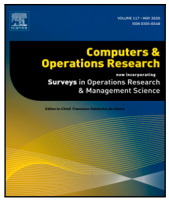
Computers & Operations Research 159 (2023) 106307

Contents lists available at ScienceDirect

Computers and Operations Research

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



Advanced network connectivity features and zonal requirements in Covering Location problems

Serena Fugaro [b,1], Antonino Sgalambro [a,b,*,1]

[a] Sheffield University Management School, Conduit Road, S10 1FL, Sheffield, United Kingdom
[b] Institute for Applications of Calculus, National Research Council of Italy, Via dei Taurini, 19, 00185, Rome, Italy

ARTICLE INFO

Keywords:
Covering Location
Networks
Multi-objective
Matheuristic
Augmented ε-constraint

ABSTRACT

Real-world facility planning problems often require to tackle simultaneously network connectivity and zonal requirements, in order to guarantee an equitable provision of services and an efficient flow of goods, people and information among the facilities. Nonetheless, such challenges have not been addressed jointly so far. In this paper we explore the introduction of advanced network connectivity features and spatial-related requirements within Covering Location Problems. We adopt a broad modelling perspective, accounting for structural and economic aspects of connectivity features, while allowing the choice for one or more facilities to serve the facility networks as depots, and containing the maximal distance between any active facility and such depot(s). A novel class of Multi-objective Covering Location problems are proposed, utilising Mixed Integer Linear Programming as a modelling tool. Aiming at obtaining efficiently the arising Pareto Sets and providing actionable decision-making support throughout real planning processes, we adapt to our problem the robust variant of the AUGMEnted ε-CONstraint method (AUGMECON-R). Furthermore, we exploit the mathematical properties of the proposed problems to design tailored Matheuristic algorithms which boost the scalability of the solution method, with particular reference to the case of multiple depots. By conducting a comprehensive computational study on benchmark instances, we provide a thorough proof of concept for the novel problems, highlighting the challenging nature of the advanced connectivity features and the scalability of the proposed Matheuristics. From a managerial standpoint, the suitability of the proposed work in responding effectively to the motivating needs is showcased.

1. Introduction

The integration of spatial and connectivity modelling features within classic Location Analysis problems allows for extending their practical impact and tackling in a more accurate manner many classes of decision-making problems occurring in both the private and public sectors (Ko et al., 2015). The aim of the present research is to enable the use of optimisation-based decision support methods for those real-world situations that require to optimally locate a set of facilities whilst coping with advanced network connectivity features and zonal requirements arising from specific administrative, managerial, and operational needs. Some natural examples of such situations take place for instance in the field of Healthcare Management: e.g. the adoption of location models is meant to support an optimal design of large scale vaccination campaigns or to efficiently organise a massive collection of medical samples from a large population for analyses. In the former case, in order to meet the immunisation demand of a given community, it is necessary to install one or more vaccination centres for each administrative district; additionally, each active centre needs to be provided with an efficient and prompt supply of vaccines (Shukla et al., 2022). As witnessed during the recent Covid-19 pandemic, such a provision poses a number of daunting challenges in terms of cold supply chain management: indeed, most of the adopted vaccines require very low conservation temperatures, making it paramount to contain the trip length – and hence the travel time – from the logistics depots to the vaccination centres. In the latter case instead, due to the specimens perishable nature, to ensure valid mass screening it is not only necessary to install one or more sample collection units within each administrative district, but also to guarantee a timely and quick delivery of samples to one or more facilities equipped as analyses laboratory. Even in Waste Management, the location of facilities providing fundamental public services has to be designed addressing zonal requirements and guaranteeing suitable connectivity of the resulting network of facilities.

* Corresponding author at: Sheffield University Management School, Conduit Road, S10 1FL, Sheffield, United Kingdom.
E-mail addresses: s.fugaro@iac.cnr.it (S. Fugaro), a.sgalambro@sheffield.ac.uk (A. Sgalambro).
[1] All authors contributed equally.

https://doi.org/10.1016/j.cor.2023.106307
Received 11 November 2022; Received in revised form 7 April 2023; Accepted 7 June 2023
Available online 20 June 2023

For instance, this is the case for the location of Household Waste Recycling Centres, which provide a reuse and recycling service to the residents of municipal districts (WRAP, 2018a,b). These centres receive large quantities of selected materials that cannot be collected by the door-to-door system, and, once sorted, the collected waste is shipped to end points (e.g. landfills and incinerators) via trucks. When it comes to transport hazardous waste, it is of paramount importance to limit the trip length to the final destinations, given the dangerous nature of the materials and the fact that transport generally takes place on public roads, highways and railways (EPA, United States Environmental Protection Agency, 2022).

Indeed, in all these examples the zones are intended as administrative districts or municipalities; this choice is instrumental at addressing all those real-world problems arising, for instance, when public authorities, such as local or regional councils and territorial authorities, need to cope with the optimal design of public facilities networks, securing as much as possible equity and quality in the service provision across different municipalities, counties or district councils.

These examples showcase the potential benefits deriving from integrating classic Covering Location problems with joint zonal requirements and advanced connectivity features while locating a set of facilities and selecting among them special nodes serving as depots. More in general, spatial equity and zonal distribution represent major concerns to address when locating either *desirable* or *obnoxious* facilities in widespread areas characterised by local and regional divisions, including health districts, counties, and neighbourhoods. Disregarding these types of constraints might result for instance in solutions that favour urban areas and those with a high demographic density to the disadvantage of rural areas (Chukwusa et al., 2019).

Before proceeding with an overview of related literature, we would like to clarify that in the rest of the paper, the terms *located* and *active* will be used as synonyms when referring to installed facilities. In the extant literature, spatial-related requirements have been addressed only for a restricted subclass of Location problems: in their seminal paper Revelle and Elzinga (1989) formulated the p-median problem in which the reference area is divided in non-overlapping zones and, for each zone, at least one facility has to be located. Also, in their model, each active facility is enabled to cover only the demand points lying in the corresponding zone. By contrast, demand can be covered by the nearest active facility – independently on the zone – in the m-median and m-centre problems defined in Berman et al. (1991). In addition, they addressed the case of overlapping and not contiguous zones. Instead, Church (1990) modelled a variant of the p-median problem in which regional requirements are employed to limit the number of facilities in each zone. Later on, Gerrard and Church (1994) addressed this same problem with a Multi-objective formulation, so as to detect a trade-off between efficiency and equity. Additionally, Gerrard and Church (1995) generalised the problem in Church (1990) by considering overlapping zones, while Murray and Gerrard (1997) also included capacities for facilities in order to encompass workload balance.

On the other hand, connectivity features in Facility Location are utilised to enable efficient flows of goods, people or information across the set of located facilities. Typically, this is obtained through constraints that refer to distance-based properties within located facilities: e.g. a threshold on the Euclidean distance between two facilities is adopted as a tool to improve the quality of the overall connections in Cherkesly et al. (2019). Additionally, the description of Covering and Median Location problems with interconnected facilities is formalised in that paper. Indeed, connectivity features in Location typically resort to utilising specific graph topologies; in this regard, a reference paper is the work of Demaine et al. (2009) on the design of a tree-structure linking forest fire-fighters. Later on, the same environment was employed in Romich et al. (2015) to define a connected placement of sensors. More recently, Blanco and Gázquez (2021) analysed different topological structures for the continuous Maximal Covering Location Problem with interconnected facilities.

However, in the scientific literature, connectivity features in Facility Location have been addressed primarily either by considering the distances between allocated demand and located facility or from routing-related and structural standpoints, whilst seeking to minimise service and/or installation costs. Nonetheless, the exploration of spatial-related requirements and connectivity constraints occurred so far only for a limited range of Location problems and no contributions have yet considered them jointly. Additionally, our analysis of the literature revealed the following gaps.

G1. There exists an evident lack of a joint perspective accounting for the structural, economic and operational aspects of connectivity features in Facility Location.
G2. In connectivity features, no measure has yet been introduced to contain the maximal distance, in the network connecting located facilities, between each located facility and the one acting as root of the network.
G3. The typical assumptions made on the network connecting active facilities seem too restrictive as these rely on one single predetermined facility acting as root node and feeding all the located facilities.

With these considerations in mind and aiming at representing and addressing realistic and prominent challenges, in this paper we incorporate zonal requirements and advanced connectivity features into a multiple objective framework for a key class of Location problems. In particular, with reference to the highlighted gaps, mathematical modelling features are introduced to:

1. embed interconnection costs among the objective functions;
2. extend the range of criteria considered to assess the quality of facility interconnection;
3. allow for the selection of single or multiple nodes (i.e. facilities) acting as root(s) (i.e. depots or distribution centres) for the underlying network of active facilities.

Two original problems are introduced and mathematically characterised: the *Multi-objective Covering Location Problem with Zonal Requirements and Shortest Path Tree of Active Facilities* (MoCLP-ZSPT), and the *Multi-objective Covering Location Problem with Zonal Requirements and Shortest Path Forest of Active Facilities* (MoCLP-ZSPF). In both cases, the aim is to determine an optimal location of facilities with one facility being active within each zone, while guaranteeing the minimisation of the overall costs and the limitation of the maximum length of any path from root to destination in the underlying network of active facilities. Additionally, the MoCLP-ZSPT entails the choice of one root, while multiple roots can be installed in the MoCLP-ZSPF. Fig. 1 shows a feasible solution for an instance of the MoCLP-ZSPT.

Specific contributions of the present work are:

- the definition of two novel Multi-objective problems, to introduce and concurrently represent advanced network connectivity features and zonal requirements for the set of located facilities within Covering Location problems;
- a complexity characterisation of the proposed problems along with mathematical formulations based on Multi-objective Mixed Integer Linear Programming (MILP) models;
- the adoption and tailored implementation of the robust version of the Augmented ε-constraint framework (AUGMECON-R) (Nikas et al., 2020), as a tool for an exact yet efficient exploration of the Pareto Set for medium sized instances;
- an original Heuristics exploiting the mathematical properties of the considered problems to obtain good quality approximations of the nadir points;
- original tailored Matheuristic algorithms exploiting the mathematical properties of the considered problems to boost the scalability of the solution approach, thus allowing to tackle large size instances and particular configurations;

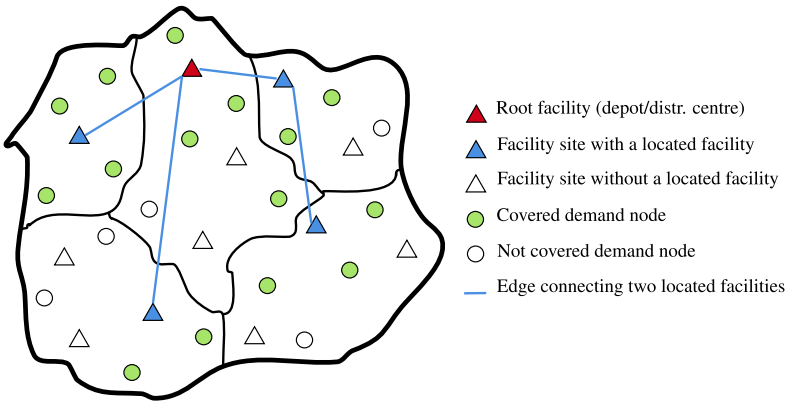


Fig. 1. Region partitioned in zone; the figure shows the shortest path tree connecting a set of active facilities (blue triangles) and depicts in green the demand nodes covered by these facilities.

- a thorough computational experimentation conducted on benchmark instances aimed at: providing a proof of concept of the proposed models, detecting which problem features result most challenging and exploring the scalability of the solution approaches;
- the optimal design of connected networks of located facilities providing maximal coverage for demand of services while accounting for economic, strategic and operational aspects of the interconnections.

In order to foster a smooth and progressive understanding of the novel problem setting, along with its modelling features, we first introduce the MoCLP-ZSPT, namely the version of the problem where only one root (depot or distribution centre) is encompassed. In Section 2 we detail the problem and propose a MILP formulation, whilst Section 3 outlines the proposed solution frameworks and Section 4 presents the computational experiments, along with a thorough analysis of the obtained results. The more general and convoluted MoCLP-ZSPF is then presented in Section 5 providing a mathematical formulation, detailing the ad-hoc designed Matheuristic procedure for this problem and reporting the related computational experiments. Conclusions and future lines of research are given in Section 6.

2. The Multi-objective Covering Location Problem with Zonal Requirements and Shortest Path Tree of Active Facilities

The aim of this Section is to introduce the *Multi-objective Covering Location Problem with Zonal Requirements and Shortest Path Tree of Active Facilities* (MoCLP-ZSPT). Specifically, in Section 2.1 we provide a description as well as a complexity characterisation of the problem, while in Section 2.2 a Multi-objective MILP formulation is presented.

2.1. Problem description

Given a region divided into K non-overlapping zones, a set of facilities has to be located in order to satisfy demand of service for that region. Formally, let $G = (N, E)$ be an undirected graph, with $N = I \cup J$, and $I \cap J = \emptyset$, where I denotes the set of *demand nodes* and J denotes the set of *facility sites* or *candidates*. Differently from the related literature where the root is a fixed candidate (see e.g. Cherkesly et al., 2019), we assume that each facility site can potentially play the role of root within the network of active facilities, with the roots being defined as follows.

Definition 1. A root facility is a special (active) facility which is tasked to serve as a depot for active facilities or to feed them as a distribution centre. Accordingly, in the following the terms root and depot are used as synonyms. Note that with this definition the root facility only provides an additional service to the network of facilities. This means that, from a demand perspective, all facilities are of the same type.

Additionally, let $\{C_k\}_{k=1\dots K}$ denote the partition of the set of nodes induced by the non-overlapping zones, and suppose that each subset contains at least one demand node and one candidate, as stated in (1):

$$\dot{\cup}_{k\le K} C_k = N, \quad \text{and} \quad C_k \cap I \neq \emptyset, \quad C_k \cap J \neq \emptyset \quad \forall k \le K. \tag{1}$$

Besides, $E = E_{IJ} \cup E_J$ where $E_{IJ} = I \times J$ contains all the edges defining potential assignment of demands to facilities, while $E_J \subset J \times J$ contains all the edges connecting two distinct candidates. It is worth emphasising how this leads to the subgraph $G_J = (J, E_J)$ being complete.

We assume that each edge $[j, v] \in E_J$ is labelled with a non-negative cost c_{jv} and that these labels verify the triangular inequality which is a quite common hypothesis in the literature on Location Problems (Laporte et al., 2019).

Let $d : E_{IJ} \mapsto \mathbb{R}^+$ denote the (e.g. Euclidean) distance function; namely $d([i, j]) = d_{ij}$ denotes the distance from demand node $i \in I$ to candidate $j \in J$; then the set of candidates which can cover the demand in i is $N_i = \{j \in J \mid d_{ij} \le S\}$, where S is the *coverage radius,* i.e. the distance beyond which a demand node is considered *uncovered*. Additionally, let $h : I \mapsto \mathbb{R}^+$ be the *demand* function, e.g. h_i is the number of users to serve at node i, and let $s : J \mapsto \mathbb{R}^+$ be the non-negative *facility activation* cost function. Table 1 summarises the introduced notations.

The MoCLP-ZSPT is based on the following set of decisions:

- selecting exactly one facility to be located for each zone;
- choosing the root among the located facilities (cf. Definition 1);
- selecting a tree stemming from the above root and connecting all the located facilities.

The decision-making process is driven by the pursuit of the following multiple objectives:

1. maximisation of the covered demand;
2. minimisation of the overall service costs, including the costs of installing the facilities and those for connecting each active facility to the root;
3. minimisation of the maximum length of any path, in terms of connection costs, from the selected root to any leaf (i.e. active facility) in the tree.

It is worthwhile highlighting the novelty and relevance of entailing the choice of the root node in the decision-making process, given its impact on the structural aspects of the interconnection of facilities and the related economic costs. Furthermore, in the extant literature no contributions cope with the length of paths from the prescribed root to the active facilities, although this clearly affects the performance in sending flows of goods or information along the network of active facilities.

Table 1
Summary of notations for the description of the MoCLP-ZSPT.

I	Set of demand nodes	d_{ij}	Distance between nodes $i \in I$ and $j \in J$
J	Set of facility sites	N_i	Set of candidates able to cover demand in $i \in I$
E_{IJ}	Set of edges connecting nodes of I and J	h_i	Value of demand at node $i \in I$
E_J	Set of edges connecting distinct nodes of J	s_j	Activation cost for facility $j \in J$
c_{jv}	Non-negative cost label for $[j,v] \in E_J$	C_k	Subset of nodes corresponding to kth zone
S	Coverage radius	K	Number of zones

By contrast, in the MoCLP-ZSPT we bridge this gap by containing the maximum length of paths from the root to the active facilities. In particular, since the optimisation of service costs and maximum path length is aimed at enhancing the above mentioned network performance, we assume that there is no extra cost to pay when a facility also serves as a depot/distribution centre for the network of active facilities. Indeed, this assumption leads to the root choice having an economic impact in terms of connection costs.

Lemma 1. *The MoCLP-ZSPT is **NP**-hard.*

Proof. Suppose that $K = 1$, namely the region is not partitioned; then, since the root needs not be an active facility, only the advanced connectivity features need to be addressed.

Let us assume that the edge labels c and the facility activation cost function are both null. With these hypotheses, the choice of the single facility to activate (because $K = 1$) and the one of the root do not affect the objectives 2. and 3. listed above. Namely, the decision-making process reduces to selecting the facility which maximises the covered demand. Therefore, the MoCLP-ZSPT reduces to a Maximal Covering Location Problem (MCLP) with $p = 1$. Then, MoCLP-ZSPT is **NP**-hard since otherwise the MCLP would be tractable, while it is well-known that this problem is **NP**-hard (Megiddo et al., 1983). □

2.2. Arc-flow multi-objective MILP formulation

In this Section, we detail a mathematical formulation for the MoCLP-ZSPT; in particular, the proposed MILP model relies on the use of *flow* variables (Landete and Marín, 2014) since, as reported in Cherkesly et al. (2019) they allow to obtain a more compact formulation of the tree-structure requirements. At this purpose, we consider the natural orientation of the edges in E_J, obtained by splitting each $[i,j] \in E_j$ in two anti-parallel arcs (i,j) and (j,i). Then, denoting with A_J the corresponding set of arcs, the cost function c is easily extended on A_J by symmetry, letting $c((v,j)) = c((j,v)) = c([v,j])$. Additionally, $\forall j \in J$ let $FS(j) = \{v \in J : \exists (j,v) \in A_J\}$ denote the *forward star* of node j, and $BS(j) = \{v \in J : \exists (v,j) \in A_J\}$ denote its *backward star*.

The following variables are adopted to formulate the MoCLP-ZSPT:

1. binary facility location variables x_j such that $x_j = 1$ if a facility is located in j, $\forall j \in J$, ;
2. binary demand coverage variables y_i, such that $y_i = 1$ if demand node i is covered, $\forall i \in I$;
3. binary root selection variables z_j, such that $z_j = 1$ if facility in j serves as root, $\forall j \in J$;
4. binary edge activation variables e_{ij}, with $e_{ij} = 1$ if edge $[i,j]$ connects active facilities i and j, $\forall [i,j] \in E_J$;
5. non-negative flow variables f_{ij}^v defined $\forall (i,j) \in A_J$ and $\forall v \in J$, denoting the amount of flow sent from root to facility in v through arc $(i,j) \in A_J$.

It is worth emphasising that the MoCLP-ZSPT is intrinsically a Multi-objective problem, given the inherently conflicting nature of the goals to be pursued in it, namely: maximisation of demand coverage, minimisation of service costs, and minimisation of maximum path length. Thus, the model encompasses three objective functions:

1. $F_{DC} = \sum_{i \in I} h_i y_i$, representing the demand coverage;
2. $F_{SC} = \sum_{v \in J} \sum_{(i,j) \in A_J} c_{ij} f_{ij}^v + \sum_{j \in J} s_j x_j$, denoting the overall *service costs*, obtained by summing up the *interconnection costs*, namely the cost of the shortest path tree of active facilities w.r.t. the cost function c, and the facility *activation costs*;
3. $F_{PL} = \max_{v \in J} \left[\sum_{(i,j) \in A_J} c_{ij} f_{ij}^v \right]$, constituting the maximum length of any feasible path.

Indeed, the minimisation of F_{PL} would yield a min–max objective function, which we linearised by introducing: one additional non-negative real variable $\mathcal{P}$ to minimise, and additional constraints $\sum_{(i,j) \in A_J} c_{ij} f_{ij}^v \leq \mathcal{P}$, defined $\forall v \in J$.

The resulting Multi-objective MIP is given by (2) where (2a) maximises the covered demand, (2b) minimises the overall costs, and (2c) minimises the maximum length of any solution path (see Box I). Constraints (2d)–(2e) state that exactly one active facility is located as root of the shortest path tree. Constraints (2f) ensure that a node $i \in I$ is covered only if at least one facility is located in a candidate in N_i, while (2g) state that exactly one facility is located within each zone. (2h)–(2n) are the *activation* constraints; namely, (2h)–(2i) state that both ends of any active edge of E_J are active facilities; (2j) couple the f variables with the corresponding e ones stating that: flow can be sent only along activated edges, and each edge can be traversed only in one direction. Finally, (2k)–(2l) couple the f variables with the z ones, stating that no flow can be sent to the facility designated to be the root of the shortest path tree, and that no flow can enter the root facility. Similarly, Constraints (2m)–(2n) couple the f variables with the x ones and state that flow can be sent only to a located facility, and that no flow can leave the destination node. Then, (2o) are the typical *flow-balancing* constraints, stating that in each solution path the unit flow can be sent only from the root node to the destination node. Additionally, (2p)–(2q) ensure that in each solution path, flow cannot make sub-tour on active edges (which are actually the only possible sub-tour due to constraints (2r), which define the dimension of the tree. (2s)–(2u) are binary constraints for the x, y, z and e variables, and (2v) are the non-negativity constraints for the flow variables.

Remark 1. Though in principle the root needs not be an active facility, for the purpose of this paper, we always assume that this is the case. Anyway, when the root represents a distribution centre rather than a depot, it might be functional to let it coincide with a candidate rather than an active facility, by replacing (2e) with $z_j \leq 1 - x_j$, $\forall j \in J$, and (2r) with $\sum_{[i,j] \in E_J} e_{ij} = \sum_{j \in J} x_j$. □

Remark 2. Zonal requirements refer to a partition of the area for administrative, managerial and operational aspects of service provision, thus only affecting location and connection of facilities. Constraints (2g) do not pose any condition on the coverage of demand for a given zone: demand node $i \in C_l \cap I$ might be covered by a facility located in zone C_p with $l \neq p$, $l,p \leq K$. As such, we are assuming that a demand node can be covered by a facility located within the coverage radius in any zone. Different assumptions can be introduced for specific applications where a more restrictive setting is needed. □

Notably, non linear Constraints (2o) can be linearised by modelling each product $z_i x_v$ with a non-negative variable k_{iv}, $\forall i,v \in$

$$(MoCLP-ZSPT)\ \max F_{DC} \tag{2a}$$

$$\min F_{SC} \tag{2b}$$

$$\min F_{PL} \tag{2c}$$

subject to

$$\sum_{j\in J} z_j = 1 \tag{2d}$$

$$z_j \le x_j, \qquad \forall j \in J \tag{2e}$$

$$\sum_{j\in N_i} x_j \ge y_i, \qquad \forall i \in I \tag{2f}$$

$$\sum_{j\in J\cap C_k} x_j = 1, \qquad \forall k = 1 \dots K \tag{2g}$$

$$e_{ij} \le x_i, \qquad \forall [i,j] \in E_J \tag{2h}$$

$$e_{ij} \le x_j, \qquad \forall [i,j] \in E_J \tag{2i}$$

$$f_{ij}^v + f_{ji}^v \le e_{ij}, \qquad \forall [i,j] \in E_J, \forall v \in J \tag{2j}$$

$$f_{ij}^v \le 1 - z_v, \qquad \forall (i,j) \in A_J, \forall v \in J \tag{2k}$$

$$f_{ji}^v \le 1 - z_i, \qquad \forall i,j,v \in J, j \ne i \tag{2l}$$

$$f_{ij}^v \le x_v, \qquad \forall (i,j) \in A_J, \forall v \in J \tag{2m}$$

$$f_{vi}^v \le 1 - x_v, \qquad \forall i,v \in J, i \ne v \tag{2n}$$

$$\sum_{j\in FS(i)} f_{ij}^v - \sum_{j\in BS(i)} f_{ji}^v = \begin{cases} (z_i - 1)x_v & \text{if } i = v, \\ z_i x_v & \text{otherwise.} \end{cases} \qquad \forall i,v \in J \tag{2o}$$

$$\sum_{j\in FS(i)} f_{ij}^v \le x_v, \qquad \forall i,v \in J \tag{2p}$$

$$\sum_{j\in BS(i)} f_{ji}^v \le x_v, \qquad \forall i,v \in J \tag{2q}$$

$$\sum_{[i,j]\in E_J} e_{ij} = \sum_{j\in J} x_j - 1, \tag{2r}$$

$$x_j, z_j \in \{0,1\}, \qquad \forall j \in J \tag{2s}$$

$$y_i \in \{0,1\}, \qquad \forall i \in I \tag{2t}$$

$$e_{ij} \in \{0,1\}, \qquad \forall [i,j] \in E_J \tag{2u}$$

$$f_{ij}^v \ge 0. \qquad \forall (i,j) \in A_J, \forall v \in J \tag{2v}$$

Box I.

J (Fortet, 1959) and including Constraints (3) in the model (Glover and Woolsey, 1974), thus adopting the so-called *standard linearisation* (Mallach, 2020).

$$k_{iv} \le z_i, \qquad k_{iv} \le x_v, \qquad k_{iv} \ge z_i + x_v - 1. \tag{3}$$

With this linearisation, a Multi-objective MILP formulation is effectively obtained for the MoCLP-ZSPT. In particular, the resulting model encompasses: $|I| + (|J|^2 + 3|J|)/2$ binary variables, $|J|^3 + 1$ continuous variables, and $3 + |I| + K + 9(|J|^2 + |J|^3)/2$ linear constraints.

3. Computing Pareto optimal solutions

In Section 2.2 we highlighted that the nature of the MoCLP-ZSPT is inherently Multi-objective as it contemplates different and possibly conflicting managerial perspectives of the organisation or department installing the facilities and the one operating the service. Therefore, to enable optimal decision-making, an accurate representation (or even a complete identification) and analysis of the *Pareto optimal* solutions is needed. Namely, *Pareto optimal* (or *efficient/non-dominated*) solutions are those solutions of the problem for which it is not possible to improve strictly in one objective function without worsening at least one of the others. In the following we will refer to the set of all these solutions as the *Pareto Set* (Mavrotas, 2009).

At this purpose, we propose a twofold contribution to the solution process: firstly, we adapt the robust version of the Augmented ε-constraint generation method, i.e. AUGMECON-R (Nikas et al., 2020) as an efficient framework to explore the corresponding Pareto Sets. As a second contribution we exploit the mathematical properties of the introduced problems to design a tailored Matheuristic algorithm which is integrated within the AUGMECON-R scheme to boost scalability of such solution method. Section 3.1 briefly outlines the framework of AUGMECON-R, while Section 3.2 gives a thorough description of the proposed Matheuristics.

3.1. AUGMECON-R framework for the MoCLP-ZSPT

The Augmented ε-constraint method has proven to be effective when the target problem includes discrete variables in which case the

size of the Pareto Set is finite (Mavrotas, 2009), as for the MoCLP-ZSPT. It produces (an approximation of) Pareto Sets for a given Multi-objective Problem (MOP) by iteratively solving – exactly or heuristically – a single-objective optimisation problem (SOP), obtained from the MOP.

In particular, the improved version, namely AUGMECON2 (Mavrotas and Florios, 2013), avoids the resolution of redundant SOPs by leveraging the information on the slack/surplus variables featured in the formulation of the SOP. Recently, Nikas et al. (2020) designed a robust variant called AUGMECON-R, effectively addressing some of AUGMECON2 weaknesses. Its paradigm is implemented through an integer $(p-1)$-dimensional array *flag*, with p being the number of objective functions, which is initialised with zero values. At each iteration: if the corresponding flag value is zero the SOP is solved; otherwise a number of jumps equal to the value of the flag within the loop used to vary the ϵ values of the second objective function – in terms of priority – is performed (Nikas et al., 2020).

To adapt the framework of AUGMECON-R for the MoCLP-ZSPT, we devised a formulation of the SOP by assuming that the maximisation of the covered demand F_{DC} has the highest priority among the three objective functions, while the minimisation of the overall service costs F_{SC} is prioritised to that of F_{PL}. This choice depicts a real-world scenario occurring, for instance, in the public sector with government agencies funding the installation of the service, and being particularly concerned with minimising general dissatisfaction. Furthermore, as to have that all the objective functions must be minimised, thus simplifying the interpretation of the output solutions, we replaced function F_{DC} with $-F_{DC}$.

3.2. Matheuristics for the MoCLP-ZSPT

The SOPs featured in the AUGMECON-R framework are generally solved through an exact solver. However, this approach might become impractical as the size of the instances increases, given the inherent complexity of the MoCLP-ZSPT. Consequently, we exploited the mathematical properties of the problem and designed a Matheuristic approach called *AugStarExplore*. Specifically, the SOPs are solved with the `StarExplore` Matheuristic procedure, based on a straightforward property of the *metric graphs*. At this purpose, recall that a weighted undirected complete graph is said to be *metric* when its cost function verifies the triangular inequality. As observed in Khuller et al. (1995), in such graphs, a shortest path tree w.r.t. a given cost function and rooted in a fixed node r can be computed in linear time; indeed, the star graph centred in r is a shortest path tree. Exploiting the fact that the subgraph G_J in MoCLP-ZSPT is metric, we designed the *AugStarExplore* Matheuristics, whose pseudo-code is given in Algorithm 1.

Among the input parameters of this procedure, α, ω, β, γ and τ, regulate the functioning of the `StarExplore` procedure. In particular: the parameter γ denotes the (minimum) number of iterations performed during each call to the `StarExplore` procedure; α and ω are used in a *pseudo-randomised* procedure; β is used to compute the slack variables and τ is a threshold value on the number of solutions inserted in the pool. Further details on the usage of these parameters are given in the following.

Once that upper and lower bounds for the cost and the path length objective functions have been computed (Line 2), the corresponding ranges are obtained as their difference (Line 3). In particular, these bounds can be computed from the payoff table obtained with the lexicographic approach (Mavrotas and Florios, 2013) or by approximation (Tautenhain et al., 2019). In particular, while there is no guarantee on the quality of the bounds obtained with a generic approximation approach, using the lexicographic method results in overestimating the nadir points (Ehrgott and Tenfelde-Podehl, 2003); therefore a larger grid is obtained, without affecting the quality of the (approximations of) Pareto Sets produced (Mavrotas and Florios, 2013).

Given the ranges, a set of fractional values for the facility variables x_j is obtained (Line 10) by solving the continuous relaxation of the Maximal Covering Location problem featuring Zonal constraints (2g). Then, at each iteration of the AUGMECON-R scheme, the `StarExplore` heuristics is invoked (Line 15); its pseudo-code is given in Algorithm 2.

This procedure receives as inputs: the problem instance, the set of fractional values for the facilities variables $\hat{x}$, the ϵ values, and the parameters α, ω, β, γ and τ. Then, at each iteration, $\hat{x}$ is used to fix a set of active facilities (one per zone) according to the `Pseudo-RandomisedRounding` procedure detailed in Algorithm 3. In particular, unlike standard randomised rounding (Raghavan and Tompson, 1987), the threshold value adopted to check whether a facility has to be located is a convex combination of ω and $\hat{x}$ through α. This choice is intended to ensure that the exploration is guided by the covered demand objective function and at the same time that it features sufficient diversification.

Then, each active facility is set as root in turn, and a star-graph centred in it (i.e. a shortest path tree) is computed. It is noteworthy that there are as many star-graphs as there are zones. For each star-graph, if the corresponding solution verifies the current ϵ constraints (Line 11, Algorithm 2), the procedure checks whether it is repeated or dominated by any previously computed solution (Line 12); if not, it is inserted in the pool. Indeed the algorithm counts the solutions inserted in the pool (Line 15) to determine if the exploration of a region of the grid is promising.

Additionally, for each feasible solution, the max and min slack variables found so far are updated (Lines 17-18). Then, at the end of the γ iterations, if the SOP is not infeasible and at least τ solutions have been added to the pool, with τ given as input (Line 24), then γ additional iterations are performed (Line 25). Finally, the slack variables are obtained as a convex combination of the max and min values detected during the iterations, using the input parameter β (Line 27).

Remark 3. The estimated running time of the *AugStarExplore* procedure is the sum of that for the resolution of the continuous relaxation of the MCLP with zonal constraints on Line 10 (Algorithm 1), which is linear in $|N|$, and $q_{SC} * q_{PL}$ times that of the `StarExplore` heuristics (Algorithm 2) used to solve the SOP at each iteration.

As for the latter, the running time of the pseudo-randomised rounding procedure is $O(|J|)$. Then, the one for computing K star-graphs (one for each zone) on Line 9 amounts to $K * O(K)$, since according to Khuller et al. (1995) the single computation has linear time. Finally the pool checking on Line 12 performs $O(K^2)$ comparisons. To sum up, the expected running time of a single call to the `StarExplore` heuristics is $2\gamma * O(|J| + K^2)$. The worst case occurs when the grid of ϵ values is defined with unitary step and no jump is performed, since *AugStarExplore* invokes the `StarExplore` heuristics $r_{SC} * r_{PL}$ times. In this case, it is straightforward to observe that the expected running time is $O\left(\gamma * r_{SC} * r_{PL} * (|J| + K^2)\right)$. □

4. Computational experiments for the MoCLP-ZSPT

The scope of the numerical experiments presented in this Section is threefold: checking the validity of the proposed model, detecting which instance features pose challenges to its solution and exploring performance and scalability of the *AugStarExplore* Matheuristics.

All the algorithms were implemented with Python 3.8.10 as programming language, while for the lexicographic method (adopted to obtain the payoff tables) we used the built-in function of the library IBM® Decision Optimisation CPLEX® Modelling for Python. The SOPs were solved with ILOG CPLEX® (version 20.1) solver. The experiments were run on a server equipped with two Intel Xeon Gold 6246R 3.4ghz CPUs, 512 GB Ram and Ubuntu Server 20.04. LTS.

Algorithm 1 AugStarExplore Procedure

procedure AugStarExplore($(G, \{C_k\}_{k\leq K}, c, d, \{N_i\}_{i\in I}, h, s), q_{SC}, q_{PL}, \alpha, \omega, \beta, \gamma, \tau$)
  Compute upper bounds (UB) and lower bounds (LB) for F_{SC} and F_{PL}.
  Set $r_{SC} = UB_{F_{SC}} - LB_{F_{SC}}$ and $r_{PL} = UB_{F_{PL}} - LB_{F_{PL}}$. ▷ Ranges (Mavrotas, 2009)
  Set $step_{PL} = r_{PL}/(q_{PL} - 1)$ and $step_{SC} = r_{SC}/(q_{SC} - 1)$.
  Set $q = 0$, $g = 0$ and $Pareto\_Set = \emptyset$.
  State $b_{SC} = 0, b_{PL} = 0, S_{SC} = 0$ and $S_{PL} = 0$. ▷ Bypass coefficients and slack variables
  for $q < q_{PL}$ and $g < q_{SC}$ **do**
    $flag[q, g] = 0$.
  endfor
  $\hat{x}$ = sol. of the continuous relaxation of MCLP featuring constraints (2g).
  while $q < q_{PL}$ **do**
    while $g < q_{SC}$ **do**
      if $flag[q, g] == 0$ **then**
        $\epsilon_{PL} = UB_{F_{PL}} - q * step_{PL}$ and $\epsilon_{SC} = UB_{F_{SC}} - g * step_{SC}$
        $Pareto\_Set, S_{SC}, S_{PL} =$ `StarExplore`$((G, \{C_k\}_{k\leq K}, c, d, \{N_i\}_{i\in I}, h, s), \hat{x}, \epsilon_{PL}, \epsilon_{SC}, \alpha, \omega, \beta, \gamma, \tau)$
        $b_{SC} = \lfloor S_{SC}/step_{SC} \rfloor$ and $b_{PL} = \lfloor S_{PL}/step_{PL} \rfloor$
        Update the flag matrix using bypass coefficients. ▷ See Nikas et al. (2020)
        $g = g + 1$
      else
        $g = g + flag[q, g]$
      endif
    end while
    $q = q + 1$
  end while
  return $Pareto\_Set$ ▷ Pareto Set (approximation)
end procedure

Algorithm 2 StarExplore Procedure

procedure StarExplore$((G, \{C_k\}_{k\leq K}, c, d, \{N_i\}_{i\in I}, h, s), \hat{x}, \epsilon_{PL}, \epsilon_{SC}, \alpha, \omega, \beta, \gamma, \tau$)
  Set $S_{SC} = 0$ and $S_{PL} = 0$. ▷ Slack variables
  Set $S^{max}_{SC} = 0$ and $S^{max}_{PL} = 0$. ▷ Minimum slack variables
  Set $S^{min}_{SC} =$ `INT_MAX` and $S^{min}_{PL} =$ `INT_MAX`. ▷ Maximum slack variables
  Set $good\_sol = 0, infeas\_sol = 0$ and $insert =$ `FALSE`.
  Set $facilities = \emptyset$ and $Pareto\_Set = \emptyset$.
  for $iter = 1$ **to** γ **do**
    $facilities =$ `Pseudo-RandomisedRounding`$(\{C_k\}_{k\leq K}, \hat{x}, \alpha, \omega)$ ▷ Activation of facilities
    Compute all the possible star-graphs centred in $j \in facilities$.
    for each star-graph $\hat{z}$ **do**
      if $F^{\hat{z}}_{SC} \leq \epsilon_{SC}$ and $F^{\hat{z}}_{PL} \leq \epsilon_{PL}$ **then** ▷ The solution is feasible
        $insert =$ `CheckPool`$(\hat{z}, Pareto\_Set)$ ▷ Check for dominated/repeated solution
        if $insert ==$ `TRUE` **then**
          $Pareto\_Set = Pareto\_Set \cup \{\hat{z}\}$
          $good\_sol = good\_sol + 1$.
        endif
        $S^{min}_{SC} = \min(S^{min}_{SC}, \epsilon_{SC} - F_{SC})$ and $S^{max}_{SC} = \max(S^{max}_{SC}, \epsilon_{SC} - F_{SC})$
        $S^{min}_{PL} = \min(S^{min}_{PL}, \epsilon_{PL} - F_{PL})$ and $S^{max}_{PL} = \max(S^{max}_{PL}, \epsilon_{PL} - F_{PL})$
      else
        $infeas\_sol = infeas\_sol + 1$
      endif
    endfor
  endfor
  if $infeas\_sol < K * \gamma$ and $good\_sol \geq \tau$ **then** ▷ SOP feasible and exploration promising.
    Repeat Lines 7-18. ▷ γ extra iterations are performed
  endif
  $S_{SC} = \beta * S^{max}_{SC} + (1 - \beta) * S^{min}_{SC}$ and $S_{PL} = \beta * S^{max}_{PL} + (1 - \beta) * S^{min}_{PL}$
  return $Pareto\_Set, S_{SC}, S_{PL}$
end procedure

Algorithm 3 Pseudo-RandomisedRounding Procedure

procedure Pseudo-RandomisedRounding$(\{C_k\}_{k\leq K}, \hat{x}, \alpha, \omega)$
  Set $facilities = \emptyset$.
  for each zone C_k **do**
    $active =$ `FALSE`.
    while $active ==$ `FALSE` **do**
      for each candidate j in the zone C_k **do**
        $p = rand(0, 1)$. ▷ Random number in $[0, 1]$
        if $p \geq \alpha * \omega + (1 - \alpha) * \hat{x}[j]$ **then**
          $facilities = facilities \cup \{j\}$ ▷ Locate facility in j
          $active =$ `TRUE`.
        endif
      endfor
    endwhile
  endfor
  return $facilities$
end procedure

Section 4.1 proposes a proof of concept for the MoCLP-ZSPT; then, Section 4.2 details the data-sets used in the experiments, while Section 4.3 describes the tuning of the parameters featured in both the solution approaches. Finally, Section 4.4 details the evaluation metrics adopted in the analysis of the results, which are in Sections 4.5 and 4.6. In particular, in these last two Sections we will refer to the Pareto Set approximations obtained with a heuristic use of AUGMECON-R as *Exact Pareto Set approximations*.

4.1. Numerical example for the MoCLP-ZSPT

The example problem consists of 10 facility sites and 18 demand nodes, partitioned in 5 subsets: $J = \{1, 2, \dots 10\}$, $I = \{11, 12, \dots 28\}$ and $K = 5$. The resulting MILP model comprises: 83 binary variables, 1001 continuous variables, and 4976 linear constraints.

Euclidean distances between demand nodes and facility sites are reported in Table 2, while edge labels are shown in Table 3; finally the distance S defining the coverage radius is equal to 3. Thus, for instance, $N_{11} = \{1, 2, 3\}$ and $N_{23} = \{7\}$. To generate the Pareto Set for a MOP, the AUGMECON-R method requires all the objective functions coefficients to be integer; thus, the s (facility activation cost) and the c coefficients are multiplied by 10.

The payoff table is given in Table 4: ranges values are $r_{SC} = 125$ and $r_{PL} = 20$, while we chose $q_{SC} = q_{PL} = 20$, yielding to discretisation

Table 2
Euclidean distances between demand nodes (i) and facility sites (j).

j \ i	1	2	3	4	5	6	7	8	9	10	j \ i	1	2	3	4	5	6	7	8	9	10
11	1.4	1.7	2.7	5	6.8	10.7	8.8	6.2	4.5	6.2	20	8.5	9	6.7	4.3	3.2	1.7	1.3	5.4	7.4	9.3
12	3.2	1.3	1.4	4.2	5.2	9.2	8	6.6	5.7	8	21	9.3	9.3	7.2	5.3	3	1.2	3.3	7	8.8	10.9
13	1.9	3	1.3	2.4	4.7	8.4	6.1	4.1	3.2	5.5	22	10.1	10.7	8.7	6.1	5	1.7	2.5	6.6	8.7	10.5
14	1.9	4.1	3	2.9	5.5	8.7	6.2	3.2	1.7	3.9	23	8.9	10.1	8	5.2	5.3	3.7	1.6	4.5	7	8.4
15	3.7	3.9	1.6	1.4	2.8	6.6	5	4.4	4.6	6.9	24	7	8.4	6.4	3.6	4.7	4.9	1.8	2.5	4.8	6.5
16	7.9	7.7	5.7	3.9	1.5	2.8	3.5	6.2	7.8	10	25	3.9	5.8	4.2	2.3	5	7.3	4.5	1.2	2	4.1
17	6	6.6	4.3	1.9	1.3	4	2.5	4.3	5.8	7.7	26	4.5	6.8	5.7	4.5	7.2	9.2	6.1	1.9	1.3	2
18	5.7	7	4.7	1.9	3	4.7	2.2	2.6	4.4	6.6	27	3.8	6.5	6	5.8	8.5	11.1	8.3	4	2	1
19	6.7	7.5	5.2	2.5	2.4	3.5	1.5	3.9	5.7	7.8	28	2	5	4.5	4.8	7.4	10.5	7.9	4	1.7	2.8

Table 3
Cost labels for edges between distinct facilities.

Edge	Cost	Edge	Cost	Edge	Cost	Edge	Cost	Edge	Cost	Edge	Cost	Edge	Cost	Edge	Cost	Edge	Cost
[1, 2]	3	[1, 7]	0	[2, 4]	4	[2, 9]	1.5	[3, 7]	0	[4, 6]	5	[5, 6]	3.5	[6, 7]	3.5	[7, 9]	4
[1, 3]	0	[1, 8]	2.1	[2, 5]	2.5	[2, 10]	0	[3, 8]	2.1	[4, 7]	7	[5, 7]	5	[6, 8]	2	[7, 10]	3
[1, 4]	7	[1, 9]	4	[2, 6]	1	[3, 4]	7	[3, 9]	4	[4, 8]	5	[5, 8]	5.5	[6, 9]	2.5	[8, 9]	4.5
[1, 5]	5	[1, 10]	3	[2, 7]	3	[3, 5]	5	[3, 10]	3	[4, 9]	4	[5, 9]	2	[6, 10]	1	[8, 10]	3
[1, 6]	3.5	[2, 3]	3	[2, 8]	3	[3, 6]	3.5	[4, 5]	2	[4, 10]	4	[5, 10]	2.5	[7, 8]	2.1	[9, 10]	1.5

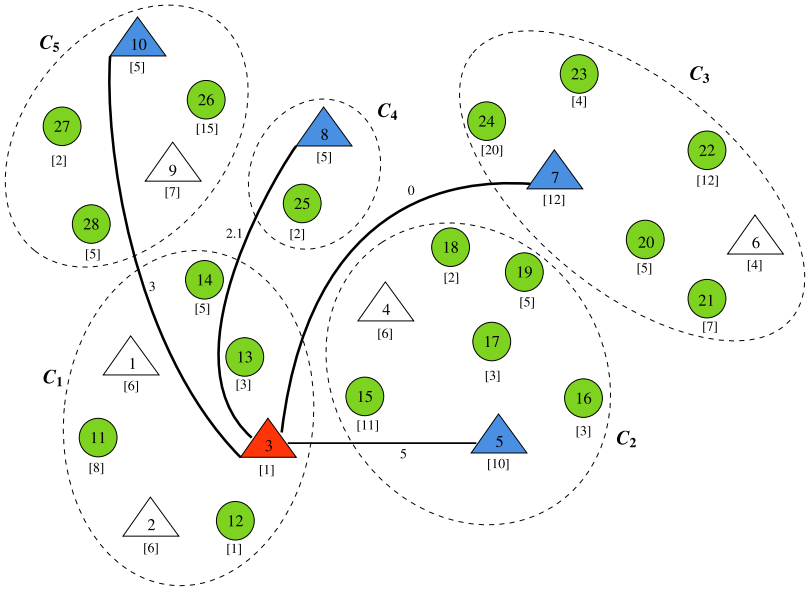


Fig. 2. Solution 2. Active facilities shown in blue, the root facility in red, and covered demand nodes in green. Demand values and facility activation costs in square brackets; $F_{SC} = 43.1$ and $F_{PL} = 5$.

Table 4
Payoff table obtained with lexicographic optimisation.

	$-F_{DC}$	F_{SC}	F_{PL}
$\min -F_{DC}$	−113	431	50
$\min F_{SC}$	−109	320	40
$\min F_{PL}$	−113	445	30

steps equal to 6 and 1, respectively. Therefore, the ε values vary as $\varepsilon_{SC} = 445 - n * 6$ and $\varepsilon_{PL} = 50 - m$ for $n, m = 0, 1, \dots 20$.

AUGMECON-R approximates the Pareto Set with five efficient solutions, whose details are reported in Table 5: for instance, in the first solution $f^{10}_{(10,3)} = f^{10}_{(3,7)} = f^{10}_{(10,5)} = f^{10}_{(10,8)} = 1$ while the remaining flow variables are null. Fig. 2 depicts the second solution.

4.2. Instances

The experiments were conducted on two sets of benchmark instances adapted from the literature on Location Problems with interconnected facilities and Clustered Shortest Path Problems, as detailed in the following. Specifically, in the first case, the instances were obtained by fixing the cardinality of the zones and partitioning the set of nodes accordingly, in order to investigate the scenario in which the zones have regular density. For the second set, on the other hand, the original partition of the instances was used in order to explore the characteristics of the problem in relation to different topologies and types of partitioning.

1. The first data-set consists of *Uncapacitated p-median problem* instances from OR-Library (Beasley, 1990) sized 100 to 600 nodes, used by Cherkesly et al. (2019) in their computational experiments. In order to define the zones, we partitioned the set of nodes by fixing the cardinality of each zone to 25 and 50 respectively, thus obtaining two classes of instances. Next, we generated three families of problems for each class, selecting the candidates as a percentage q of the nodes in each zone, with $q = 0.10, 0.15, 0.20$ for the instances of size 100 and $q = 0.10, 0.15$ for the remaining ones. In this way, we obtained 82 instances whose characteristics are summarised in Table 6.
   Since, in Cherkesly et al. (2019) no activation costs for facilities nor demand values were assigned, we defined these coefficients

Table 5
Efficient solutions found by AUGMECON-R. Those marked with asterisk cover all demand nodes except 23.

Active facilities	Root	Edges	min $-F_{DC}$	min F_{SC}	min F_{PL}
3, 5, 7, 8, 10	10	[10, 3], [3, 7], [10, 5], [10, 8]	−113	44.5	3.0
3, 5, 7, 8, 10	3	[3, 5], [3, 7], [3, 8], [3, 10]	−113	43.1	5.0
3, 4, 6, 8, 10	10	[10, 3], [10, 4], [10, 6], [10, 8]	$-109^{(*)}$	32.0	4.0
3, 5, 7, 8, 10	10	[10, 3], [10, 7], [10, 5], [10, 8]	−113	44.5	3.0
3, 5, 6, 8, 10	10	[10, 3], [10, 5], [10, 6], [10, 8]	$-109^{(*)}$	34.5	3.0

Table 6
Characteristics of the pmed data-set.

Size	Nodes	Demands	Candidates	Zones	Tot. problems
Small	100	80–92	20–8	2, 4	30
Medium	200	172–184	28–16	4, 8	20
Medium	300	258–276	42–24	6, 12	20
Large	400	344–368	56–32	8, 16	4
Large	500	430–460	70–40	10, 20	4
Large	600	516–552	84–48	12, 24	4

Table 7
Characteristics of the RC-CluSPT data-set.

Type	Demands	Candidates	Zones	Tot. problems
1	38–81	11–28	5, 10	14
5	44–99	11–29	5, 10	16
6	39–88	9–26	2, 4, 6, 9	22

as random integers in the set $\{1, 2, \dots, 100\}$ for each demand node, and in the set $\{100, 101, \dots, 200\}$ for each candidate.
To obtain labels verifying the triangular inequality, we defined an appropriate scaling of the costs for the edges in G_J, and added missing edges with random integer cost between 1 and the maximum edge cost. Then, we solved an *All-pairs Shortest Path Problem* on G_J and set c_{jv} as the cost of the shortest path from j to v, for all $j, v \in J$. Costs of the original instances were set as distance values, with missing d_{ij} values chosen as random integers in $\{1, 2, \dots, 100\}$, and S chosen as the maximum distance divided by 5.

2. The second data-set consists of a subset of the High-R instances of Type 1, 5 and 6 used by Ferone et al. (2022) for the *Resource Constrained Clustered Shortest Path Tree Problem*. They are complete graphs whose costs verify the triangular inequality, and for which the set of nodes is partitioned in subsets called *clusters*. As reported in Mestria (2016) these clusters are defined through: k-means (Type 1 networks), geometric centres (Type 5 networks), and grouping in quadrilaterals (Type 6 networks).
We defined two families of problems for each Type as for the pmed data-set, with $q = 0.15, 0.20$: we obtained 52 instances whose characteristics are summarised in Table 7.
Node resource values of the original instance were used as demand values and facility activation costs, while resource values of arcs were used as distance values. The value of S was chosen as the maximum distance divided by 5.

4.3. Tuning of parameters

This Section presents the calibration of the parameters featured in the solution frameworks adopted to approximate the Pareto Sets of the MoCLP-ZSPT. Namely, for AUGMECON-R method we needed to tune the number of grid-points (q_{SC} and q_{PL}) and the coefficient δ featured in the SOP objective function, as reported in Section 4.3.1. Instead, details on the calibration of the parameters featured in the *AugStarExplore* Matheuristics are given Section 4.3.2. They are: α and ω used in the Pseudo-RandomisedRounding procedure; β used to combine the slack variables; the number γ of repetitions of the exploration phase; the threshold value τ on the number of solutions inserted in the pool.

Table 8
Configuration parameters for AUGMECON-R obtained in the tuning phase.

Parameter	Set of values	Selected
δ	$\{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}\}$	10^{-3}
grid-pts	$\{100, 196, 289, 400, 484, 576, 676, 784, 900, 1024\}$	484

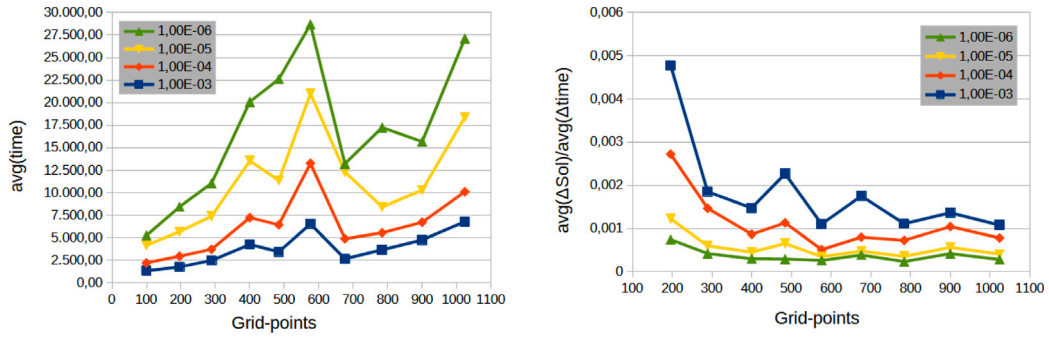
4.3.1. Calibration of parameters for AUGMECON-R method

We considered a sample set consisting of ≈20% of the pmed instances of small and medium sizes. Indeed such a choice relies on the fact that, comprising a greater number of candidates if compared to the RC-CluSPT ones, the pmed instances feature potentially wider ranges, thus making the exact computation of the Pareto Sets significantly time consuming. Moreover, a preliminary experimentation revealed that the computational times on the large instances were extremely prohibitive, therefore only those sized up to 300 nodes were considered. The instances were chosen at random but with one representative for each possible configuration of size, cardinality of zones and percentage of candidates per zone.

A preliminary experimentation revealed that by adopting a unitary step to define the grid of ϵ values, at most 1100 single-objective problems were solved on the considered sample set; thus, we set the number of grid-points on each side of the grid as $\lfloor \sqrt{p} \rfloor$, with $p \in \{100, 200, \dots, 1024\}$. Mavrotas and Florios (2013) suggested to select δ from $\{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}\}$. The calibrated parameters and their respective set of values are reported in Table 8.

Interestingly enough, the experimentation revealed that the lowest average times are relative to $\delta = 10^{-3}$, as depicted in Fig. 3(a). Additionally, the average times grow almost linearly with the number of grid-points until it equals 576 and then from 676 grid-points, for any value of the parameter δ. In fact, as the number of grid-points increases, potentially a greater number of SOPs have to be solved. Indeed, AUGMECON-R has a critical behaviour on two 300 sized instances, which affects the trend of the average values. Namely, the times for the configurations with 576 grid-points doubles and quadruples those relative to 676 grid-points, respectively since on the latter the method performs a higher number of jumps.

However, we conducted a further analysis with a twofold objective: to detect the values of the parameters which would provide the fairest compromise between number of Pareto optimal solutions and computation time; to assess more in depth the impact of increasing the number of grid-points both on times and in terms of efficient solutions determined. At this purpose, we computed the average $avg(\Delta Sol)$ (respect. $avg(\Delta time)$) of the differences between the number of efficient solutions (respect. times) obtained with g and 100 grid-points, with $g > 100$. Clearly, the ratio $avg(\Delta(Sol))/avg(\Delta time)$ grows when either $avg(\Delta Sol)$ increases or $avg(\Delta time)$ decreases, with the value of the other average remaining almost the same. Consequently, the greater this ratio, the better the performance of AUGMECON-R in terms of cardinality of the Pareto Set approximation and relative computational effort. The analysis of these values confirmed that the best choice for the parameter δ is 10^{-3}, as depicted in Fig. 3(b); additionally, the best ratios are obtained when the number of grid-points is either 196 or 484. Actually, comparing this information with the trend of the average times in Fig. 3(a), it comes with no surprise that the highest values corresponds to 196 grid-points, since the average times are lower. Nevertheless, the choice of this value for the parameter could result in a rather dense representation of the Pareto Set for larger instances. Therefore we chose grid-pts = 484 and $\delta = 10^{-3}$.



(a) Average time in seconds for each value of the parameter δ as the number of grid-points is increased from 100 to 1024.

(b) Ratios between the average absolute increase of the solutions number and the average absolute increase of time when increasing grid-points from 100 to each remaining value.

Fig. 3. Representation of data relative to the tuning of AUGMECON-R parameters.

Table 9
Configuration parameters for *AugStarExplore* obtained in the tuning phase.

Parameter	Set of values	Selected	
		pmed	RC-CluSPT
α	$\{0.25, 0.50, 0.75\}$	0.75	0.75
ω	$\{0.20, 0.40, 0.60\}$	0.60	0.60
β	$\{0.25, 0.50, 0.75\}$	0.25	0.25
γ	$\{10, 30, 50, 100\}$	50	100
τ	$\{1, 3, 5\}$	1	5

4.3.2. Calibration of parameters for the AugStarExplore Matheuristic

In order to determine the optimal method configuration based on the specific characteristics of the data-set, we conducted the calibration of the Matheuristic's five parameters by differentiation. That is, in addition to the sample set considered in Section 4.3.1, we chose ≈20% of the RC-CluSPT instances, at random but with at least one representative for each possible configuration of Type and number of zones. The experiments were conducted with unitary step and a time limit of 900 seconds for each test problem of the pmed sample set and of 500 seconds for each test problem of the RC-CluSPT sample set. The calibrated parameters and their respective set of values, for each data-set, are reported in Table 9.

To select the optimal combination of parameters for each data-set, we compared the approximation of the Pareto Set obtained with *AugStarExplore* with the one produced with the AUGMECON-R approach. In more details, we chose the combination corresponding to the minimum value average percentage of heuristic solutions that are dominated by at least one solution obtained with the exact approach used heuristically.

It is worth emphasising that for both parameters used in the convex combinations, i.e. α and β, the set of possible values was chosen as $\{0.25, 0.5, 0.75\}$ thus excluding extreme cases 0 and 1. This choice relies on the fact that $\alpha = 0$ would yield a fixing only guided by the covered demand objective function, which does not guarantee an exhaustive diversification; by contrast $\alpha = 1$ would yield a total random fixing, thus ignoring the priority given to the covered demand objective function. Analogously, $\beta = 0$ would render the bypass coefficients in the AUGMECON-R framework ineffective because as few jumps as possible would be performed. Instead, with $\beta = 1$ a potentially poorer approximation of the Pareto Set would be obtained.

4.4. Evaluation metrics

To assess the efficiency of the Pareto Set representations obtained with the Matheuristics and the (heuristic use of) AUGMECON-R method, we considered the following evaluation metrics.

1. The *Overall Non-dominated Vector Generation*, that is the number of non-dominated solutions obtained with the considered approach.
2. The number μ of non-dominated solutions of one of the two approaches that are actually dominated by the other.
3. The *Overall Pareto Spread* of the Pareto Set approximation $\mathcal{S}$, computed as follows:

$$OPS(\mathcal{S}) = \frac{|\max(-F_{DC}(s)) - \min(-F_{DC}(s))|}{|F_{DC}(s^I) - F_{DC}(s^N)|} \times \frac{|\max F_{SC}(s) - \min F_{SC}(s)|}{|F_{SC}(s^N) - F_{SC}(s^I)|} \times \frac{|\max F_{PL}(s) - \min F_{PL}(s)|}{|F_{PL}(s^N) - F_{PL}(s^I)|}$$

   Max and min values of the objective functions are obtained with reference to the Pareto Set approximation; instead, s^I and s^N denote (approximations of) the ideal and nadir points which are obtained from the payoff table. The larger OPS value, the better it is.
4. The *Spacing* of the Pareto Set $\mathcal{S}$ approximation, computed as follows:

$$SP(\mathcal{S}) = \sqrt{\frac{1}{|\mathcal{S}| - 1} \sum_{i=1}^{|\mathcal{S}|} (\bar{d} - d_i)^2}$$

   where $d_i = \min_{(s_i, s_j) \in \mathcal{S}, s_i \neq s_j} \|(-F_{DC}(s_i), F_{SC}(s_i), F_{PL}(s_i)) - (-F_{DC}(s_j), F_{SC}(s_j), F_{PL}(s_j))\|_1$ and $\bar{d}$ is the mean of the d_i. The smaller $SP(\mathcal{S})$, the higher is the diversification of $\mathcal{S}$.

In particular, 1. and 2. are *cardinality indicators*, 3. is a *spread indicator* measuring the portion of the Pareto Set covered by an approximation; finally, 4. is a *distribution indicator* that measures the quality of the distribution of points on the Pareto Set approximation (Audet et al., 2021).

4.5. Numerical results for the pmed data-set

In this Section we report and analyse the aggregated results relative to the experimentation conducted on the pmed instances. In particular, the payoff tables relative to instances sized 100 and 200 nodes were obtained with the lexicographic optimisation. Instead, the computational times of this approach on the instances sized 300 nodes or more appeared too prohibitive; therefore, to obtain upper and lower bounds for cost and path length functions, we used two heuristic approaches, one adapted from the literature (Tautenhain et al., 2019) and the other based on the StarExplore framework. Section 4.5.1 details these methods, while Section 4.5.2 presents AUGMECON-R results on small and medium instances; then Section 4.5.3 compares them with those obtained by the *AugStarExplore* Matheuristics. Finally, Section 4.5.4 details the results relative to *AugStarExplore* on the large pmed instances.

4.5.1. Heuristic methods to compute the payoff table

The first approach we adopted to approximate the payoff tables consists in optimising each objective function in turn by solving the mathematical model of the MoCLP-ZSPT featuring only one objective function (Tautenhain et al., 2019). However, the resulting bounds might be of poor quality (Isermann and Steuer, 1988). Therefore, we devised a heuristic procedure which produces payoff tables featuring only non-dominated solutions, thus providing overestimates for the nadir points (cf. Ehrgott and Tenfelde-Podehl, 2003).

Specifically, our original method exploits the mathematical properties of the MoCLP-ZSPT, being based on the adaptation of the γ iterations of the `StarExplore` procedure (cf. Algorithm 2, Lines 7-15): these operations are performed thrice, each time starting from a vector $\hat{x}$ obtained by using a criterion based on the optimisation of one of the objective functions in turn. In this way, a more diversified exploration of the efficient set is obtained. The pseudo-code is given in Algorithm 4, while the three criteria are detailed in the following.

`MaxCovering_Criterion`: $\hat{x}$ is obtained by solving the continuous relaxation of MCLP featuring constraints (2g).

`ServiceCost_Criterion`: each node $j \in J$ is used as root and the star-graph centred in it is computed by connecting j with $\hat{i} = argmin_{i \in C_k \cap J}(c_{ji} + s_i)$ for all the zones C_k except the one containing j. Denoted with SC_j the service cost of the star-graph centred in j and computed $minSC = \min_{j \in J} SC_j$, the vector $\hat{x}$ is obtained by assigning to each candidate j the value $\frac{minSC}{SC_j}$.

`PathLength_Criterion`: each node $j \in J$ is used as root and the star-graph centred in it is computed by connecting j with $\hat{i} = argmin_{i \in C_k \cap J}(c_{ji})$ for all the zones C_k except the one containing j. Denoted with PL_j the maximum cost in the star-graph centred in j (i.e. the path length) and computed $minPL = \min_{j \in J} PL_j$, the vector $\hat{x}$ is obtained by assigning to each candidate j the value $\frac{minPL}{PL_j}$.

Algorithm 4 Heuristic Payoff Procedure

procedure HEURISTIC_PAYOFF($(G, \{C_k\}_{k \le K}, c, d, \{N_i\}_{i \in I}, h, s), \alpha, \omega, \gamma$)
  Set $facilities = \emptyset$, $Pool = \emptyset$, $i, j = 0$.
  for $i < 3$ and $j < 3$ **do**
    $payoff[i, j] = 0$
  endfor
  $\hat{x}$ = `MaxCovering_Criterion`
  for $iter = 1$ **to** γ **do**
    $facilities$ = `Pseudo-RandomisedRounding`$(\{C_k\}_{k \le K}, \hat{x}, \alpha, \omega)$ ▷ Activation of facilities
    Compute all the possible star-graphs centred in $j \in facilities$.
    for each star-graph $\hat{z}$ **do**
      $insert$ = `CheckPool`$(\hat{z}, Pool)$ ▷ Check for dominated/repeated solution
      if $insert$ == `TRUE` **then**
        $Pool = Pool \cup \{\hat{z}\}$
      endif
    endfor
  endfor
  $\hat{x}$ = `ServiceCost_Criterion`
  Repeat Lines 7-13.
  $\hat{x}$ = `PathLength_Criterion`
  Repeat Lines 7-13.
  Compute $z_{DC} \in argmax_{Pool}(F_{DC})$, $z_{SC} \in argmin_{Pool}(F_{SC})$, and $z_{PL} \in argmin_{Pool}(F_{PL})$.
  Fill the columns of $payoff$ with objective function values relative to z_{DC}, z_{SC}, z_{PL}.
  return $payoff$.
end procedure

In the experiments we implemented this procedure by using a time limit which depends on the number cb of possible combinations of active facilities over the set J, i.e. $cb = \prod_{k=1}^{K} |C_k \cap J|$. Specifically, the time limit is set as $\max\left\{300, \min\{7200, 0.6 * 0.01 * cb\}\right\}$ seconds, in order to ensure that the procedure runs at least 5 min and at most 2 h, and that at least 1% of cb is explored.

Remark 4. In the following, we will refer to the payoff tables obtained with the method adapted from the literature as "approximated", and to the ones obtained with our heuristics as "heuristic". □

4.5.2. Exact Pareto Set Approximations for small and medium sized instances

Results are given in Tables 10 and 11 which report the number of: nodes, tested instances (Nr.), demand nodes (Dem.), facility sites (Cand.), and zones (Zones), and the percentage of candidates in each zone (q). Additionally, column "avg(Pay-L)" contains the average time in seconds to obtain the payoff table with the lexicographic approach, while column "avg(AUG-R)" reports the average computational time of the AUGMECON-R method. Finally, the average number of: non-dominated solutions (avg(Sol.)), grid-points explored (avg(gp)), jumps for bypass (avg(J-byp)) and for infeasibility (avg(J-inf)) are given.

On the small instances, as the number of candidates increases, the average times for both the lexicographic optimisation and AUGMECON-R increase; such an outcome is in accordance with the fact that for the MoCLP-ZSPT the number of variables and linear constraints are $O(|J|^3)$. In particular, the trend of AUGMECON-R average times depends on the greater number of grid-points explored on average. However, the performance of the methods is also affected by the number of zones growing: at this purpose, it is sufficient to note that the ratio between the average times for instances with 20 candidates and 4 zones and those with 20 candidates and 2 zones is 4.5 and 4.7 respectively. This is because a higher number of zones implies a higher number of candidates to connect, therefore a significant effort is needed to address the advanced connectivity features. Additionally, the statistics on the objective function values revealed that, as expected, the best average values of $-F_{DC}$ are relative to instances with more zones, which allows the activation of more facilities. However, this is at the expense of the overall costs, and consequently of the maximum length of any path from the selected root, which increase.

The results relative to medium instances are reported in Table 12, where column "avg(Pay-A)" refers to average computational times of the approximation approach adapted from the literature (Tautenhain et al., 2019), while column "avg(Pay-H)" reports those relative to our heuristics. As observed for the small instances, also on the medium problems the increase in the number of candidates and zones results in higher average computational times. Notably, as the size of the instances grows, the MoCLP-ZSPT becomes more challenging, as highlighted by the average computational times of the approximated computation of the payoff table (cf. Table 12).

In particular, the Pareto Set approximations relative to the heuristic payoff tables feature more solutions; such an outcome relies on the characteristics of the grids explored by AUGMECON-R. In fact, these results suggest that the approximated payoff tables provide poorer bounds than those obtained with the heuristic ones, thus resulting in wider ranges. Consequently, with the same number of grid-points, the discretisation step is greater with the former ranges, thus providing less dense grids. It is worthwhile mentioning that the average computational times of AUGMECON-R are greater when using the heuristic payoff tables since an increased number of grid-points is explored.

Table 13 reports the values of the adopted evaluation metrics (cf. Section 4.4): column "avg(μ)" is the average number of solutions that are dominated by at least one solution belonging to the other approximation; "avg(OPS)" is the average Overall Pareto Spread, and "avg(SP)" denotes the average Spacing of the corresponding Pareto Set approximation. Finally, the average number of solutions shared by the two methods is reported in the column "avg(sh.)".

Table 10
Aggregated numerical results of the AUGMECON-R method on the small pmed instances.

Nodes	Nr.	q	Dem.	Cand.	Zones	avg(Pay-L)	avg(AUG-R)	avg(Sol.)	avg(gp)	avg(J-byp)	avg(J-inf)
100	5	0.10	90	10	2	0.64 s	6.29 s	4.80	95.80	386.60	1.60
100	5	0.15	86	14	2	1.53 s	10.55 s	4.00	74.00	253.80	55.00
100	5	0.20	80	20	2	4.42 s	46.71 s	9.20	110.20	347.80	26.00
100	5	0.10	92	8	4	0.66 s	6.90 s	4.00	69.40	246.00	10.20
100	5	0.15	88	12	4	2.28 s	37.68 s	8.60	123.20	300.00	34.40
100	5	0.20	80	20	4	19.77 s	219.77 s	14.60	155.20	307.40	21.40

Table 11
Aggregated numerical results of the AUGMECON-R method on the pmed instances of size 200.

Nodes	Nr.	q	Dem.	Cand.	Zones	avg(Pay-L)	avg(AUG-R)	avg(Sol.)	avg(gp)	avg(J-byp)	avg(J-inf)
200	5	0.10	180	20	4	30.72 s	309.18 s	16.00	159.40	222.20	80.40
200	5	0.15	172	28	4	284.14 s	2287.91 s	19.80	189.80	231.20	58.60
200	5	0.10	184	16	8	17.28 s	162.32 s	23.00	172.40	201.40	39.80
200	5	0.15	176	24	8	389.86 s	1862.99 s	26.40	208.60	203.80	10.00

Table 12
Aggregated numerical results of the AUGMECON-R method on the pmed instances of size 300.

Nr.	q	Dem.	Cand.	Zones	Approximated payoff tables				Heuristic payoff tables			
					avg(Pay-A)	avg(AUG-R)	avg(Sol.)	avg(gp)	avg(Pay-H)	avg(AUG-R)	avg(Sol.)	avg(gp)
5	0.10	270	30	6	1065.08 s	4019.46 s	18.80	194.80	300.06 s	7432.66 s	30.40	237.00
5	0.15	258	42	6	11471.07 s	29320.71 s	27.20	225.80	300.02 s	48647.14 s	41.20	256.00
5	0.10	276	24	12	298.96 s	542.91 s	7.00	124.40	300.19 s	1239.26 s	19.25	138.75
5	0.15	264	36	12	8503.98 s	14131.53 s	17.20	213.60	318.88 s	30937.31 s	48.00	262.60

Table 13
Comparison of the aggregated evaluation metrics relative to AUGMECON-R and the medium pmed instances with 300 nodes.

Nodes	q	Zones	Approximated payoff tables					Heuristic payoff tables			
			avg(Sol.)	avg(μ)	avg(OPS)	avg(SP)	avg(sh.)	avg(Sol.)	avg(μ)	avg(OPS)	avg(SP)
300	0.10	6	18.80	1.00	0.05	54.78	13.40	30.40	0.20	0.66	55.22
300	0.15	6	27.00	1.80	0.10	75.06	14.00	41.20	0.60	1.38	63.52
300	0.10	12	7.00	0.60	0.02	87.92	2.60	17.00	0.20	0.96	35.25
300	0.15	12	17.00	0.20	0.05	54.48	10.00	48.00	0.40	0.91	28.45

Table 14
Comparison of the aggregated evaluation metrics relative to the small pmed instances.

Nodes	q	Zones	AUGMECON-R					*AugStarExplore*			
			avg(Sol.)	avg(μ)	avg(OPS)	avg(SP)	avg(sh.)	avg(Sol.)	avg(μ)	avg(OPS)	avg(SP)
100	0.10	2	4.80	0.40	0.88	54.03	4.40	5.20	0.00	1.00	52.03
100	0.15	2	4.00	0.20	0.95	132.03	3.60	4.60	0.20	1.02	123.73
100	0.20	2	8.80	0.00	1.03	62.46	8.80	9.60	0.00	1.05	58.82
100	0.10	4	4.00	0.00	0.89	53.47	3.80	4.20	0.20	1.02	55.83
100	0.15	4	8.40	0.00	0.82	69.82	7.00	11.00	1.40	1.29	76.22
100	0.20	4	14.60	0.00	1.06	66.26	13.80	21.00	0.80	1.31	49.44

These data reveal that the heuristic payoff tables led to better Pareto Sets approximations also in terms of spread and distribution indicators. Indeed, on instances with 6 zones the two approximations share on average 41% of solutions; the average percentage decreases to 17% on the remaining problems.

Finally, the statistics for the objective function values, confirm the trend already observed on the small pmed instances. It is noteworthy that considering the instances with the same number of zones, as the number of candidates increases, the average values of both F_{SC}, and F_{PL} objective functions decrease, as there are generally more possibilities to efficiently address the advanced connectivity features.

4.5.3. Exact vs Heuristic Pareto Set Approximations for small and medium sized instances

The experiments with the *AugStarExplore* Matheuristics were conducted with a unitary step and defining a time limit of 600 s for every 100 nodes (e.g. time limit is 1200s and 1800s on medium instances). Results are given in Tables 14 and 15 which report, for each solution approach, the values of the considered evaluation metrics as in Table 13.

These results show that as the numbers of candidates and zones increases, both the approaches detect a larger number of non-dominated solutions. As already observed, this outcome relies on the greater number of choices to efficiently tackle the connectivity features. Additionally, the heuristic approximation of the Pareto Set contains on average more solutions than the exact one. A closer look at the average number of common solutions shows that, on average, 92% of the exact solutions are also found by the heuristics. Indeed, on the small pmed instances, the average percentage of heuristic solutions dominated by the exact ones is at most 12.92%. As concerns the remaining indicators, the results show that the Matheuristic approximation is able cover a greater portion of the Pareto Set, being its average OPS values bigger. Instead, the diversification of the exact approximation appears better on the instances with 4 zones and $q = 0.10, 0.15$. Overall, these results prove that the quality of the heuristic approximations of the Pareto Set obtained are remarkable.

On the pmed instances sized 200 nodes we can draw conclusions similar to those relative to the small ones as concerns the average number of solutions and μ values, with the percentage of common solutions being $\approx$91%. In particular, as the number of zones doubles, the average

Table 15
Comparison of the aggregated evaluation metrics relative to the medium pmed instances sized 200 nodes.

Nodes	q	Zones	AUGMECON-R					*AugStarExplore*			
			avg(Sol.)	avg(μ)	avg(OPS)	avg(SP)	avg(sh.)	avg(Sol.)	avg(μ)	avg(OPS)	avg(SP)
200	0.10	4	16.00	0.20	1.24	80.03	14.60	17.60	1.40	1.30	75.45
200	0.15	4	19.40	0.20	1.09	75.77	16.20	24.00	2.20	1.24	80.70
200	0.10	8	22.80	0.00	1.15	36.14	21.60	25.20	1.20	1.24	36.60
200	0.15	8	25.60	0.40	0.85	50.57	23.60	38.00	1.20	1.11	40.29

Table 16
Comparison of the aggregated evaluation metrics for medium pmed instances with 300 nodes.

Nodes	q	Zones	AUGMECON-R					*AugStarExplore*			
			avg(Sol.)	avg(μ)	avg(OPS)	avg(SP)	avg(sh.)	avg(Sol.)	avg(μ)	avg(OPS)	avg(SP)
Approximated payoff tables											
300	0.10	6	18.80	0.20	0.05	54.78	15.20	57.00	2.00	0.11	48.36
300	0.15	6	27.00	0.00	0.10	75.06	9.40	82.60	37.20	0.16	50.49
300	0.10	12	7.00	0.00	0.01	78.71	5.00	37.00	8.80	0.03	40.15
300	0.15	12	17.00	0.00	0.06	63.69	3.20	69.20	27.80	0.09	34.22
Heuristic payoff tables											
300	0.10	6	30.40	0.00	0.66	55.22	26.60	53.40	2.20	1.23	59.34
300	0.15	6	41.20	0.00	1.38	63.52	12.60	81.60	46.80	2.02	49.29
300	0.10	12	17.00	0.00	0.96	35.25	15.80	23.00	3.40	1.31	51.25
300	0.15	12	48.00	0.00	0.91	28.45	28.00	64.80	16.20	1.12	24.05

number of heuristic solutions dominated by the exact ones decreases: in terms of percentage, it is at most equal to 8.68% when there are 4 zones, and to 5.06% in case of 8 zones. Also on these instances the Matheuristic approximation covers a greater portion of the Pareto Set, though the exact approximation is slightly better diversified on the instances with 4 (8) zones and 15% (10%) of candidates among nodes.

Table 16 compares the Pareto Set Approximations obtained on the pmed instances with 300 nodes when the payoff tables are either approximated or heuristic.

The scalability of the Matheuristics emerged from these data: it finds on average ≈4.3 (resp. ≈1.9) times the number of solutions detected by the AUGMECON-R method in less than half of its average computational time by using approximated (resp. heuristic) payoff tables. Moreover, on average only the 27.7% of these solutions are dominated by those obtained with the AUGMECON-R. Additionally, the average values of the spread indicators confirm that as the size of the instances increases, the heuristic Pareto Sets approximation is richer and better diversified. In particular, the one obtained with heuristic payoff tables is characteristic by better average values of the Overall Pareto Spread. This outcome confirms that defining the grid from tighter ranges enables the Matheuristics to perform a more exhaustive exploration of the feasible region. At this purpose, it is worthwhile noticing that there is a higher percentage of shared solutions between heuristic and exact approximations relative to approximated payoff tables for instances with less zones; this trend is reversed for those problems with more zones.

Finally, we further analysed the heuristic Pareto Set approximations: as reported in Table 17 we considered the (average) percentage of solutions shared between the two heuristic approximations and those of mutually dominated solutions. As it might be anticipated, a high percentage of the solutions found with the heuristic payoff tables are also determined in the other case, and yet the remaining solutions obtained from approximated payoff tables turn out to be dominated by those obtained with the heuristic ones. This means that, although with reduced cardinalities, the approximations obtained with the heuristic payoff tables are of better quality.

4.5.4. Heuristic Pareto Set Approximations for large sized instances

The results obtained with *AugStarExplore* on medium instances were encouraging, suggesting considerable scalability of the method and good quality of the Pareto Set approximations provided. Consequently, we conducted further experiments on the large pmed instances, for the resolution of which the exact approach (though used heuristically) would present prohibitive computational times, as evidenced by the times involved in the approximated computation of the payoff tables. The results are given in Table 18.

On the instances sized 400 nodes, more solutions are found as the number of zones increases. This trend is reversed on the remaining instances, suggesting that a greater time limit could allow for a more thorough exploration of the Pareto Set, given that as the number of zones increases, so do active facilities to connect and the number of possible shortest path trees. However, adopting the proposed heuristics to compute the payoff tables and thus the ranges, yields to better Pareto Sets approximations with respect to all the evaluation metrics. In fact, a lower percentage of solutions are non-dominated and spread and distribution indicators show how more profitable Pareto Set approximations are detected.

It is worth noting that in less than an hour, the algorithm is able to provide an average of 51.5 efficient solutions to the decision maker, which are sufficiently diversified. This result represents a considerable support to the decision-making process in view of the fact that AUGMECON-R provides a Pareto Set approximation in 3.3 h on instances sized 300 nodes.

4.6. Numerical results for the RC-CluSPT data-set

In this Section we report and analyse the aggregated results relative to the experimentation conducted on the RC-CluSPT instances. For all the test problems, the payoff tables were obtained with the lexicographic approach. Specifically, Section 4.6.1 deals with the results of the AUGMECON-R method, while Section 4.6.2 details the comparison between the results obtained by AUGMECON-R and the *AugStarExplore* Matheuristics on these instances.

4.6.1. Exact Pareto Set Computation

Preliminary tests revealed that the ranges of the F_{SC} and F_{PL} objective function are not too extended, if compared to those of the pmed instances; therefore we set the discretisation step equal to 1 to define the grid-points. It must be underlined that with this choice of the discretisation step, AUGMECON-R performs a complete exploration of the Pareto Set.

Results are given in Table 19 which reports, for each the instance Type the same information given in Table 14. These results show that also on this data-set the increase in the number of candidates affects

Table 17
Comparison of the percentage of shared and dominated solutions for `AugStarExplore` Matheuristics relative to medium `pmed` instances with 300 nodes when payoff tables are computed either by approximation or heuristically.

Nodes	q	Zones	Approx. payoff tables		Heuristic payoff tables	
			avg(%sh.)	avg(%μ)	avg(%sh.)	avg(%μ)
300	0.10	6	88.40	0.00	78.82	5.68
300	0.15	6	47.33	6.17	45.78	35.26
300	0.10	12	98.78	0.00	60.89	0.00
300	0.15	12	29.08	1.95	31.83	57.26

Table 18
Comparison of the numerical results of the Matheuristic on the large `pmed` instances.

Nodes	q	Dem.	Cand.	Zones	Approximated payoff tables					Heuristic payoff tables			
					Sol.	μ	OPS	SP	sh.	Sol.	μ	OPS	SP
400	0.10	360	40	8	51.00	38.00	0.06	41.14	10.00	34.00	0.00	1.06	47.11
400	0.15	344	56	8	46.00	33.00	0.09	34.32	10.00	56.00	0.00	1.93	45.74
400	0.10	368	32	16	87.00	11.00	0.48	24.50	53.00	73.00	2.00	0.99	20.11
400	0.15	352	48	16	84.00	76.00	0.02	41.90	3.00	103.00	3.00	0.90	18.86
500	0.10	450	50	10	61.00	39.00	0.24	51.86	10.00	71.00	11.00	1.18	38.03
500	0.15	430	70	10	54.00	29.00	0.32	36.43	6.00	53.00	19.00	0.89	77.94
500	0.10	460	40	20	33.00	33.00	0.01	35.61	0.00	38.00	0.00	2.40	25.71
500	0.15	440	60	20	36.00	35.00	0.04	42.63	0.00	32.00	0.00	3.88	43.67
600	0.10	540	60	12	68.00	61.00	0.05	38.66	2.00	32.00	1.00	0.94	29.10
600	0.15	516	84	12	58.00	43.00	0.11	85.52	3.00	72.00	12.00	0.97	34.83
600	0.10	552	48	24	16.00	16.00	0.17	45.15	0.00	12.00	0.00	1.13	38.54
600	0.15	528	72	24	24.00	23.00	0.06	38.16	0.00	38.00	1.00	1.06	27.10

Table 19
Aggregated numerical results of the AUGMECON-R method on the `RC-CluSPT` data-set.

avg(Nodes)	Nr.	q	avg(Dem.)	avg(Cand.)	Zones	avg(Pay-L)	avg(AUG-R)	avg(Sol.)	avg(gp)	avg(J-byp)	avg(J-inf)
						Type 1					
65.00	5	0.15	52.60	12.40	5	2.74 s	132.59 s	10.00	335.40	695.40	112.00
65.00	5	0.20	48.60	16.40	5	11.49 s	607.86 s	18.40	576.60	1208.80	525.60
99.50	2	0.15	80.00	19.50	10	481.04 s	113.85 s	5.00	85.50	11.50	150.00
99.50	2	0.20	72.50	27.00	10	45.50 s	6034.33 s	19.00	560.00	200.00	272.00
						Type 5					
80.00	6	0.15	65.50	14.50	5	9.80 s	426.57 s	12.33	370.50	569.33	239.67
80.00	6	0.20	60.83	19.17	5	51.64 s	445.27 s	12.33	258.00	503.50	356.83
105.00	2	0.15	84.50	20.50	10	105.62 s	1175.53 s	18.00	371.00	321.50	156.50
105.00	2	0.20	78.50	26.50	10	1037.03 s	4846.66 s	21.50	488.00	675.50	320.00
						Type 6					
105.00	1	0.15	88.00	17.00	2	2.95 s	91.52 s	8.00	353.00	2371.00	147.00
105.00	1	0.20	83.00	22.00	2	6.37 s	192.61 s	16.00	373.00	2107.00	0.00
63.75	4	0.15	52.25	11.50	4	2.04 s	66.20 s	6.00	244.25	565.25	129.00
63.75	4	0.20	48.75	15.00	4	5.76 s	339.65 s	15.00	423.75	761.25	238.50
73.00	2	0.15	59.50	13.50	6	2.47 s	120.06 s	11.50	275.00	846.00	136.00
73.00	2	0.20	55.00	18.00	6	12.57 s	870.18 s	26.00	638.50	1156.50	136.00
80.75	4	0.15	64.25	16.50	9	11.57 s	323.34 s	12.25	304.50	632.00	104.25
80.75	4	0.20	58.75	22.00	9	121.64 s	1825.73 s	23.50	392.25	529.25	139.75

the average computational times of both lexicographic approach and AUGMECON-R. However, this trend for the latter method, on instances with same Type and number of zones relies on the greater number of grid-points explored.

In particular, on Type 1 instances - whose zones are defined with k-means method - and on Type 5 instances - whose zones are identified through geometric centres -, the average computational times are affected by the doubling of the number of zones. This is due to the reduced zone density, if compared to the Type 6 instances, which renders connectivity features particularly challenging. Notably, on Type 1 instances with 10 zones and $q = 15\%$ the average computational times of the lexicographic optimisation are greater than that of AUGMECON-R method since the latter explores a smaller number of grid-points on these problems. Moreover, the reduced number of candidates for the Type 6 instances – whose zones are defined through quadrilaterals – yields to decreased average computational times, when the number of zones passes from 2 to 4. Additionally, the statistics for the objective function values confirmed the trend observed on the `pmed` data-set; namely, as the number of zones increases, on average $-F_{DC}$ values decrease as there is a greater number of active facilities, though this negatively impacts on the remaining functions, mainly on F_{SC}.

4.6.2. Exact Pareto Sets vs Heuristic Pareto Set Approximations

The experiments with the *AugStarExplore* Matheuristics were conducted considering a unitary step and the parameter setting obtained in the calibration (Section 4.3.2), and using 500 s as time limit. Results are given in Table 20 with the same format adopted in Table 14.

Similarly to what was observed for the `pmed`, the Matheuristics is able to determine a number of solutions on this data-set that grows with the number of candidates and zones. However, the heuristic approximation of the Pareto Set contains on average only 80% of the exact solutions. The average percentage of heuristic solutions dominated by the exact ones is greater on the Type 6 instances (≈20.11%). This outcome relies on the higher zone density of these problems which might lead to a greater number of solutions to explore; consequently the Matheuristics hardly succeeds in improving the pool of solutions found within the allocated time limit. As concerns the remaining indicators, the results show that the Matheuristic approximation of the Pareto Set

Table 20
Comparison of the aggregated evaluation metrics relative to the RC-CluSPT instances.

avg(Nodes)	q	Zones	AUGMECON-R					*AugStarExplore*			
			avg(Sol.)	avg(μ)	avg(OPS)	avg(SP)	avg(sh.)	avg(Sol.)	avg(μ)	avg(OPS)	avg(SP)
	Type 1										
65.00	0.15	5	10.00	0.00	1.00	75.82	7.80	9.80	2.00	1.00	77.76
65.00	0.20	5	18.40	0.00	1.60	70.37	14.40	18.20	3.60	1.57	71.84
99.50	0.15	10	5.00	0.00	2.00	85.22	4.50	5.00	0.50	2.00	85.22
99.50	0.20	10	19.00	0.50	0.50	22.39	14.00	19.50	3.50	0.50	58.24
	Type 5										
80.00	0.15	5	12.33	0.00	1.10	117.01	10.17	12.33	2.00	1.10	117.81
80.00	0.20	5	12.33	0.00	2.77	162.87	10.50	12.67	1.83	2.77	160.64
105.00	0.15	10	18.00	0.00	0.69	40.16	16.50	19.50	1.50	1.27	83.54
105.00	0.20	10	21.50	0.00	1.54	40.12	19.00	20.50	1.50	1.54	40.05
	Type 6										
105.00	0.15	2	8.00	0.00	1.00	676.58	6.00	8.00	2.00	1.00	676.58
105.00	0.20	2	16.00	0.00	1.00	58.11	15.00	15.00	0.00	1.00	77.28
63.75	0.15	4	6.00	0.00	1.00	73.78	4.50	6.00	1.50	1.00	73.78
63.75	0.20	4	15.00	0.00	1.48	41.46	11.50	15.00	3.50	1.48	41.38
73.00	0.15	6	11.50	0.00	1.06	50.24	7.00	11.50	3.50	1.06	51.13
73.00	0.20	6	26.00	0.00	1.17	43.06	20.50	29.50	4.50	1.19	35.11
80.75	0.15	9	11.25	0.00	1.08	43.19	9.00	11.50	2.50	1.08	38.44
80.75	0.20	9	24.50	0.00	1.15	32.13	20.25	25.00	3.75	1.08	30.02

has on average the same Overall Pareto Spread of the complete Pareto Set, given by the exact, though it is slightly less diversified. It is worth emphasising, however, that this approximation of the Pareto Set is obtained with a time limit of 500 s, which is approximately half the average time taken by the AUGMECON-R to calculate the Pareto Set on these instances.

5. The case with multiple roots: the Shortest Path Forest of Active Facilities

As highlighted in Sections 1 and 2, both economic and operational aspects of interconnecting the active facilities can be accounted for by determining where to install the depot (i.e. the root), and by constructing a shortest path tree rooted in it and connecting these sites.

Indeed, specific real-world scenarios, e.g. those characterised by a significant number of active facilities to connect, may necessitate the presence of multiple roots to ensure that the efficiency of connections is preserved. Therefore, we considered a variant of the MoCLP-ZSPT in which, a predetermined number of roots has to be installed. This led us to the definition of a novel Multi-objective problem: the *Multi-objective Covering Location Problem with Zonal Requirements and Shortest Path Forest of Active Facilities* (MoCLP-ZSPF). Similarly to the MoCLP-ZSPT, its decision-making process involves the selection of $\mathcal{R}$ facilities that serve as roots of as many shortest path trees connecting active facilities.

Lemma 2. *The MoCLP-ZSPF is **NP**-hard.*

Proof. Letting $\mathcal{R} = 1$, the MoCLP-ZSPF reduces to the MoCLP-ZSPT. Thus, the MoCLP-ZSPF is **NP**-hard since otherwise the MoCLP-ZSPT would be tractable (Lemma 1). □

A mathematical formulation of the MoCLP-ZSPF is detailed in Section 5.1 while the adaptation of the *AugStarExplore* Matheuristics is described in Section 5.2. Finally, Section 5.3 presents and analyses the experimentation we conducted.

5.1. Arc-flow multi-objective MILP formulation

The proposed MILP model relies again on *flow* variables, though they present four indices this time, since both the origin and destination of flows have to be considered. Specifically, the proposed model relies on the use of facility location, demand coverage, root selection and edge activation variables (cf. Section 2.2), and of the following variables:

1. non-negative flow variables f_{ij}^{uv} defined $\forall u, v \in J, u \neq v$, $\forall (i,j) \in A_J$, denoting the flow sent from facility in u to facility in v through arc (i,j);
2. binary assignment variables t_{uv} defined $\forall u, v \in J, u \neq v$, such that $t_{uv} = 1$ if facility in v is directly connected with facility in u.

The definition of overall service costs and maximum path length objective functions is updated as follows. The former is $F_{SC} = \sum_{\substack{u,v \in J \\ u \neq v}} \sum_{(i,j) \in A_J} c_{ij} f_{ij}^{uv} + \sum_{j \in J} s_j x_j$, where the first term is the *interconnection costs*, i.e. the shortest path forest cost; the latter is $F_{PL} = \max_{v \in J} \sum_{u \in J} \left[\sum_{(i,j) \in A_J} c_{ij} f_{ij}^{uv} \right]$.

Indeed, the min–max objective function resulting from minimising F_{PL} is linearised similarly to what we did in Section 2.2. Thus, the resulting Multi-objective MILP is given by (4) in which (4a) accounts for covered demand maximisation, (4b) denotes the overall costs minimisation, and (4c) represents the minimisation of the maximum length of any solution path (see Box II).

Constraints (2e)–(2i) are adopted to state that the roots are chosen among active facilities, along with coverage and location rule and the activation conditions for the edge variables. Constraints (4d) states that $\mathcal{R}$ active facilities have to be set as roots. Then (4e)–(4j) are the *assignment* constraints: (4e)–(4f) state that any active facility can be linked only to any root facility; (4g) ensure that an active facility is either a root itself or is linked with a root facility, while (4h) ensure that each root has at least an active facility assigned to it. (4i) state that each active and non-root facility is assigned to exactly one root facility, and (4j) force the flow to be sent only from coupled root-destination facilities. Constraints (4k) are the *activation* constraints (cf.(2h)–(2j)). Similarly, (4l) are the typical *flow-balancing* constraints, and Constraints (4m) define the dimension of the shortest path forest. Finally, binary constraints for the x, z, y, e and t variables, and non-negativity constraints for the continuous ones are stated. In particular, the model (4) encompasses: $|I| + (3|J|^2 + |J|)/2$ binary variables, $[|J|(|J|-1)]^2 + 1$ continuous variables, and $3 + |I| + K + 11|J|^2/2 - 4|J|^3 + 5|J|^4/2$ linear constraints.

5.1.1. Numerical example for the MoCLP-ZSPF

Considering the same example problem of Section 4.1, the aim of this subsection is to provide a proof of concept for model (4). In particular, the MILP model comprises: 173 binary variables, 8101 continuous variables, and 21576 linear constraints. We set $S = 3$ and assumed that two roots have to be chosen, i.e. $\mathcal{R} = 2$. The relative payoff table is reported in Table 21.

$$(MoCLP-ZSPF)\ \max F_{DC} \tag{4a}$$

$$\min F_{SC} \tag{4b}$$

$$\min F_{PL} \tag{4c}$$

subject to

Constraints (2e)-(2i)

$$\sum_{j\in J} z_j = \mathcal{R}, \tag{4d}$$

$$t_{uv} \le z_u, \qquad \forall u,v \in J, u \ne v \tag{4e}$$

$$t_{uv} \le x_v, \qquad \forall u,v \in J, u \ne v \tag{4f}$$

$$z_v + t_{uv} \le 1, \qquad \forall u,v \in J, u \ne v \tag{4g}$$

$$\sum_{v\in J\setminus\{u\}} t_{uv} \ge z_u, \qquad \forall u \in J \tag{4h}$$

$$\sum_{u\in J\setminus\{v\}} t_{uv} = x_v - z_v, \qquad \forall v \in J \tag{4i}$$

$$f_{ij}^{uv} \le t_{uv}, \qquad \forall (i,j) \in A_J, \forall u,v \in J, u \ne v \tag{4j}$$

$$f_{ij}^{uv} + f_{ji}^{uv} \le e_{ij}, \qquad \forall [i,j] \in E_J, \forall u,v \in J, u \ne v \tag{4k}$$

$$\sum_{j\in FS(i)} f_{ij}^{uv} - \sum_{j\in BS(i)} f_{ji}^{uv} = \begin{cases} t_{uv} & \text{if } i = u, \\ -t_{uv} & \text{if } i = v, \\ 0, & \text{otherwise.} \end{cases} \qquad \forall i,u,v \in J,\ u \ne v \tag{4l}$$

$$\sum_{[i,j]\in E_J} e_{ij} = \sum_{j\in J} x_j - \mathcal{R}, \tag{4m}$$

Constraints (2s)-(2u)

$$t_{uv} \in \{0,1\}, \qquad \forall u,v \in J, u \ne v \tag{4n}$$

$$f_{ij}^{uv} \ge 0. \qquad \forall (i,j) \in A_J, \forall u,v \in J, u \ne v \tag{4o}$$

Box II.

Table 21
Payoff table obtained with lexicographic optimisation.

	$-F_{DC}$	F_{SC}	F_{PL}
$\min -F_{DC}$	−113	376	25
$\min F_{SC}$	−109	281	40
$\min F_{PL}$	−109	370	20

We chose $q_{SC} = q_{PL} = 20$ yielding to discretisation steps equal to 4 and 1, respectively. AUGMECON-R found eight efficient solutions. In particular, comparing the objective function values in Table 22 with those in Table 5 it emerges that, as expected, adding a new root yields the reduction of the maximum values of both F_{SC} and F_{PL}. For example the first solution activates facilities in $3, 5, 7, 8, 10$ and places the roots in facilities 3 and 5. Also $f^3_{(3,7)} = f^3_{(5,10)} = f^8_{(8,7)} = 1$, and $t_{37} = t_{38} = t_{510} = 1$ while the remaining f and t variables are null. As an example, Fig. 4 shows the second solution from the right-hand side of Table 22.

5.2. Designing a matheuristics for the MoCLP-ZSPF

This Section details the Matheuristics *AugForestExplore*, devised to approximate the Pareto Set for the MoCLP-ZSPF. Actually, the logic behind this approach is similar to that of the *AugStarExplore* algorithm detailed in Section 3.2: it adopts a tailored heuristic procedure, called `ForestExplore`, to solve the SOPs at each iteration of the AUGMECON-R scheme.

At this purpose, it receives as input the same parameters of *AugStarExplore*, and the parameters α, ω, β, γ and τ have the same role as in the `StarExplore` procedure. Once that a set of fractional values for the facility variables x_j is obtained from the continuous relaxation of the Maximal Covering Location problem featuring Zonal constraints (2g), at each iteration the `ForestExplore` heuristics is invoked, whose pseudo-code is given in Algorithm 5.

After the initialisation operations (Lines 2–6), a set of K active facilities is defined through the pseudo-randomised rounding procedure detailed in Algorithm 3 (Line 8). Then, each possible combination of $\mathcal{R}$ active facilities is used as roots to compute a forest (Line 10). At this purpose, the `GetForest` function (Algorithm 6) is invoked which, following the k-means logic, assigns each non-root facility to the closest root. Actually, this is the best assignment that can be made from a Pareto optimality perspective, for functions F_{SC} and F_{PL}, as stated in Lemma 3. It is noteworthy that the number of possible forests is $\binom{K}{\mathcal{R}}$.

Later on, if the forest obtained with `GetForest` presents at least an isolated root, then all these roots are processed sequentially by repeatedly invoking the function `FixForest` (Line 14) detailed in Algorithm 7. This function evaluates the differential contributions related to switching a non-root facility from the root it has been assigned to and the isolated root currently processed. Indeed, switches are possible only from roots with at least two assigned nodes, to avoid creating new isolated roots (Line 6, Algorithm 7). Then, following a minimum criterion, a switch occurs whenever it makes the smallest contribution to either the cost or the path length objective function. Indeed, as many new forests are created as there are possible switches. Then these forests are explored (Lines 18–29, Algorithm 5): for each of them, if the corresponding solution verifies the current ϵ constraints (Line

Table 22
Solutions found by AUGMECON-R. Those marked with asterisk cover all demand nodes except 23.

Active facilities	Roots	Edges	min $-F_{DC}$	min F_{SC}	min F_{PL}	Active facilities	Roots	Edges	min $-F_{DC}$	min F_{SC}	min F_{PL}
3, 5, 7, 8, 10	3, 5	[3, 7], [7, 8], [5, 10]	−113	37.6	2.5	3, 4, 6, 8, 10	3, 10	[3, 8], [4, 10], [6, 10]	$-109^{(*)}$	28.1	4.0
3, 5, 6, 8, 10	8, 10	[3, 8], [5, 10], [6, 10]	$-109^{(*)}$	30.6	2.5	3, 5, 6, 8, 10	3, 10	[3, 8], [5, 10], [6, 10]	$-109^{(*)}$	30.6	2.5
3, 5, 7, 8, 10	3, 5	[3, 7], [3, 8], [5, 10]	−113	37.6	2.5	3, 5, 7, 8, 10	5, 7	[7, 3], [7, 8], [5, 10]	−113	37.6	2.5
3, 5, 6, 8, 9	8, 9	[9, 5], [8, 3], [8, 6]	$-109^{(*)}$	33.1	2.1	2, 5, 6, 8, 9	5, 6	[5, 9], [6, 8], [6, 2]	$-109^{(*)}$	37.0	2.0

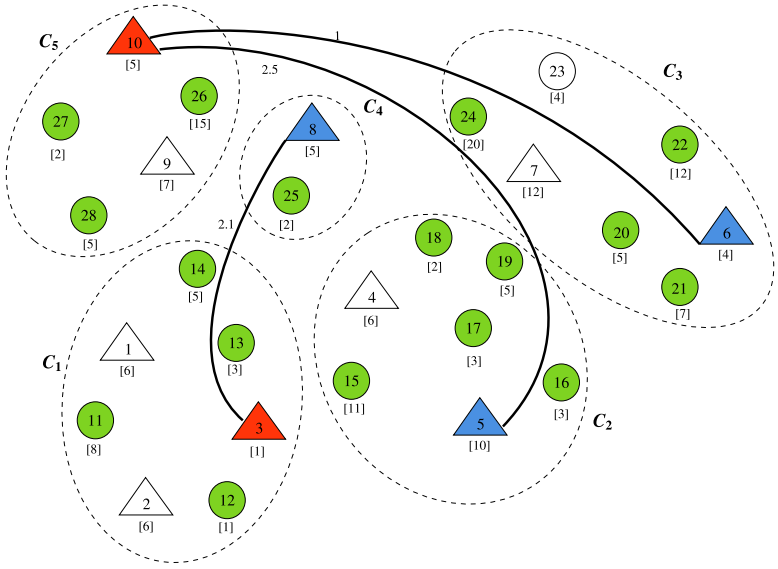


Fig. 4. Solution 2 (right-hand side of Table 22). Active facilities shown in blue while root facilities in red, and covered demand nodes in green. $F_{SC} = 30.6$ and $F_{PL} = 2.5$.

19), the procedure checks whether it is repeated or dominated by any previously computed solution (Line 21); if not, it is inserted in the pool. The procedure counts the solutions inserted in the pool (Line 24) and updates max and min slack variables found (Lines 26–27) analogously to Algorithm 2. If the exploration of the region of the grid is promising, γ additional iterations are performed (Line 36). Finally, the slack variables are obtained as in Algorithm 2.

Lemma 3. *The procedure* `GetForest` *(Algorithm 6) defines a Pareto optimal assignment of active facilities to roots, for the overall service costs objective function and the maximum path length objective function. That is, there does not exist a different assignment which improves the values of functions F_{SC} and F_{PL} obtained with this procedure.*

Proof. Given the vector $\bar{x}$ of facility variables obtained with `PseudoRandomisedRounding` procedure (cf. Algorithm 3), let $\bar{f}$ be the vector of flow variables relative to the assignment obtained with `GetForest` procedure. Let us assume that there exists a different assignment of non-root active facilities in $\bar{x}$ to roots such that, given the corresponding vector $\bar{f}'$ of flow variables, $\left(F_{SC}(\bar{f}', \bar{x}), F_{PL}(\bar{f}')\right)$ dominates $\left(F_{SC}(\bar{f}, \bar{x}), F_{PL}(\bar{f})\right)$. Namely, it holds that:

$$F_{SC}(\bar{f}', \bar{x}) \leq F_{SC}(\bar{f}, \bar{x}) \quad \text{and} \quad F_{PL}(\bar{f}') \leq F_{PL}(\bar{f}),$$

with at least one strict inequality. However, this outcome is only possible if there exists at least one path from a root $r \in J$ to an active (non-root) facility $j \in J$, relative to the assignment $\bar{f}'$, whose cost is less than that of the path from r to j in the assignment $\bar{f}$, i.e. c_{rj}. Nonetheless, the triangular inequality for the cost function c ensures that such a path could not exist. □

Remark 5. Similar to what was observed in Section 3.2, the running time of the *AugForestExplore* procedure is determined by the $q_{SC} * q_{PL}$ calls to the `ForestExplore` heuristics (cf. Algorithm 5). Therefore it is necessary to estimate its running time.

As already mentioned, the execution time of the pseudo-randomised rounding procedure is $O(|J|)$, while both `GetForest` and `FixForest` complexity is $O(K\mathcal{R})$. The estimated execution time for the loop on Lines 12–16 is $O(K\mathcal{R}^2)$. In particular, the cardinality of *Forest_Pool* in the worst case (i.e. when there are $\mathcal{R} - 1$ isolated roots and all the switches are possible) is equal to $\prod_{j=0}^{\mathcal{R}-2}[K - \mathcal{R} - j] = O(K^{(\mathcal{R}-1)})$. Therefore, the pool checking on Line 21 performs at most $O(K^{2(\mathcal{R}-1)})$ comparisons. Since the loop on Lines 9–29 entails $\binom{K}{\mathcal{R}}$ iterations, the expected running time of the `ForestExplore` heuristics is $O\left(\gamma * \binom{K}{\mathcal{R}} * (K^{2(\mathcal{R}-1)} + K\mathcal{R}^2)\right)$.

The worst case occurs when the grid of ε values is defined with unitary step and no jump is performed, since *AugForestExplore* invokes the `ForestExplore` heuristics $r_{SC} * r_{PL}$ times. □

5.3. Computational experiments on the MoCLP-ZSPF

To check the validity of the MoCLP-ZSPF model as well, we performed a series of experiments on a subset of the instances detailed in Section 4.2. Particularly, we adapted the framework of the AUGMECON-R method (Section 3.1) to obtain an approximation of the Pareto Sets which is then compared with the one produced by the *AugForestExplore* Matheuristics.

Remark 6. Since we have assumed that only active facilities can act as roots and that exactly one facility is located in each zone, the maximum number of allowed roots is $\lfloor K/2 \rfloor$. □

Given that, when $\mathcal{R} = 1$ the MoCLP-ZSPF reduces to MoCLP-ZSPT, we first conducted a set of experiments aimed at comparing the two formulations (2) and (4), detailed in Section 5.3.1. Then, Section 5.3.2 presents the approximations of the Pareto Sets obtained with the AUGMECON-R and *AugForestExplore* methods when $\mathcal{R} = 2$ and $\mathcal{R} \leq 6$, respectively.

Algorithm 5 ForestExplore Procedure

procedure ForestExplore($(G, \{C_k\}_{k\leq K}, c, d, \{N_i\}_{i\in I}, h, s, \mathcal{R}), \hat{x}, \varepsilon_{PL}, \varepsilon_{SC}, \alpha, \omega, \beta, \gamma, \tau$)
  Set $S_{SC} = 0$ and $S_{PL} = 0$. ▷ Slack variables
  Set $S_{SC}^{max} = 0$ and $S_{PL}^{max} = 0$. ▷ Minimum slack variables
  Set $S_{SC}^{min} =$ `INT_MAX` and $S_{PL}^{min} =$ `INT_MAX`. ▷ Maximum slack variables
  Set $good\_sol = 0, feas\_sol = 0, infeas\_sol = 0$ and $insert =$ `FALSE`.
  Set $facilities = \emptyset$, $Pool = \emptyset$ and $Forest\_Pool = \emptyset$.
  for $iter = 1$ **to** γ **do**
    $facilities =$ `Pseudo-RandomisedRounding`$(\{C_k\}_{k\leq K}, \hat{x}, \alpha, \omega)$ ▷ Activation of facilities
    for each tuple $roots$ of $\mathcal{R}$ active facilities **do**
      $Forest =$ `GetForest`$(roots, facilities)$
      Add *Forest* to *Forest_Pool*.
      if there is at least an isolated root **then**
        for each isolated root j **do**
          $Forest\_Pool =$ `FixForest`$(j, Forest\_Pool)$
        endfor
      endif
      $feas\_sol = 0$.
      for each forest $\hat{f}$ in *Forest_Pool* **do**
        if $F_{SC}^{\hat{f}} \leq \varepsilon_{SC}$ and $F_{PL}^{\hat{f}} \leq \varepsilon_{PL}$ **then** ▷ The forest is feasible
          $feas\_sol = feas\_sol + 1$.
          $insert =$ `CheckPool`$(\hat{f}, Pool)$ ▷ Check for dominated/repeated solution
          if $insert ==$ `TRUE` **then**
            $Pool = Pool \cup \{\hat{f}\}$
            $good\_sol = good\_sol + 1$.
          endif
          $S_{SC}^{min} = \min(S_{SC}^{min}, \varepsilon_{SC} - F_{SC})$ and $S_{SC}^{max} = \max(S_{SC}^{max}, \varepsilon_{SC} - F_{SC})$
          $S_{PL}^{min} = \min(S_{PL}^{min}, \varepsilon_{PL} - F_{PL})$ and $S_{PL}^{max} = \max(S_{PL}^{max}, \varepsilon_{PL} - F_{PL})$
        endif
      endfor
      if $feas\_sol == 0$ **then** ▷ The SOP in unfeasible
        $infeas\_sol = infeas\_sol + 1$
      endif
    endfor
  endfor
  if $infeas\_sol < \binom{K}{R} * \gamma$ **and** $good\_sol \geq \tau$ **then** ▷ SOP feasible and exploration promising
    Repeat Lines 7-29. ▷ γ extra iterations
  endif
  $S_{SC} = \beta * S_{SC}^{max} + (1-\beta) * S_{SC}^{min}$ and $S_{PL} = \beta * S_{PL}^{max} + (1-\beta) * S_{PL}^{min}$
  return $Pool, S_{SC}, S_{PL}$
end procedure

Algorithm 6 GetForest Procedure

procedure GetForest($roots, facilities$)
  Set $Forest = facilities$.
  Set $F_{SC} = \sum_{j\in facilities} s_j$ and $F_{PL} = 0$.
  for each $v \in facilities \setminus roots$ **do**
    assign v to $j = argmin_{i\in roots} c_{iv}$.
    Update *Forest*.
    $F_{SC} = F_{SC} + c_{jv}$.
    if $c_{jv} > F_{PL}$ **then**
      $F_{PL} = c_{jv}$
    endif
  endfor
  return *Forest*
end procedure

Algorithm 7 FixForest Procedure

procedure FixForest($j, Forest\_Pool$)
  Set $New\_Forest\_Pool = \emptyset$.
  for each forest in *Forest_Pool* **do**
    Remove the forest from *Forest_Pool*.
    for $v \in facilities \setminus roots$ **do**
      if v is assigned to $i \in roots$ **and** at least two nodes are assigned to i **then**
        $\Delta_{SC}[v] = c_{jv} - c_{iv}$ and $\Delta_{PL}[v] = c_{jv}$.
      endif
    endfor
    for $v \in facilities \setminus roots$ **do**
      if $v = argmin\Delta_{SC}$ **or** $v = argmin\Delta_{PL}$ **then**
        assign v to j.
        $F_{SC} = F_{SC} + \Delta_{SC}[v]$ and $F_{PL} = \Delta_{PL}[v]$
        Add the obtained forest to *New_Forest_Pool*.
      endif
    endfor
  endfor
  return *New_Forest_Pool*
end procedure

5.3.1. Comparison of the MoCLP-ZSPT and the MoCLP-ZSPF with single root

The experiments were conducted considering ≈20% of the small `pmed` instances and of the `RC-CluSPT` ones. The payoff tables were obtained with the lexicographic optimisation, while the parameters of the AUGMECON-R method were set analogously to Sections 4.5 and 4.6 respectively. The results are reported in Tables 23 and 24.

These results show that, as expected, the presence of a fourth index in the formulation has a strong impact on the resolution of the SOPs. In fact, comparing the average computational times of the lexicographic optimisation approach for the same subset of instances, it appears that the maximum ratio is 17.68 on the `pmed` problems, and even 402.82 on the `RC-CluSPT` ones. Analogously, the average computational times

Table 23
Comparison of results on the small pmed instances using MoCLP-ZSPT and MoCLP-ZSPF with $\mathcal{R} = 1$.

Nodes	Nr.	q	Dem.	Cand.	Zones	MoCLP-ZSPT			MoCLP-ZSPF		
						avg(Pay-L)	avg(AUG-R)	avg(Sol.)	avg(Pay-L)	avg(AUG-R)	avg(Sol.)
100	3	0.10	90	10	2	0.63 s	6.02 s	4.67	2.61 s	22.73 s	5.00
100	3	0.15	86	14	2	1.52 s	8.01 s	3.00	15.61 s	125.17 s	4.00
100	3	0.20	80	20	2	4.48 s	45.12 s	10.00	64.03 s	833.35 s	10.00
100	3	0.10	92	8	4	0.65 s	6.09 s	3.67	1.88 s	14.62 s	3.67
100	3	0.15	88	12	4	2.26 s	39.43 s	8.00	12.59 s	187.15 s	10.00
100	3	0.20	80	20	4	23.21 s	269.39 s	17.67	410.51 s	3649.20 s	17.67

Table 24
Comparison of results on RC−CluSPT problems using MoCLP-ZSPT and MoCLP-ZSPF with $\mathcal{R} = 1$.

Type	Nodes	Nr.	q	avg(Dem.)	avg(Cand.)	Zones	MoCLP-ZSPT		MoCLP-ZSPF			
							avg(Pay-L)	avg(AUG-R)	avg(Sol.)	avg(Pay-L)	avg(AUG-R)	avg(Sol.)
1	99.00	1	0.15	79.00	20.00	10	81.02 s	41.06 s	2.00	27314.95 s	1878.74 s	2.00
1	68.00	3	0.20	50.67	17.33	5	15.26 s	665.52 s	17.67	631.57 s	10196.75 s	17.33
5	67.50	4	0.20	50.75	16.75	5	8.19 s	181.25 s	10.00	288.36 s	2333.27 s	10.00
6	76.00	1	0.20	58.00	18.00	4	8.68 s	447.24 s	19.00	149.81 s	5251.35 s	18.00
6	76.00	1	0.20	58.00	18.00	4	13.89 s	1272.30 s	32.00	252.64 s	5251.35 s	18.00
6	73.00	3	0.20	52.50	20.50	9	40.62 s	498.80 s	10.50	16362.50 s	15879.33 s	10.50

Table 25
Aggregated numerical results on the small and medium pmed instances with $\mathcal{R} = 2$.

Nodes	Nr.	q	Dem.	Cand	Zones	Roots	avg(Pay-L)	avg(AUG-R)	avg(Sol.)	avg(gp)	avg(J-byp)	avg(J-inf)
100	1	0.10	92	8	4	2	1.62 s	29.80 s	7.00	103.00	381.00	0.00
100	1	0.15	88	12	4	2	10.52 s	195.37 s	13.00	145.00	282.00	57.00
100	1	0.20	80	20	4	2	348.53 s	2887.03 s	10.00	159.00	325.00	0.00
200	2	0.10	184	16	8	2	1031.65 s	3617.05 s	27.00	236.50	226.00	21.50
200	2	0.10	180	20	4	2	635.65 s	3415.26 s	13.50	158.50	280.00	45.50

Table 26
Aggregated numerical results on the RC−CluSPT instances with $\mathcal{R} = 2$.

Type	Nodes	Nr.	q	avg(Dem.)	avg(Cand.)	Zones	Roots	avg(Pay-L)	avg(AUG-R)	avg(Sol.)	avg(gp)	avg(J-byp)	avg(J-inf)
1	76.00	2	0.20	57.00	19.00	5	2	1121.24 s	10080.34 s	19.50	400.00	441.00	534.00
5	75.00	1	0.20	57.00	18.00	5	2	173.89 s	6074.62 s	8.00	395.00	1649.00	0.00
6	76.00	1	0.20	58.00	18.00	6	2	1415.11 s	4898.44 s	16.00	233.00	275.00	234.00
6	70.00	1	0.20	50.00	20.00	9	2	25375.50 s	54637.36 s	27.00	810.00	1016.00	274.00

of the AUGMECON-R method are one order of magnitude lower when the three indices formulation is adopted. This outcome confirms the strong speed-up due to the presence of one less index; in particular, the average ratio of the average computational times is equal to 9.76 on the pmed instances, and 20.28 on the RC−CluSPT. Finally, on the RC−CluSPT problems the exact approximation of the Pareto Set contains a greater number of non-dominated solutions when adopting the MoCLP-ZSPT formulation; while on average only 1 less solution is detected on the pmed ones. However, considering these results, we can conclude that the MoCLP-ZSPT formulation has to be preferred when $\mathcal{R} = 1$.

5.3.2. Results with multiple roots

A further computational phase was conducted on both the data-sets with the aims of: detecting the peculiarities of the MoCLP-ZSPF formulation as the number $\mathcal{R}$ of roots varies, and appraising the scalability of the proposed *AugForestExplore* Matheuristics. Indeed, preliminary experimentation revealed that the MoCLP-ZSPF is more challenging for the AUGMECON-R method already when $\mathcal{R} = 2$, as emerged from the results in Tables 25 and 26.

These results show that activating one additional root leads to average computational times increased by at least one order of magnitude. Though AUGMECON-R explores more grid-points than the single-root case. Additionally, on the small pmed instances, the average computational are strongly impacted when the number of candidates increases. Instead, on the RC−CluSPT instances, the average computational times of the lexicographic optimisation grow dramatically, revealing that as the number of roots increases the advanced connectivity constraints become more challenging due to the smaller number of candidates per zone.

In view of these considerations, we adopted *AugForestExplore* to heuristically solve the MoCLP-ZSPF with up to 6 roots. Specifically, preliminary experiments revealed that with the parameter values detailed in Section 4.3.2, the resulting heuristic Pareto Set approximation was competitive with the exact one. Therefore, the experimentation was conducted with these parameters and setting the time limit as 500 s on the RC−CluSPT instances. In particular, we considered a representative subset of the RC−CluSPT instances.

The results in Table 27 reveal that on the Type 1 instances with the same characteristics (zones and q), as $\mathcal{R}$ increases, the Matheuristics detects on average one less efficient solution. This outcome is again related to the lower zone density which makes the advanced connectivity features more challenging for the heuristic procedure too. Instead, on the Type 6 instances with 9 zones and the same q values, the Pareto Set approximations relative to $\mathcal{R} = 2$ and $\mathcal{R} = 3$ contain on average the same number of solutions. Though, OPS and Spacing values of the latter are worse. These outcomes suggest that an extended time limit could allow the Matheuristics to perform a more thorough exploration of the frontier. However, on this same Type instances, the Matheuristics efficiently performs a thorough exploration for problems with the same demand, candidates and q values, with 4 and 6 zones. This is confirmed by the doubled number of solutions detected.

Concerning the pmed data-set, a preliminary set of experiments revealed that an adequate exploration of their Pareto Sets would require impractical computational times, given the larger size of the instances and the corresponding density of zones. However, the logic underlying the heuristic framework ForestExplore is well suited to a parallel implementation, to which we switched.

Table 27
Aggregated numerical results of the *AugForestExplore* Matheuristic on the RC-CluSPT data-set.

avg(Nodes)	Nr.	q	avg(Dem.)	avg(Cand.)	Zones	Roots	avg(Pay-L)	Matheur.	avg(Sol.)	avg(OPS)	avg(SP)
						Type 1					
68.50	4	0.20	51.25	17.25	5	2	645.52 s	500.00 s	18.50	1.15	49.46
99.50	2	0.15	80.00	19.50	10	2	84695.07 s	500.00 s	5.00	0.89	63.12
99.50	2	0.15	80.00	19.50	10	3	143654.92 s	500.00 s	4.00	0.28	54.00
						Type 5					
67.50	4	0.20	50.75	16.75	5	2	165.70 s	500.00 s	9.50	0.79	110.87
90.00	1	0.20	66.00	24.00	10	3	236628.48 s	500.00 s	14.00	0.01	100.62
						Type 6					
76.00	1	0.20	58.00	18.00	4	2	270.03 s	500.00 s	6.00	0.30	49.69
76.00	1	0.20	58.00	18.00	6	2	1415.11 s	500.00 s	13.00	1.15	45.04
76.00	1	0.15	61.00	15.00	9	2	775.26 s	500.00 s	12.00	0.75	31.22
76.00	1	0.15	61.00	15.00	9	3	290.64 s	500.00 s	12.00	0.06	61.85
73.00	2	0.20	52.50	20.50	9	2	24494.17 s	500.00 s	11.00	0.67	15.37
73.00	2	0.20	52.50	20.50	9	3	28187.94 s	500.00 s	12.00	0.16	82.98

Table 28
Aggregated numerical results of the *AugForestExplore* on the pmed instances with 100 and 200 nodes.

Nodes	Nr.	q	Dem.	Cand.	Zones	Roots	avg(Pay-L)	Matheur.	avg(Sol.)	avg(OPS)	avg(SP)
100	3	0.10	90	10	4	2	1.77 s	301.23 s	3.67	0.44	34.13
100	3	0.15	86	14	4	2	10.57 s	301.21 s	15.33	0.92	47.80
100	3	0.20	80	20	4	2	322.42 s	301.22 s	24.33	1.06	43.47
200	3	0.10	180	20	4	2	1764.61 s	306.02 s	5.33	0.20	37.33
200	3	0.10	184	16	8	2	1283.39 s	311.25 s	20.67	0.66	46.28
200	3	0.10	184	16	8	3	525.59 s	322.47 s	30.00	1.14	40.39
200	3	0.10	184	16	8	4	2340.11 s	328.08 s	23.33	0.68	30.89

On these instances we defined the time limit as $300+\binom{K}{\mathcal{R}} * Nodes * 0.002$ seconds, in order to guarantee a 2-day exploration on the largest problems with 6 roots. In particular, we considered 3 out of 5 instances for the small and medium pmed ones, along with the large ones. Table 28 reports the results obtained on a subset of the pmed instances sized 100 and 200 nodes, for which the payoff tables were obtained with the lexicographic approach.

Recalling that as the number of candidates grows there is potentially a greater number of solutions to explore, we can see how on small instances the Matheuristics succeeds in performing this exploration. In fact, when the number of candidates doubles, it finds 6 times as many solutions which are also characterised by better spreads. Nevertheless, on instances with 200 nodes, when $\mathcal{R} = 2$, as the density of zones increases, the heuristic approximation of the Pareto Set is poorer, in terms of cardinality, spread and distribution indicators. Such an outcome suggests again that zone density affects the performance of the approach and makes the problem more challenging. Similarly, it happens that as the number of roots doubles the average computational times of the lexicographic approach double too, thus confirming that also $\mathcal{R}$ impacts the complexity of the problem. Curiously, the configuration with 3 roots is the one characterised by the best approximation of the Pareto Sets, in terms of cardinality and spread indicators.

Remark 7. The average computational times for the exact computation of the payoff tables grow dramatically as the number of roots increases; thus, we adapted the operations of the γ iterations in the ForestExplore procedure (cf. Algorithm 5, Lines 7–29) to approximate the payoff table for the pmed instances sized 300 to 600 nodes, using a time limit of 900 s. □

Table 29 reports the results relative to instances with 300 nodes: it emerges that on problems with 6 zones, as the number of candidates grows the Matheuristic approximation of the Pareto Set is poorer in terms of cardinality and spread indicators, though the solutions are better diversified. Again, this outcome is related to the increased zone density which challenges the exploration of the pool of feasible forests performed in the ForestExplore procedure. However, the results obtained for instances with 12 zones and $\mathcal{R} = 2$ prove that the parallel implementation of the heuristics is beneficial to the exploration process; in fact, the resulting approximations of the Pareto Sets are better with respect to all indicators in comparison with those obtained for instances with 6 zones.

Additionally, on these instances, for the same $\mathcal{R}$ values, and the same time limits, when the number of candidates increases the Matheuristics provides increasingly better approximations with respect to cardinality and spread indicators. This reveals how successful this approach is in exploring and determining multiple efficient configurations for scenarios characterised by the presence of several depots. In particular, it is noteworthy that these approximations are obtained in at most 15 min.

Finally, Fig. 5 depicts the average of average values of objective functions F_{SC} and F_{PL} as the number of roots varies from 1 to 4. In particular, the values relative to $\mathcal{R} = 1$ are those of the solutions obtained with *AugStarExplore* method. These graphics reveal how allowing for the activation of multiple roots (depots) is instrumental at fulfilling the reachability goal, by decreasing the maximum length of paths from the root to any active facilities in the spanning forest, as well as the overall costs. Indeed, the advantages in terms of path length and related to the installation of multiple roots are already significant when one additional root is located.

The analysis of results relative to the large pmed problems revealed that on the instances with 400 nodes, 16 zones and the same number of candidates, as the number of roots increases, AugForestExplore finds fewer solutions and produces worse Pareto Sets approximations with respect to diversification indicators (cf. Table 30). Reversely, on instances with 8 zones, the approximations with the best values of all the indicators are those relative to the maximum allowed number of roots, i.e. $\mathcal{R} = 4$. Similarly, on instances with 500 nodes, the "worst" approximations are relative to configurations with fewer zones and roots (i.e. $\mathcal{R} = 2$ and $K = 10$), probably due to the reduced time limit that does not allow for exhaustive exploration. On the other hand, for the remaining instances, with the same number of candidates, the most challenging configurations are those with 5 and 6 roots (cf. Table 31). This is because there might be more isolated roots after the call to GetForest (cf. Algorithm 6); consequently, the cardinality of the pool of solutions obtained with FixForest increases rapidly.

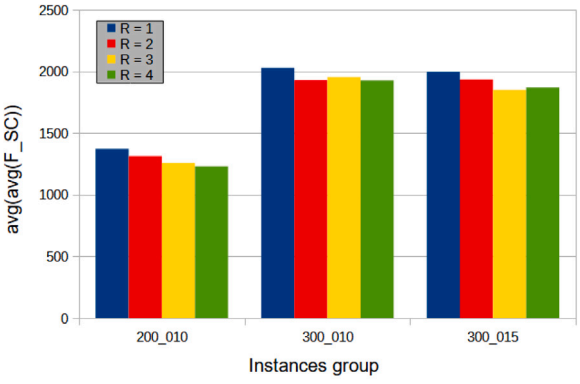
Finally, the data relative to experiments conducted on instances with 600 nodes revealed that, as expected, the higher the number of

Table 29
Aggregated numerical results of the *AugForestExplore* Matheuristic on the pmed instances with 300 nodes.

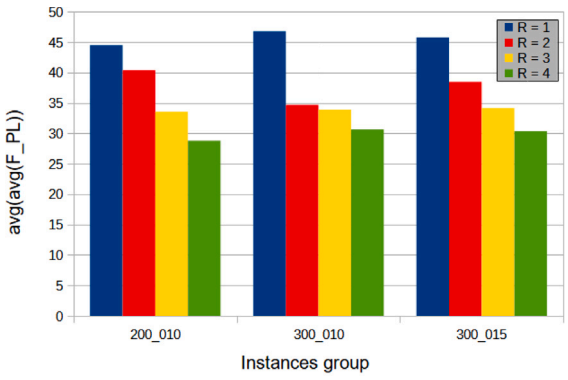
Nodes	Nr.	q	Dem.	Cand.	Zones	Roots	Pay-A	Matheur.	avg(Sol.)	avg(OPS)	avg(SP)
300	3	0.10	270	30	6	2	900 s	315.15 s	8.33	0.16	100.16
300	3	0.15	258	42	6	2	900 s	311.95 s	5.33	0.11	48.50
						2		425.54 s	15.33	0.74	36.67
						3		432.10 s	14.00	1.41	96.43
300	3	0.10	276	24	12	4	900 s	598.11 s	17.00	1.62	55.27
						5		806.21 s	16.67	2.90	94.50
						6		855.73 s	12.67	2.78	99.16
						2		339.66 s	27.67	0.68	52.54
						3		432.13 s	31.33	1.09	55.41
300	3	0.15	264	36	12	4	900 s	597.53 s	31.33	4.93	42.03
						5		776.43 s	19.33	16.84	66.48
						6		855.36 s	23.00	76.62	54.56

Table 30
Numerical results of the *AugForestExplore* Matheuristic on the pmed instances with 400 nodes.

Nodes	q	Dem.	Cand.	Zones	Roots	Pay-A	Matheur.	avg(Sol.)	avg(OPS)	avg(SP)
					2		322.43 s	33	0.49	45.75
400	0.10	360	40	8	3	900 s	344.85 s	16	0.70	62.98
					4		356.12 s	40	3.39	56.71
					2		322.44 s	10	0.40	106.09
400	0.15	344	56	8	3	900 s	344.86 s	20	0.63	100.71
					4		356.17 s	26	0.89	50.60
					2		396.02 s	15	0.50	39.81
					3		749.51 s	42	3.63	30.01
400	0.10	368	32	16	4	900 s	1766.82 s	12	9.58	111.06
					5		3811.70 s	20	0.00	53.40
					6		6726.94 s	10	0.00	70.89
					2		396.10 s	37	0.47	31.72
					3		749.63 s	31	0.79	45.00
400	0.15	352	48	16	4	900 s	1756.71 s	20	3.18	57.99
					5		3798.23 s	19	2117.99	54.77
					6		8080.96 s	6	0.00	118.70



(a) F_{SC} values with $\mathcal{R} = 1, 2, 3, 4$.



(b) F_{PL} values with $\mathcal{R} = 1, 2, 3, 4$.

Fig. 5. Representation of the heuristic average values for F_{SC} (left) and F_{PL} (right) objective functions. Each block of four columns represents average values as $\mathcal{R}$ changes and refers to groups of three medium pmed instances with cardinality of zones equal to 25, and $q = 0.10, 0.15$, as reported in the group name.

zones, the more challenging the MoCLP-ZSPF. In fact, the configurations with $K = 24$ and $\mathcal{R} = 4, 5, 6$ are characterised by very poor approximations of the Pareto Sets, with respect to all the indicators. For example, as reported in Table 32, in more than 2 days of computation the Matheuristics did not detect any solution for the problem with 6 roots. By contrast, the instances with $K = 12$ feature good quality approximations with multiple roots. In particular, it is noteworthy that on these problems the Matheuristic approach is able to detect on average 22 solutions in 15 min which is ≈30% of the time needed by AUGMECON-R to find the same number of solution but on instances with only 200 nodes and 2 roots.

6. Conclusions and future lines of research

This paper introduced advanced network connectivity features and zonal requirements within Covering Location, giving rise to a novel class of NP-hard Multi-objective Covering Location Problems. By adopting a broad modelling perspective, it was possible to fill relevant gaps highlighted in the literature, with the aim of extending the range of applicability of Network optimisation tools and Location Problems to practical contexts. Specifically, this research allowed to address all those real-world scenarios that require not only to locate facilities that provide maximum coverage for a certain demand for services, but also to include the economic, strategic and operational aspects of

Table 31
Numerical results of the *AugForestExplore* Matheuristic on the pmed instances with 500 nodes.

Nodes	q	Dem.	Cand.	Zones	Roots	Pay-A	Matheur.	avg(Sol.)	avg(OPS)	avg(SP)
500	0.10	450	50	10	2	900 s	345.01 s	16	0.50	137.50
					3		420.17 s	49	0.92	45.40
					4		510.17 s	61	3.52	51.64
					5		552.04 s	18	0.64	48.76
500	0.15	430	70	10	2	900 s	345.07 s	8	0.17	104.40
					3		420.02 s	36	1.22	72.15
					4		510.33 s	40	15.67	74.53
					5		552.14 s	17	3.39	66.46
500	0.10	460	40	20	2	900 s	490.04 s	8	0.71	43.60
					3		1441.16 s	9	1.05	91.82
					4		5154.51 s	22	2.05	36.92
					5		15840.61 s	36	0.00	12.79
					6		44939.19 s	5	0.00	89.52
500	0.15	440	60	20	2	900 s	490.18 s	4	0.25	56.10
					3		1440.95 s	14	1.75	103.52
					4		5751.49 s	6	2.22	79.27
					5		16237.91 s	17	0.00	51.08
					6		41841.59 s	2	0.00	0.00

Table 32
Numerical results of the *AugForestExplore* Matheuristic on the pmed instances with 600 nodes.

Nodes	q	Dem.	Cand.	Zones	Roots	Pay-A	Matheur.	avg(Sol.)	avg(OPS)	avg(SP)
600	0.10	540	60	12	2	900 s	379.24 s	19	0.31	41.58
					3		564.18 s	13	0.95	139.58
					4		894.07 s	16	3.60	94.81
					5		1251.18 s	12	5.48	90.19
					6		1412.05 s	23	12.21	54.00
600	0.15	516	84	12	2	900 s	379.30 s	34	0.86	55.67
					3		564.32 s	42	0.72	73.59
					4		1116.41 s	10	0.01	8.34
					5		1251.96 s	29	4.64	61.00
					6		1410.06 s	26	124.14	53.91
600	0.10	552	48	24	2	900 s	631.29 s	6	0.53	50.58
					3		2729.60 s	11	1.57	25.15
					4		15871.66 s	4	0.00	187.28
					5		83180.68 s	7	0.00	33.33
					6		200361.43 s	3	0.00	45.03
600	0.15	528	72	24	2	900 s	631.38 s	22	1.53	26.28
					3		2730.35 s	24	3.19	57.01
					4		13296.22 s	8	0.00	30.77
					5		53667.85 s	3	0.00	28.29
					6		395014.97 s	0	0.00	0.00

their network connections in the decision-making process. To this end, while modelling the problems through Multi-objective Mixed Integer Linear Programming, innovative networks measures were introduced to contain the distance between any active facility and selected depots or distribution centres. In addition, the choice of the network structures to task for these functions was entailed in the decision-making process. As a result of this workflow, an optimal design was defined for those scenarios in which the size of the underlying network is restricted; furthermore, contexts characterised by underlying network structures of significant size were efficiently equipped with an approximate planning of facility location and connection.

The arising optimisation problems are inherently Multi-objective as they contemplate different and conflicting managerial perspectives that come into play when planning systems with the mentioned characteristics. Therefore, this paper proposed a twofold solution approach aimed at providing an accurate representation of their relevant Pareto Sets, in order to support optimal decision-making. Firstly, we tailored the robust version of the Augmented ε-constraint method (AUGMECON-R) (Nikas et al., 2020) for an exact exploration of the Pareto Sets. Secondly, we exploited the mathematical properties of the introduced problems to design tailored Matheuristic algorithms to boost scalability of the solution method, thus enabling to tackle large size instances and multiple depots configurations. In particular, from the thorough experimental analysis conducted, several elements emerged which appear to yield an increased computational burden during the exploration of the Pareto Set, namely: an elevated zone density – in terms of candidates and demands – results in greater computational effort, while zone sparsity yields to significant challenges for the advanced network connectivity purposes. Furthermore, specific topologies of the zones (e.g. not-contiguity) can result in complexities for the fulfilment of all the objectives. Additionally, from a managerial standpoint, it emerged how the proposed research effectively responded to the motivating needs, obtaining a proof of concept for the proposed models while solving benchmark instances of realistic size.

The number of Pareto optimal solutions keeps limited while increasing network size, despite the massive number of feasible solutions. This evidence confirms the suitability of adopting the proposed Multi-objective modelling approach to support real-world decision making, as the final choice can be safely made by policy makers among a limited number of efficient configurations.

On the same note, a characterisation of the problems related to the use of multiple depots/distribution centres was obtained, showing how investing in the installation of multiple depots proves to be cost-effective whenever ensuring a high efficiency in sending flows along the network of active facilities becomes a strategic priority.

As concerns future lines of research, we first aim to develop improved mathematical formulations for the proposed problems. In fact,

the current MILP model for configurations with multiple depots adopts a 4-indices formulation. However, as shown by the experimentation, when the instance size increases, both the resolution of the model and the heuristic approach require a significant computational effort. In addition, this research has enabled bridging the gaps highlighted in the literature on those scenarios in which the primary objective is to maximise the coverage of demand for specific services. Consequently, in an attempt to fill similar and/or further gaps, we intend to broaden our perspective to the cases where the strategic objectives encompass the allocation of demand.

Data availability

No data was used for the research described in the article

References

Audet, C., Bigeon, J., Cartier, D., Le Digabel, S., Salomon, L., 2021. Performance indicators in multiobjective optimization. European J. Oper. Res. 292 (2), 397–422.

Beasley, J., 1990. OR library. http://people.brunel.ac.uk/~mastjjb/jeb/info.html.

Berman, O., Einav, D., Handler, G., 1991. The zone-constrained location problem on a network. European J. Oper. Res. 53 (1), 14–24. http://dx.doi.org/10.1016/0377-2217(91)90089-E.

Blanco, V., Gázquez, R., 2021. Continuous maximal covering location problems with interconnected facilities. Comput. Oper. Res. 132 (May 2020), 105310. http://dx.doi.org/10.1016/j.cor.2021.105310, arXiv:2005.03274.

Cherkesly, M., Landete, M., Laporte, G., 2019. Median and covering location problems with interconnected facilities. Comput. Oper. Res. 107, 1–18. http://dx.doi.org/10.1016/j.cor.2019.03.002.

Chukwusa, E., Verne, J., Polato, G., Taylor, R., Higginson, I.J., Gao, W., 2019. Urban and rural differences in geographical accessibility to inpatient palliative and end-of-life (PEoLC) facilities and place of death: a national population-based study in England, UK. Int. J. Health Geogr. 18 (1), 8.

Church, R.L., 1990. The regionally constrained p-median problem. Geogr. Anal. 22 (1), 22–32.

Demaine, E.D., Hajiaghayi, M., Mahini, H., Sayedi-Roshkhar, A.S., Oveisgharan, S., Zadimoghaddam, M., 2009. Minimizing movement. ACM Trans. Algorithms (TALG) 5 (3), 1–30.

Ehrgott, M., Tenfelde-Podehl, D., 2003. Computation of ideal and nadir values and implications for their use in MCDM methods. European J. Oper. Res. 151 (1), 119–139.

EPA, United States Environmental Protection Agency, 2022. Learn the basics of hazardous waste. Available at https://www.epa.gov/hw/learn-basics-hazardous-waste.

Ferone, D., Festa, P., Fugaro, S., Pastore, T., 2022. The resource constrained clustered shortest path tree problem: mathematical formulation and branch&price solution algorithm. Networks.

Fortet, R., 1959. Applications de l'algèbre de Boole en recherché opérationnelle. Revue Française d'Automatique d'Informatique et de Recherche Opérationnelle 4, 5–36.

Gerrard, R.A., Church, R.L., 1994. Analyzing tradeoffs between zonal constraints and accessibility in facility location. Comput. Oper. Res. 21 (1), 79–99. http://dx.doi.org/10.1016/0305-0548(94)90064-7.

Gerrard, R.A., Church, R.L., 1995. A general construct for the zonally constrained p-median problem. Environ. Plan. B: Plan. Des. 22 (2), 213–236. http://dx.doi.org/10.1068/b220213.

Glover, F., Woolsey, E., 1974. Converting the 0-1 polynomial programming problem to a 0-1 linear program. Oper. Res. 22 (1), 180–182.

Isermann, H., Steuer, R.E., 1988. Computational experience concerning payoff tables and minimum criterion values over the efficient set. European J. Oper. Res. 33 (1), 91–97.

Khuller, S., Raghavachari, B., Young, N., 1995. Balancing minimum spanning trees and shortest-path trees. Algorithmica 14 (4), 305–321.

Ko, J., Nazarian, E., Nam, Y., Guo, Y., 2015. Integrated redistricting, location-allocation and service sharing with intra-district service transfer to reduce demand overload and its disparity. Comput. Environ. Urban Syst. 54, 132–143.

Landete, M., Marín, A., 2014. Looking for edge-equitable spanning trees. Comput. Oper. Res. 41, 44–52. http://dx.doi.org/10.1016/j.cor.2013.07.023, URL https://www.sciencedirect.com/science/article/pii/S0305054813002025.

Laporte, G., Nickel, S., da Gama, F.S., 2019. Location Science. Springer.

Mallach, S., 2020. Inductive linearization for binary quadratic programs with linear constraints. 4OR 1–23.

Mavrotas, G., 2009. Effective implementation of the ϵ-constraint method in multi-objective mathematical programming problems. Appl. Math. Comput. 213 (2), 455–465.

Mavrotas, G., Florios, K., 2013. An improved version of the augmented ϵ-constraint method (AUGMECON2) for finding the exact pareto set in multi-objective integer programming problems. Appl. Math. Comput. 219 (18), 9652–9669.

Megiddo, N., Zemel, E., Hakimi, S.L., 1983. The maximum coverage location problem. SIAM J. Algebr. Discrete Methods 4 (2), 253–261.

Mestria, M., 2016. A hybrid heuristic algorithm for the clustered traveling salesman problem. Pesquisa Operacional 36, 113–132.

Murray, A.T., Gerrard, R.A., 1997. Capacitated service and regional constraints in location-allocation modeling. Locat. Sci. 5 (2), 103–118. http://dx.doi.org/10.1016/S0966-8349(97)00016-8.

Nikas, A., Fountoulakis, A., Forouli, A., Doukas, H., 2020. A robust augmented ϵ-constraint method (AUGMECON-r) for finding exact solutions of multi-objective linear programming problems. Oper. Res. 1–42.

Raghavan, P., Tompson, C.D., 1987. Randomized rounding: a technique for provably good algorithms and algorithmic proofs. Combinatorica 7 (4), 365–374.

Revelle, C., Elzinga, D.J., 1989. An algorithm for facility location in a districted region. Environ. Plan. B: Plan. Des. 16 (1), 41–50. http://dx.doi.org/10.1068/b160041.

Romich, A., Lan, G., Smith, J.C., 2015. A robust sensor covering and communication problem. Nav. Res. Logist. 62 (7), 582–594.

Shukla, S., Fressin, F., Un, M., Coetzer, H., Chaguturu, S.K., 2022. Optimizing vaccine distribution via mobile clinics: a case study on COVID-19 vaccine distribution to long-term care facilities. Vaccine 40 (5), 734–741.

Tautenhain, C.P., Barbosa-Povoa, A.P., Nascimento, M.C., 2019. A multi-objective matheuristic for designing and planning sustainable supply chains. Comput. Ind. Eng. 135, 1203–1223.

WRAP, 2018a. Household waste recycling centre (HWRC) guide. Available at https://wrap.org.uk/sites/default/files/2021-02/HWRC_Guidance_2018_4.pdf.

WRAP, 2018b. Summary: WRAP household waste recycling centres (HWRC) guide. Available at https://wrap.org.uk/sites/default/files/2021-02/HWRC_Summary.pdf.