging 5.58% (as shown the "Avg OPTgap"-column) and reaching 22.87% in the worst case (see 8—specifically, experiment 4). In contrast, the heuristic runs approximately four times faster, converging in just over 2800 s. More importantly, it outperforms the solver significantly in terms of solution quality. The proposed algorithm improves the best integer solution returned by the solver by an average of about 4.76% (as shown in the "Avg Δ_{OF}"-column). Remarkably, the relative gap of the heuristic from the lower bound provided by CPLEX is very limited and equal to 0.89% on average (see the "Avg GAP_{HLB}"-column).

The results once again confirm the effectiveness of the heuristic, demonstrating that it can represent a valuable tool for solving medium-sized instances involving up to five stages and 16 scenarios. However, results for six stages, which are not presented for brevity, show that the running times of the heuristic increase significantly. This indicates the need for more refined solution methods to handle instances involving higher-cardinality scenario sets. Nevertheless, and on a positive note, it is important to underline that appropriate scenario-reduction techniques can be applied to make the latter more manageable for the proposed algorithm.

6 Conclusions

In this paper, a multi-period stochastic districting problem was investigated. A finite set of scenarios inducing a scenario tree was assumed to capture uncertainty. In the proposed multi-stage stochastic programming model, an initial districting plan was designed at the root node of the scenario tree (the beginning of the planning horizon). Due to the time-dependent and uncertain demands, districts could be rearranged, i.e., some Territorial Units (TUs) could be reassigned at each stage to meet the balancing requirements. Extraordinary actions compensating shortages/surplus in each district at each node were also considered. The proposed model seeks to optimize an objective function given by the sum of three terms: the assignment costs for the initial districting, the costs occurring for redistricting (i.e., TUs' reassignments), and the penalty costs for unmet balancing constraints. A heuristic algorithm to find approximate solutions to the problem was also devised. The introduced procedure solves a restricted model considering a subset of candidate representatives of the districts, found by exploiting information obtained from the solution of the linear relaxation of the problem.

Computational tests performed using test-bed instances from the literature, built upon real geographical data, show the model's capability to solve real-world-like case studies and, above all, the relevance of capturing uncertainty in the investigated problem. To this end, appropriate measures such as the Dynamic Value of the Stochastic Solution (DVSS) and the Expected Value of Perfect Information (EVPI) were computed. Besides, the proposed heuristic proved effective, producing near-optimal solutions with reduced computational effort.

The work done has, of course, some limitations which open various future research directions. First of all, it should be acknowledged that artificially generated data was used for the experiments. While this reveals the relevance of the modelling framework w.r.t. a generic demand distribution, it also calls for future applications to a real-world case study (and, hence, an actual demand distribution). Secondly, the proposed heuristic represents a first attempt to solve the MSSDP. Therefore, another line of research worth pursuing leads to the definition of more sophisticated algorithms for solving larger-sized instances involving a higher number of TUs, more complex scenario trees, and the inclusion of explicit contiguity requirements. Finally, it should be noted that the proposed model's mentioned bi-objective flavour calls for (explicit) multi-objective extensions of MSSDPs.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s10479-024-06459-7.

Acknowledgements The authors are thankful to two anonymous reviewers, whose comments helped to improve the quality of our paper.

Funding Open Access funding enabled and organized by Projekt DEAL.

References

Aldasoro, U., Escudero, L., Merino, M., & Pérez, G. (2013). An algorithmic framework for solving large-scale multistage stochastic mixed 0–1 problems with nonsymmetric scenario trees. Part II: Parallelization. *Computers & Operations Research, 40*, 2950–2960.

Baldassarre, S., Bruno, G., Diglio, A., & Piccolo, C. (2023). A sample approximation solution procedure for chance-constrained districting problems. *Computers & Operations Research, 160*, 106376.

Baron, O., Milner, J., & Naseraldin, H. (2011). Facility location: A robust optimization approach. *Production and Operations Management, 20*, 772–785.

Beardwood, J., Halton, J. H., & Hammersley, J. M. (1959). The shortest path through many points. *Mathematical proceedings of the Cambridge philosophical society, 55*(4), 299–327.

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

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

Bruno, G., Esposito, E., Genovese, A., & Piccolo, C. (2016). Institutions and facility mergers in the Italian education system: Models and case studies. *Socio-Economic Planning Sciences, 53*, 23–32.

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

Diglio, A., Nickel, S., & Saldanha-da Gama, F. (2020). Towards a stochastic programming modeling framework for districting. *Annals of Operations Research, 292*(1), 249–285.

Diglio, A., Peiró, J., Piccolo, C., & Saldanha-da-Gama, F. (2021). Solutions for districting problems with chance-constrained balancing requirements. *Omega, 103*, 102430.

Diglio, A., Peiró, J., Piccolo, C., & Saldanha-da Gama, F. (2023). Approximation schemes for districting problems with probabilistic constraints. *European Journal of Operational Research, 307*(1), 233–248.

Escudero, L., Garín, M., Merino, M., & Pérez, G. (2007). The value of the stochastic solution in multistage problems. *TOP, 15*, 48–64.

Escudero, L., Garín, M., Merino, M., & Pérez, G. (2010). On BFC-MSMIP strategies for scenario cluster partitioning, and twin node family branching selection and bounding for multistage stochastic mixed integer programming. *Computers & Operations Research, 37*, 738–753.

Escudero, L., Garín, M., Merino, M., & Pérez, G. (2012). An algorithmic framework for solving large-scale multistage stochastic mixed 0–1 problems with nonsymmetric scenario trees. *Computers & Operations Research, 39*, 1133–1144.

Escudero, L. F., & Monge, J. F. (2018). On capacity expansion planning under strategic and operational uncertainties based on stochastic dominance risk averse management. *Computational Management Science, 15*, 479–500.

Farughi, H., Tavana, M., Mostafayi, S., & Arteaga, F. J. S. (2019). A novel optimization model for designing compact, balanced, and contiguous healthcare districts. *Journal of the Operational Research Society*.

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

Juan, A. A., Faulin, J., Grasman, S. E., Rabe, M., & Figueira, G. (2015). A review of simheuristics: Extending metaheuristics to deal with stochastic combinatorial optimization problems. *Operations Research Perspectives, 2*, 62–72.

Kalcsics, J. and Ríos-Mercado, R. (2019). Districting problems. In G. Laporte, S. Nickel, & F. Saldanha-da-Gama (Eds.), *Location Science* (2nd ed., Chapter 25, pp. 705–743). Springer International Publishing.

Kaut, M., Midthun, K. T., Werner, A. S., Tomasgard, A., Hellemo, L., & Fodstad, M. (2014). Multi-horizon stochastic programming. *Computational Management Science, 11*(1), 179–193.

Lei, H., Laporte, G., & Guo, B. (2012). Districting for routing with stochastic customers. *EURO Journal on Transportation and Logistics, 1*, 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.

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

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. (Ed.). (2020). *Optimal districting and territory design*. Springer International Publishing.

Ríos-Mercado, R. Z., & López-Pérez, J. F. (2013). Commercial territory design planning with realignment and disjoint assignment requirements. *Omega, 41*(3), 525–535.

Salazar-Aguilar, M. A., Ríos-Mercado, R. Z., & Cabrera-Ríos, M. (2011). New models for commercial territory design. *Networks and Spatial Economics, 11*(3), 487–507.

Saldanha-da-Gama, F., & Wang, S. (2024). Territory design. In F. Saldanha-da-Gama & S. Wang (Eds.), *Facility Location Under Uncertainty: Models, Algorithms and Applications* (Chapter 13, pp. 415–436). Springer International Publishing.

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

Yanık, S., Kalcsics, J., Nickel, S., & Bozkaya, B. (2019). A multi-period multi-criteria districting problem applied to primary care scheme with gradual assignment. *International Transactions in Operational Research, 26*(5), 1676–1697.

Ziegler, H.-P. (2012). *Algorithms for linear stochastic programs and their application in supply chain network design problems*. PhD thesis, Karlsruhe Institute of Technology, KIT.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.