Multicommodity Flows and Benders Decomposition for Restricted Continuous Location Problems

Murat Oğuz Tolga Bektaş[1] Julia A. Bennell

Southampton Business School
Centre for Operational Research, Management Science and Information Systems (CORMSIS)
University of Southampton, Southampton, United Kingdom, SO17 1BJ

Abstract

The restricted continuous facility location problem arises when there is a need to locate a number of facilities to serve a discrete set of demand points, and where the location of a facility can be anywhere on the plane except for in restricted regions. The problem finds applications in urban planning, disaster management, and healthcare logistics. The restricted regions can occur randomly or are known in advance. The paper describes a new model for the problem that is based on multicommodity flows with unknown destinations and defined on a discretization of the plane. The model and discretization are applied to both the deterministic and the stochastic continuous restricted location problem, where the latter is converted into a deterministic equivalent problem by minimizing the expected value of the objective function weighted by the probabilities of scenarios. The paper also describes a Benders decomposition algorithm to optimally solve the model. Extensive computational results are presented on both benchmark instances from the literature and new instances, on both the deterministic and stochastic variant of the problem. The results indicate that the proposed algorithm is superior to an off-the-shelf solver in terms of computational time. To the best of the authors' knowledge, the exact algorithm described here is the first to address both the deterministic and the stochastic variants of continuous restricted location problems with any number of facilities.

Keywords. location, mathematical modeling, network design, multicommodity network flow, Benders decomposition

1 Introduction

Restricted continuous facility location problems arise in the assembly of printed circuit boards (Foulds and Hamacher, 1994), obnoxious facility planning (Carrizosa and Plastria 1993), urban planning, and locating emergency facilities. These problems are concerned with locating one or more facilities on the plane containing restricted regions. The restriction types can be classified into three categories, namely forbidden regions, barriers and congested regions (Canbolat and Wesolowsky, 2010). In problems with forbidden regions, locating new facilities within the region

[1] Corresponding author.

is not permitted but traveling through them is possible. In the case of barriers, neither locating a facility within the barrier nor travel through the barrier is permitted. In problems with congested regions, locating new facilities within such regions is not allowed, but traveling through them is possible with at the expense of a penalty.

A restricted continuous location problem is said to be deterministic if the locations of all restricted regions are known in advance of locating the facilities and do not change over time. In stochastic restricted location problems, restricted regions occur randomly, as a result of real-world situations such as sudden railway system failures, accidents, unplanned construction and natural disasters, as a result of which random obstacles arise. Natural disasters such as earthquakes, typhoons and hurricanes, for example, result in barriers due to the collapse of buildings, viaducts and bridges, crushed cars and trucks, and rubble in the streets.

A very common example for restricted regions being formed by natural disasters is floods. Many countries around the globe suffer from flooding caused by heavy rainfall (pluvial floods), rising ground water, or coastal floods caused by wave actions and storm surges. Floods have adverse effects on transportation activities due to the formation of large areas of surface water, mudslides and possible debris after the natural disaster, all of which can be regarded as barriers in the context of restricted location problems. Figure 1 shows a flood hazard map of Europe (Alfieri et al., 2014), in which the darker shades show the locations of possible flood occurrences, which, in our context correspond to probabilistic barriers. Recent examples of disruption include that due to flooding in Central Europe in 2002, affecting several countries such as Germany, the Czech Republic, Austria, Italy and Slovakia, causing over 100 fatalities and economic loss (Ulbrich et al., 2003). In 2009, heavy rainfall resulted in the loss of three bridges and the temporary closure of nearly 20 bridges (Cumbria County Council, 2011). In May 2015, flooding due to severe storms in Texas caused at least 27 million dollars in infrastructure damage, including damage to 167 roads in the counties of Texas and 24 causalities (Gallucci, 2015b). Observations from real-life cases show that the locating a facility within a flood zone can be unsafe and costly. Furthermore, transportation of goods or people through them would be difficult after a possible flood.

The stochastic restricted continuous facility location problem is to find the optimal locations of a set of new facilities on the plane to serve a set of demand points, and where there is uncertainty in

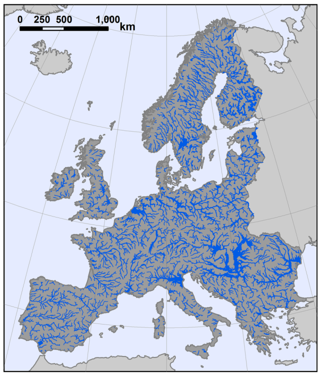

Figure 1: European flood hazard map (Alfieri et al., 2014)

the location of the restricted regions. The objective is to minimize the total cost that is expressed as a function of the distance between the locations of the demand points and locations of the new facilities. We make no assumptions on the shape of the restricted region and the distance metric. The uncertainty around the restricted region locations is characterized by a number of predefined scenarios, each with a given probability. The deterministic variant of the problem where the locations of the regions are known arises as a special case of the problem.

This paper makes four unique contributions which are listed below:

1. We propose the way in which the continuous space can be discretized, allowing for any distance metric to be used in the objective function of the problem.

2. We present a new model for continuous restricted facility location problems, both for stochastic and deterministic cases, that is based on multicommodity flows on the discretized network. The novelty of the model is that the destination node of each commodity is unknown.

3. We describe an exact algorithm that is based on Benders decomposition to optimally solve the proposed model.

4. We report extensive computational results on benchmark instances for the deterministic and the stochastic variants of the problem, and for single and multiple facilities.

The remainder of this paper is organized as follows. A brief overview of the literature on restricted facility location problems is presented in Section 2. In Section 3, we first provide a formal definition of the problem, then present the way in which the continuous network is discretized, and finally describe the multicommodity network flow model with unknown destination. Section 4 describes the Benders decomposition algorithm. The computational results are presented in Section 5. The paper concludes in Section 6.

2 Literature Review

The deterministic restricted facility location problem has been studied to a good extent. We will not present a review of this problem here but instead refer the reader to Oğuz et al. (2016) for a review of single and multiple facility problems with forbidden regions, and single facility problems with barriers. To the best of our knowledge, Bischoff et al. (2009) is the only paper to study the (deterministic) restricted multi-facility location problem with barriers, who describe a genetic algorithm.

In contrast, only a few studies exist on restricted location problems with probabilistic barriers. The first of these papers, to the best of our knowledge, is by Canbolat and Wesolowsky (2010), who consider a barrier in the form of a probabilistic line. The aim is to locate a new facility so as to minimize total expected rectilinear distance between its location and those of a set of demand points. The authors describe an algorithm based on dividing the problem into two, the individual solutions of which are used to construct a solution for the original problem. Canbolat and Wesolowsky (2010) make several strict assumptions, including the uniformity of the barrier location and the invariability of the y-coordinate of the line barrier. A location problem with a probabilistic line barrier that is uniformly distributed on the plane has been studied by Amiri-Aref et al. (2011), where the objective is to minimize the maximum expected rectilinear distance from the demand points to the new facility, and for which the authors describe a heuristic algorithm. The multi-facility version of this problem is studied by Shiripour et al. (2012). For small size instances, the authors use a commercial solver to solve a mixed integer quadratic programming formulation of the problem. For larger size instances, they describe a genetic algorithm and an imperialist competitive algorithm. Amiri-Aref et al. (2013) describe a heuristic for a restricted problem with a probabilistic polyhedral barrier. Finally, a location problem under a multi-period planning horizon using rectilinear distances and a uniformly distributed line barrier is studied by Javadian et al. (2014), who propose a genetic algorithm and an imperialist competitive algorithm.

Stochasticity in facility location has primarily been studied from the point of view of considering uncertainty in customer demands and the ability to meet the demands under limited operational capacity. There exists a rather significant body of work on the facility location problem under uncertainty (Snyder, 2006) which we will not review here for the sake of conciseness, but point out similarities and differences with our work. Compared to existing studies, we take a similar methodological approach in terms of modeling the problem in the form of a two-stage stochastic program and converting it into its deterministic equivalent formulation by using the probabilities of a given set of scenarios as weights in the objective function (Correia and Saldanha da Gama, 2015). The type of uncertainty we consider here, however, is in the formation of restricted regions, rather than customer demands, which breaks away from the general body of work in this area.

3 Modeling with Discretization and Network Flows

This section presents a formal description of the problem studied in this paper, following which two types of network discretizations are presented. This section also describes the new multicommodity flow formulation and the way in which it is applied to the problem using the discretized network.

3.1 Formal Problem Description

Consider a set M of demand points located on a plane, each with a positive weight z_m corresponding to the amount of demand at each $m \in \mathsf{M}$. These demand points need to be served by θ new facilities to be located on the plane. We are also given a finite set of possible locations of the restricted regions on the continuous plane, where the realization of the set of occurrences is modeled as a scenario. Let S be the set of scenarios and p_s be the probability of occurrence of a scenario $s \in \mathsf{S}$. The stochastic restricted continuous facility location problem is to find an optimal placement of the new facilities to serve the demand points. The objective is to minimize a function expressed as the expected value of the weighted total distance between the locations of the demand points and the new facilities, where the expectation is taken over the probabilities defined over the set of scenarios. The deterministic version of the problem is a special case where $|\mathbf{S}| = 1$.

3.2 Construction of the Discrete Network

A straightforward way to discretize a continuous plane is to divide the space into unit squares, where each unit square contains four nodes and six edges. Every pair of nodes are connected with an edge to be able to represent various distance metrics (e.g., Euclidean). An example of this network with four unit squares, nine nodes and 20 edges is given in Figure 2. Using such a discretization, the size of the unit square will decrease with increasing number of vertices, which is expected to refine the approximation.

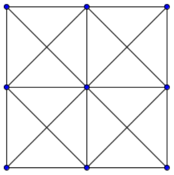

Figure 2: Unit square network

An alternative way to model the discrete network and to refine the approximation is to increase the number of edges by using unit rectangles that are formed by unit squares. The unit rectangles form the columns and rows of the general network, where each node is connected in a cross-cutting manner as shown in Figure 3. In contrast to the discretization shown in Figure 2, the new approach increases the number of diagonal edges on the general network but keeps the number of nodes the same.

Figure 3 shows two unit rectangles, where one of the rectangles represents a column, and the other rectangle forms a row of the discretized network. Each unit rectangle is formed by unit squares. This means that the total number of nodes in unit square network and unit rectangle network is the same. On the other hand, it is clearly seen that the number of edges leaving and entering each vertex has now increased, making more links available as compared to a unit square network with the same number of nodes. Figure 4 presents two examples of networks with same number of nodes, one constructed by a unit square and the other with a unit rectangle network. Even though

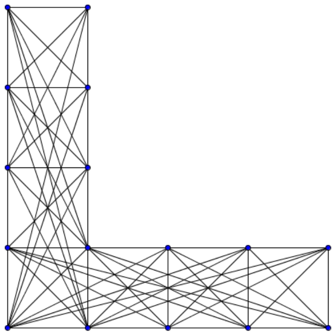

Figure 3: Unit rectangle network

the two networks contain the same number of unit squares and nodes, unit rectangle design offers more edges to connect nodes, which improves the transportation flexibility between nodes.

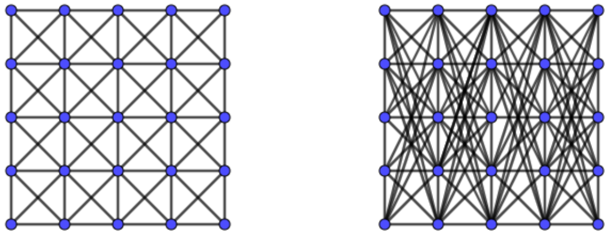

Figure 4: Network Design Comparisons

One point of consideration is regarding the convex hull of the network formed by the locations of the existing facilities and the vertices of the barriers. It is sufficient to construct the discretized network within this convex hull since it is known that the global optimal facility location lies in that region (Butt and Cavalier, 1996). A second point to note is that it is possible to develop a tessellation based on other geometric shapes, such as triangles and hexagons. In this paper, however, the tessellation is limited to unit squares and rectangles given the satisfactory results obtained by the latter. The results of a numerical comparison of quality of the solutions obtained by the two alternative discretizations will be reported in Section 5. To the best of our knowledge, these discretization techniques have not been used previously in restricted facility location problems.

3.3 Multicommodity Network Flows for Restricted Facility Location Problems

The nature of multicommodity network flow problems and facility location problems are, in a sense, similar. Multicommodity network flow problems determine the flows of a set of commodities from their origins to destinations in order to minimize the total cost of routing along the arcs of a given graph (Moradi et al., 2015). In the original statement of the problem, the origin and destination nodes for each commodity are known.

In the model that we propose, the destination nodes of the commodities correspond to the locations of the new facilities, and are unknown. The total number of destination nodes, however, is known and is equal to the total number θ of facilities. Consider, now, the network G formed by the discretization explained in the previous section, defined by a set $\mathbf{N}$ of nodes and a set $\mathbf{A}$ of uncapacitated arcs defined between pairs of nodes as per the discretization. In the stochastic variant of the problem, the set of available arcs is shown by the set $\mathbf{A}_s \subseteq \mathbf{A}$, one for each scenario $s \in \mathbf{S}$. Each vertical or horizontal arc in $\mathbf{A}$ is assumed to have a unit cost, whereas the cost of the diagonal arcs are calculated using standard geometry.

We define a commodity for each demand point. The index set of demand points is denoted by $\mathbf{M}$, which also corresponds to the index set of commodities as there is one-to-one correspondence between the two. In particular, a commodity $m \in \mathbf{M}$ originates from demand point $m \in \mathbf{M}$, which, in graph G, corresponds to a node shown by $O(m)$, where $O(m) \in \mathbf{N}$. The unit flow cost of commodity $m \in \mathbf{M}$ through the directed arc (i, j) in scenario $s \in \mathbf{S}$ is shown by c_{smij}.

With the new notation and terminology, the problem can now be described as sending each commodity $m \in \mathbf{M}$ from its origin $O(m)$ to one of the θ destinations using a least-cost path, where the cost of a path is defined as the total cost of the arcs forming the path. We are not aware of any other work where multicommodity flow problems with unknown destinations have been used, especially for facility location type problems.

3.4 A multicommodity network flow formulation

To formulate the problem, we define three sets of variables. The continuous variable $x_{smij} \geq 0$ is the amount of flow of commodity m through the arc $(i, j) \in \mathbf{A}_s$ in scenario $s \in \mathbf{S}$. The second variable $v_{smi} \geq 0$ is equal to 1 if node $i \in \mathbf{N}$ is designated as the destination of commodity $m \in \mathbf{M}$ in scenario $s \in \mathbf{S}$, and 0 otherwise. This variable is inherently binary, but does not need to be explicitly defined so for the reasons explained below. In a similar way, we define a (binary) variable y_i that takes the value of 1 if a facility is located at node $i \in \mathbf{N}$, and 0 otherwise.

Since there is a finite set of scenarios and the objective is to optimize an expected value, it is possible to form a deterministic equivalent of a stochastic model (DESM) for the problem, as shown below.

$$\text{minimize} \quad \mathrm{E}\left\{F\left(z_m, c_{smij}, x_{smij}\right)_{m \in \mathbf{M}, \{i,j\} \in \mathbf{A}_s}\right\} \tag{3.1}$$

subject to

$$\sum_{j:(i,j) \in \mathbf{A}_s} x_{smji} - \sum_{j:(i,j) \in \mathbf{A}_s} x_{smij} = \begin{cases} -1, & \text{if } i = O(m) \\ v_{smi}, & \text{otherwise} \end{cases} \qquad \forall s \in \mathbf{S}, \forall i \in \mathbf{N}, m \in \mathbf{M} \tag{3.2}$$

$$v_{smi} \leq y_i \qquad \forall s \in \mathbf{S}, \forall i \in \mathbf{N}, \forall m \in \mathbf{M} \tag{3.3}$$

$$\sum_{i \in \mathbf{N}} y_i = \theta \tag{3.4}$$

$$x_{smij} \geq 0 \qquad \forall s \in \mathbf{S}, \forall \{i, j\} \in \mathbf{A}_s, \forall m \in \mathbf{M} \tag{3.5}$$

$$v_{smi} \in \{0, 1\} \qquad \forall s \in \mathbf{S}, \forall i \in \mathbf{N}, \forall m \in \mathbf{M} \tag{3.6}$$

$$y_i \in \{0, 1\} \qquad \forall i \in \mathbf{N}. \tag{3.7}$$

The objective function (3.1) minimizes the expected value of a function defining the total weighted distance between the origin nodes (demand points) and destination nodes (new facilities). For example, the expected value of the objective function of a median problem can be written as follows:

$$\text{minimize} \quad \sum_{s \in \mathsf{S}} \sum_{m \in \mathsf{M}} \sum_{\{i,j\} \in \mathsf{A}_s} z_m p_s c_{smij} x_{smij}. \tag{3.8}$$

Other objective functions can be written in a similar manner. Constraints (3.2) model flow conservation for each commodity from demand point $m \in \mathsf{M}$ in scenario $s \in \mathsf{S}$. Constraints (3.3) state that a commodity $m \in \mathsf{M}$ can only be destined to node $i \in \mathsf{N}$ in scenario $s \in \mathsf{S}$ if $y_i = 1$. Constraints (3.4) set the number of destination nodes, or equivalently, the number of facilities to be located. Constraints (3.5), (3.6) and (3.7) define the non-negativity and binary restrictions on the decision variables, respectively.

We now present a result that allows to relax the integrality of the location variables y_i.

Proposition 1 *Let $DESM_L$ denote model DESM where constraints (3.7) are replaced by $0 \leq y_i \leq 1$ for all $i \in \mathsf{N}$. Then, there always exists an optimal solution (x^L, v^L, y^L) of $DESM_L$ where $y_i^L \in \{0,1\}$ for all $i \in \mathsf{N}$.*

Proof The proof is by contradiction. Assume that $\exists\ i \in \mathsf{N}$ such that $0 < y_i^L < 1$, which, by virtue of constraints (3.4) implies that there exists at least one other node $j \in \mathsf{N} \setminus \{i\}$ such that $0 < y_j^L < 1$. Constraints (3.3) then stipulate that there must exist at least one $m \in \mathsf{M}$ such that $0 < v_{smi}^L < 1$ and $0 < v_{smj}^L < 1$. Let C_{p_1} (resp. C_{p_2}) be the cost of the path between node m and node i (resp., node j). Assume, without loss of generality, that $C_{p_1} \geq C_{p_2}$. There are two cases. If $C_{p_1} = C_{p_2}$, then solution (x^L, v^L, y^L) can be transformed into an alternative solution ($\bar{x}^L$, $\bar{v}^L$, $\bar{y}^L$) with the same cost, where the flows on path p_1 are fully shifted on to path p_2. If, on the other hand, $C_{p_1} > C_{p_2}$, then the cost of the solution ($\bar{x}^L$, $\bar{v}^L$, $\bar{y}^L$) is cheaper than that of (x^L, v^L, y^L), contradicting the assumption. □

Even though the results of Proposition 1 reduces the complexity of the model, its size will increase with the number of demand points and scenarios, which in turn will increase the difficulty of optimally solving $DESM_L$. To overcome this difficulty, we describe an exact algorithm based on Benders decomposition in the next section.

4 Benders Decomposition

Benders decomposition (Benders, 1962) is a technique to efficiently solve large scale formulations by decomposing a model into a subproblem and a master problem, and iterates between the two problems to identify an optimal solution to the original model. Below, we show how standard Benders decomposition can be applied to the DESM$_L$, where the objective function is that of the median problem and is as shown in (3.8).

Let $Y = \{y_i \in \{0,1\}, i \in \mathsf{N}\}$. Fixing the y_i variables as $\bar{Y} = \{\bar{y}_i, i \in \mathsf{N}\}$, such that $\bar{y}_i \in \{0,1\}$ and $\sum_{i \in \mathsf{N}} \bar{y}_i = \theta$ yields the following Benders subproblem,

$$\text{minimize} \quad \sum_{s \in \mathsf{S}} \sum_{m \in \mathsf{M}} \sum_{\{i,j\} \in \mathsf{A}_s} z_m p_s c_{smij} x_{smij} \tag{4.1}$$

subject to

$$\sum_{j \in \mathsf{N}} x_{smji} - \sum_{j \in \mathsf{N}} x_{smij} = \begin{cases} -1, & \text{if } i = O(m), \\ v_{smi}, & \text{otherwise.} \end{cases} \qquad \forall s \in \mathsf{S}, \forall i \in \mathsf{N}, m \in \mathsf{M} \tag{4.2}$$

$$v_{smi} \leq \bar{y}_i \qquad \forall s \in \mathsf{S}, \forall i \in \mathsf{N}, \forall m \in \mathsf{M} \tag{4.3}$$

$$x_{smij} \geq 0 \qquad \forall s \in \mathsf{S}, \forall \{i,j\} \in \mathsf{A}_s, \forall m \in \mathsf{M} \tag{4.4}$$

$$v_{smi} \in \{0,1\} \qquad \forall s \in \mathsf{S}, \forall i \in \mathsf{N}, \forall m \in \mathsf{M}, \tag{4.5}$$

which further decomposes by each scenario $s \in \mathsf{S}$ and each commodity $m \in \mathsf{M}$, as shown in the following by $\mathrm{S}_{sm}(\bar{Y})$:

$$\text{minimize} \quad z_m p_s \sum_{\{i,j\} \in \mathsf{A}_s} c_{smij} x_{smij} \tag{4.6}$$

subject to

$$\sum_{j\in\mathbf{N}} x_{smji} - \sum_{j\in\mathbf{N}} x_{smij} = \begin{cases} -1, & \text{if } i = O(m) \\ v_{smi}, & \text{otherwise.} \end{cases} \qquad \forall i \in \mathbf{N} \tag{4.7}$$

$$v_{smi} \leq \bar{y}_i \qquad \forall i \in \mathbf{N} \tag{4.8}$$

$$x_{smij} \geq 0 \qquad \forall \{i,j\} \in \mathbf{A}_s \tag{4.9}$$

$$v_{smi} \in \{0,1\} \qquad \forall i \in \mathbf{N}. \tag{4.10}$$

Consider, now, the relaxation of $\mathrm{S}_{sm}(\bar{Y})$ where constraints (4.10) are replaced by $0 \leq v_{smi} \leq 1$ for all $i \in \mathbf{N}$, and let A be the matrix of coefficients of constraints (4.7) and (4.8). When $\bar{y}_i$ is fixed to integer values, the subproblem possesses a special structure as shown by the next proposition.

Proposition 2 *The constraint matrix A of $S_{sm}(\bar{Y})$ is totally unimodular.*

Proof Each column of matrix A contains exactly two non-zero entries, where one entry is equal to 1 and the other entry is equal to -1, and where the other entries are equal to zero, therefore satisfying the conditions for a matrix to be totally unimodular (see, e.g., Nemhauser and Wolsey, 1988). □

$\mathrm{S}_{sm}(\bar{Y})$ is a generalization of the well-known shortest path problem, where one seeks to find, for a given $m \in \mathbf{M}$, the path to a node $i \in \mathbf{N}$ with $\bar{y}_i = 1$ that yields the lowest cost, and one which can be solved by a repeated application of the shortest path algorithm (Dijkstra, 1959). Let u_{smi} and w_{smi} be the dual variables associated to constraints (4.7) and (4.8), respectively. The dual $\mathrm{D}_{sm}(\bar{Y})$ of $\mathrm{S}_{sm}(\bar{Y})$ is shown as below:

$$\text{maximize} \quad \sum_{i\in\mathbf{N}} \bar{y}_i w_{smi} - \sum_{\substack{i\in\mathbf{N} \\ i=O(m)}} u_{smi} \tag{4.11}$$

subject to

$$u_{smj} - u_{smi} \leq z_m p_s c_{smij} \qquad \forall \{i,j\} \in \mathbf{A}_s \tag{4.12}$$

$$w_{smi} - u_{smi} \leq 0 \qquad \text{if } i \neq O(m), \forall i \in \mathbf{N} \tag{4.13}$$

$$w_{smi} \leq 0 \qquad \forall i \in \mathbf{N}. \tag{4.14}$$

Once an optimal solution to $\mathrm{S_{sm}}(\bar{Y})$ is obtained, it is possible to compute the values of the $\mathrm{D_{sm}}(\bar{Y})$ as the following proposition shows.

Proposition 3 *Let $(x^\star, v^\star)$ be an optimal solution to the $S_{sm}(\bar{Y})$ where $x^\star = \{x^\star_{smij} | \forall (i,j) \in \mathbf{A}_s\}$ and $v^\star = \{v^\star_{smi} | i \in \mathbf{N}\}$. Then, an optimal solution $(u^\star, w^\star)$ to the $D_{sm}(\bar{Y})$, where $u^\star = \{u^\star_{smi} | \forall i \in \mathbf{N}\}$ and $w^\star = \{w^\star_{smi} | \forall i \in \mathbf{N}\}$, is given by the following:*

$$u^\star_{smi} = 0, \qquad \textit{if } x^\star_{smij} = 1 \textit{ and } v^\star_{smi} = 1 \tag{4.15}$$

$$u^\star_{smj} - u^\star_{smi} = z_m p_s c_{smij}, \qquad \textit{if } x^\star_{smij} = 1 \tag{4.16}$$

$$u^\star_{smj} - u^\star_{smi} \leq z_m p_s c_{smij}, \qquad \textit{if } x^\star_{smij} = 0 \tag{4.17}$$

$$w^\star_{smi} = 0, \qquad \textit{if } \bar{y}_i = 1. \tag{4.18}$$

Proof Equation (4.15) is a fixing of the dual variable $u^\star_{smi}$ that corresponds to the destination of commodity m to 0. Constraints (4.16) and inequalities (4.17) follow from complementary slackness. As for (4.18), we first observe that when $\bar{y}_i = 1$ for a particular node $i \in \mathbf{N}$, then $v^\star_{smi}$ can either be 0 or 1 due to constraint set (3.3). First, consider the case where $v^\star_{smi} = 0$, for a particular node $i \in \mathbf{N}$. In this case, the complementary slackness condition $(v^\star_{smi} - \bar{y}^\star_i)w^\star_{smi} = 0$ implies $w^\star_{smi} = 0$. If, on the other hand, $v^\star_{smi} = 1$, then the complementary slackness condition indicates that $w^\star_{smi} - u^\star_{smi} = 0$, which, together with (4.15) yields $w^\star_{smi} = 0$.□

One implication of Proposition 3 is that the first summation term of (4.11) will be zero at optimality for any fixing $\bar{y}_i$ to either 0 or 1. As for when $\bar{y}_i = 0$ for a given $i \in \mathbf{N}$, we use the tightest values for $w^\star_{smi}$ given by:

$$w^\star_{smi} = \begin{cases} u^\star_{smi}, & \text{if } u^\star_{smi} \leq 0, \\ 0, & \text{otherwise.} \end{cases} \qquad \forall s \in \mathbf{S}, m \in \mathbf{M}. \tag{4.19}$$

Using optimal solutions $(\bar{U}, \bar{W}) = \{(\bar{u}_{smi}, s \in \mathbf{S}, m \in \mathbf{M}, i \in \mathbf{N}), (\bar{w}_{smi}, s \in \mathbf{S}, m \in \mathbf{M}, i \in \mathbf{N})\}$ of the $\mathrm{D_{sm}}(\bar{Y})$, a reformulation of DESM_L, which corresponds to the Benders Master Problem (MP), is given below:

$$\text{minimize} \quad \sum_{s \in \mathsf{S}} \sum_{m \in \mathsf{M}} \alpha_{sm} \tag{4.20}$$

subject to

$$\sum_{i \in \mathsf{N}} y_i \bar{w}^{(k)}_{smi} - \sum_{\substack{i \in \mathsf{N} \\ i = O(m)}} \bar{u}^{(k)}_{smi} \leq \alpha_{sm} \qquad \forall k \in \mathbf{K}_{sm}, \forall s \in \mathbf{S}, \forall m \in \mathbf{M} \tag{4.21}$$

$$\sum_{i \in \mathsf{N}} y_i \bar{w}^{(l)}_{smi} - \sum_{\substack{i \in \mathsf{N} \\ i = O(m)}} \bar{u}^{(l)}_{smi} \leq 0 \qquad \forall l \in \mathbf{L}_{sm}, \forall s \in \mathbf{S}, \forall m \in \mathbf{M} \tag{4.22}$$

$$\sum_{i \in \mathbf{N}} y_i = \theta \tag{4.23}$$

$$0 \leq y_i \leq 1 \qquad \forall i \in \mathbf{N}, \tag{4.24}$$

where α_{sm} is a continuous variable and $\mathbf{K}_{sm}$ and $\mathbf{L}_{sm}$ are the subsets of extreme points and extreme rays of feasible space of the dual problem, respectively, for each $s \in \mathbf{S}$ and $m \in \mathbf{M}$. Constraints (4.21) are written for each extreme point in the feasible region of the dual problem and are known as Benders optimality cuts. Constraints (4.22) are defined for each extreme ray of the master problem whenever it is infeasible and are named Benders feasibility cuts.

We note that it is possible to eliminate Benders feasibility cuts by replacing them with the following constraint in the MP,

$$\sum_{i \in \mathbf{N}_b} y_i = 0 \qquad \forall b \in \mathbf{B}, \tag{4.25}$$

where $\mathbf{B}$ is the set of barriers and $\mathbf{N}_b$ is the subset of nodes that fall within a barrier $b \in \mathbf{B}$. Constraints (4.25) ensure that any solution $\bar{Y}$ will result in feasible $\mathrm{S_{sm}}(\bar{Y})$, which will in turn always render $\mathrm{D_{sm}}(\bar{Y})$ feasible.

The way in which MP is solved is shown in Algorithm 1, which we name BDA. The algorithm which starts with a relaxed MP, shown by MP($\mathbf{K'}$), with a (possibly empty) subset $\mathbf{K'}$ of constraints (4.21), and solves the subproblems to identify violated cuts (4.21), which are then appended to the MP($\mathbf{K'}$). The algorithm iterates in this manner until an optimal solution is identified. At each iteration, the relaxed MP yields a lower bound with a solution $\bar{Y}$, whereas the summation of the objective values of each $\mathrm{D_{sm}}(\bar{Y})$ yields an upper bound to the optimal value of the original model.

Algorithm 1 BDA for Restricted Facility Location Problems

```
$NCut$: Number of Iterations; $z[NCut]$: Optimal value of MP at iteration $NCut$; $Z[NCut]$: Optimal value of the subproblem at iteration $NCut$.
**Initialization:** $NCut \leftarrow 0$, $\mathbf{K'} \leftarrow \emptyset$.
**START**
    Solve MP($\mathbf{K'}$). Let $\bar{Y}$ be an optimal solution and $v_{MP}$ the optimal value.
    $z[NCut] \leftarrow v_{MP}$
    $Z[NCut] \leftarrow \infty$
    **While** $Z[NCut] \geq z[NCut] + \epsilon$
        Solve $\mathrm{D_{sm}}(\bar{Y})$ for all $s \in \mathbf{S}$, $m \in \mathbf{M}$.
        Let $\left(\bar{U}, \bar{W}\right)_{sm}$ be an optimal solution and $v_{\mathrm{D_{sm}}(\bar{Y})}$ the optimal value.
        **If** $(Z[NCut] > \sum_{s \in \mathbf{S}} \sum_{m \in \mathbf{M}} v_{\mathrm{D_{sm}}(\bar{Y})})$ **then** $Z[NCut] = \sum_{s \in \mathbf{S}} \sum_{m \in \mathbf{M}} v_{\mathrm{D_{sm}}(\bar{Y})}$
        Add optimality cuts induced by $\left(\bar{U}, \bar{W}\right)_{sm}$ to set $\mathbf{K'}$
        $NCut \leftarrow NCut + 1$
        Solve MP($\mathbf{K'}$). Let $\bar{Y}$ be an optimal solution and $v_{MP}$ the optimal value.
        $z[NCut] \leftarrow v_{MP}$
    **End While**
**END**
```

There are various ways to improve the performance of the Benders decomposition algorithm. The Pareto-optimal cut approach (Magnanti and Wong, 1981) is a way to generate stronger optimality cuts, which is expected to yield a decrease in the number of iterations to reach optimality, but requires an auxiliary problem to be solved in each iteration. For further details on this approach, see Cordeau et al. (2001), Papadakos (2008) and Papadakos (2009). We have implemented this approach, and denoted the resulting algorithm by PBDA, which will be tested in the following section.

5 Computational Results

In this section, we present results of computational experiments to show the efficiency of the proposed models and algorithms. We first describe how the instances are generated in Section 5.1. The quality of the discretization of the network is tested in Section 5.2. Sections 5.3 and 5.4 report the computational results for the deterministic and the stochastic restricted multi-facility location problem instances, respectively. Finally, we investigate the value of the stochastic solutions in

Section 5.5.

All experiments in Section 5.2 were run on a PC with an Intel(R) Core(TM) 2.60 GHz processor and 4.00 GB of RAM using CPLEX 12.5. Given the size of the instances solved in Sections 5.3 and 5.4, it was necessary to use the computer cluster IRIDIS 4 at the University of Southampton with dual 2.6 GHz Intel Sandybridge processors and 64 GB of RAM using CPLEX 12.5. Both BDA and PBDA were implemented in C++.

5.1 Description of the Instances

There are, to the best of our knowledge, only three restricted facility location problem instances with barriers and four instances with forbidden regions that have been described in other studies. Tables 1 and 2 show the characteristics of these instances including the instance name, the source, the type, the number of demand points ($|\mathsf{M}|$), the type of restricted regions (RRT), and the distance metric used.

Table 1: Barrier instances

Name	Source	Type	$\lvert\mathsf{M}\rvert$	RRT	Distance
B1	Katz and Cooper (1981)	1-median	5	Circular	Euclidean
B2	Katz and Cooper (1981)	1-median	10	Circular	Euclidean
B3	Aneja and Parlar (1994)	1-median	18	Polygonal	Euclidean

Table 2: Forbidden region instances

Name	Source	Type	$\lvert\mathsf{M}\rvert$	RRT	Distance
FR1	Aneja and Parlar (1994)	1-median	4	Polygonal	Euclidean
FR2	Hamacher and Nickel (1994)	1-median	3	Polygonal	Euclidean
FR3	Hamacher and Nickel (1994)	1-median	3	Polygonal	Rectilinear
FR4	Hamacher and Nickel (1994)	1-median	3	Polygonal	Chebyshev

The instances shown in Tables 1 and 2 all involve location of a single facility, which we use here to evaluate the quality of the discretization presented in Section 3.3.

As for multi-facility instances, we have generated both deterministic and stochastic instances based on a complex barrier instance originally described by Aneja and Parlar (1994). The original instance has 12 polygonal barriers and 18 demand points, which are located at points with coordinates $(1, 2)$, $(6, 1)$, $(9, 1)$, $(14, 2)$, $(5, 5)$, $(7, 4)$, $(9, 5)$, $(14, 4)$, $(17, 4)$, $(2, 8)$, $(8, 8)$, $(16, 8)$, $(3, 12)$, $(6, 11)$, $(9, 10)$,

$(17, 10)$, $(10, 12)$ and $(19, 13)$. The distance metric is Euclidean. The instance is shown in Figure 5, where each barrier is identified by a number and the demand points are shown at locations e1–e18.

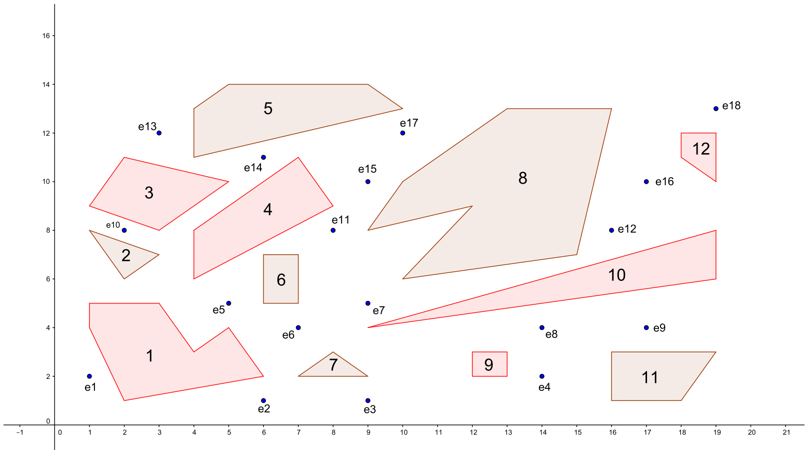

Figure 5: Aneja and Parlar (1994) barrier instance

New and larger deterministic instances were generated by considering 36, 54 and 72 demand points. For the 36 demand points case, an additional 18 points were located at coordinates $(19, 2)$, $(2, 14)$, $(4, 9)$, $(5, 11)$, $(7, 11)$, $(8, 6)$, $(9, 12)$, $(10, 8)$, $(12, 1)$, $(12, 13)$, $(13, 14)$, $(15, 7)$, $(16, 4)$, $(17, 5)$, $(18, 1)$, $(19, 5)$, $(19, 9)$ and $(19, 10)$. For the 54 demand points case, a further 18 additional points were added to the instance with 36 demand points, located at coordinates $(1, 6)$, $(1, 10)$, $(3, 6)$, $(3, 1)$, $(4, 4)$, $(4, 14)$, $(6, 3)$, $(7, 8)$, $(8, 12)$, $(9, 7)$, $(10, 4)$, $(11, 8)$, $(13, 6)$, $(14, 14)$, $(15, 1)$, $(16, 12)$, $(17, 5)$ and $(5, 6)$. Finally, the 72 demand point instance was constructed by adding a further 18 demand points to the instance with 54 demand points, located at coordinates $(1, 13.5)$, $(3, 11)$, $(3, 14)$, $(4, 1)$, $(5, 10)$, $(9, 8)$, $(9, 14)$, $(10, 3)$, $(10, 11)$, $(11, 14)$, $(12, 6)$, $(13, 13)$, $(14, 6)$, $(15, 14)$, $(17, 14)$, $(19, 1)$, $(10, 12)$ and $(19, 5)$.

The stochastic instances were generated with respect to different parameters, shown in Table 3 including the number of scenarios (NS), the corresponding barriers with reference to the numbers shown in Figure 5, and three sets of probabilities. For example, in the first scenario with NS $= 2$, the barriers numbered 1, 3, 5, 7, 9 and 11 occur with 50% probability in Set 1, with 60% probability in Set 2, and with 80% probability in Set 3. In the second scenario with NS $= 2$, the barriers numbered 2, 4, 6, 8, 10 and 12 in Figure 5 occur with 50%, 40% and 20% probabilities in Sets 1, 2 and 3, respectively.

Table 3: Probability sets for the stochastic restricted facility location instances

NS	Scenarios(s)	Barriers	p_s (Set 1)	p_s (Set 2)	p_s (Set 3)
2	1	1-3-5-7-9-11	0.5	0.6	0.8
	2	2-4-6-8-10-12	0.5	0.4	0.2
3	1	1-4-7-10	0.333	0.6	0.4
	2	2-5-8-11	0.333	0.2	0.3
	3	3-6-9-12	0.333	0.2	0.3
4	1	1-5-9	0.25	0.3	0.4
	2	2-6-10	0.25	0.3	0.1
	3	3-7-11	0.25	0.2	0.4
	4	4-8-12	0.25	0.2	0.1
5	1	1-6-11	0.2	0.3	0.4
	2	2-7-12	0.2	0.2	0.2
	3	3-8	0.2	0.1	0.1
	4	4-9	0.2	0.3	0.1
	5	5-10	0.2	0.1	0.2
6	1	1-7	0.1667	0.2	0.4
	2	2-8	0.1667	0.1	0.2
	3	3-9	0.1667	0.2	0.1
	4	4-10	0.1667	0.1	0.1
	5	5-11	0.1667	0.3	0.1
	6	6-12	0.1667	0.1	0.1

The use of three probability sets, four instance groups (with 18, 36, 54, 72 demand points), six values for $\theta = 1, ..., 6$, and five different combinations of scenarios give rise to a total of $3 \times 4 \times 6 \times 5 = 360$ stochastic problem instances.

5.2 Quality of the Discretization

In this section, we assess the quality of the two types of discretization, namely the unit square and the unit rectangle network described in Section 3.2, by solving the deterministic instances shown in Tables 1 and 2. Optimal or near-optimal solutions for the continuous problems were obtained using the models proposed by Oğuz et al. (2016). The solutions obtained from the discretization and the original solutions are compared by comparing the relative gap between them.

5.2.1 Unit Square Network

We initially consider instance B1 and solve it using different number of discretization points. The best-known objective value for this instance is 48.257 and is reported in Butt and Cavalier (1996); Klamroth (2001); Bischoff and Klamroth (2007); Klamroth (2004) and Oğuz et al. (2016), which we use to assess the approximation quality of the unit square network. Table 4 presents the number of nodes (NN) used in the unit square network and the percentage gap (PG) between the best-known objective value of the continuous problem (OVCP) and the optimal objective value of the discretized problem (OVDP), calculated as PG = 100 × (OVDP-OVCP)/OVCP.

Table 4: B1 results with unit square network

NN	PG
400	6.76%
1,600	5.74%
6,400	5.68%
25,600	5.59%
102,400	5%

Table 4 shows that increasing the number of nodes improves the approximation. However, even with $102,400$ nodes on the discretized network, the percentage gap remains at 5%, which is unsatisfactory for the purposes of an exact algorithm. The results suggest that the unit square network discretization is inefficient in producing near-optimal approximations.

We believe the main reason for the inefficiency of the unit square network is due to the limited number of links leaving a node. In particular, a flow leaving a node can only travel through three possible links in this case.

5.2.2 Unit Rectangle Network

To test the quality of the unit rectangle network, all instances shown in Tables 1 and 2 are discretized and solved using this network. All the instances described in Table 1 were constructed with 400 and 1600 nodes in the resulting graph, with the exception of one instance, FR1, necessitated by the size of this instance.

Tables 5 presents the results obtained for the barrier instances, where the best known values (OVCP) are from Bischoff and Klamroth (2007) and Oğuz et al. (2016) for instance B2 and from Aneja and Parlar (1994), Bischoff and Klamroth (2007) and Oğuz et al. (2016) for instance B3.

Comparing the new results obtained for instance B1 with those in Table 4, a fewer number of nodes is needed to achieve much better percentage gaps as compared with the 5%. In particular, the PG is only 1.2% with 400 discretization points, and is further reduced to 0.67% when 1600 discretization points are used. Similar results are obtained for instances B2 and B3.

Table 5: Barrier instances results comparison

Name	NN	OVCP	OVDP	PG
B1	400	48.257	48.817	1.2%
B1	1600	48.257	48.581	0.67%
B2	400	88.326	90.188	2.1%
B2	1600	88.326	89.218	0.99%
B3	400	119.139	120.573	1.2%
B3	1600	119.139	119.53	0.33%

The best known solutions for the forbidden region instances are from Aneja and Parlar (1994) and Oğuz et al. (2016) for instance FR1 and Hamacher and Nickel (1994) and Oğuz et al. (2016) for instances FR2–FR4. Table 6 shows the comparison results.

Table 6: Forbidden region instances results comparison

Name	NN	OVCP	OVDP	PG
FR1	484	48.5	48.738	0.49%
FR1	1936	48.5	48.655	0.32%
FR2	400	8.566	8.595	0.33%
FR2	1600	8.566	8.594	0.32%
FR3	400	10	10	0%
FR3	1600	10	10	0%
FR4	400	8	8	0%
FR4	1600	8	8	0%

Both Tables 5 and 6 show that all instances in Tables 1 and 2 solved using the unit rectangle network yield satisfactory gaps, all within 1%. The construction of the unit rectangle network leads to better results as compared to the unit square network as more links are made available in the former. For this reason, all results presented in the following section will use the unit rectangle network with 1600 nodes.

5.3 Deterministic Restricted Multi-facility Location Problem Instances

The deterministic instances arise as a special case of the stochastic restricted facility location problem in which there is a single scenario. A total of 24 instances were generated and solved for the deterministic case. To solve the resulting model, we used three different methods, namely DESM$_L$ solved by CPLEX 12.5, BDA and PBDA. The results are shown in Table 7, which shows, for each instance, the total number $|\mathbf{M}|$ of demand points, the number θ of new facilities, the optimal objective value $v^\star$, the total CPU time t_C (in seconds) spent by CPLEX 12.5 to solve the formulation, the total CPU time t_D (in seconds) required by BDA, the average CPU time t_M (in seconds) required to solve the master problem at each iteration of BDA, the total number η_i of BDA iterations, the total CPU time t_P (in seconds) needed to solve each instance by PBDA, and the total number η_p of PBDA iterations.

Table 7: Results for the deterministic restricted multi-facility location instances

$\vert\mathsf{M}\vert$	θ	$v^\star$	t_C	t_D	t_M	η_i	t_P	η_p
18	1	119.53	29.02	1.21	0.008	3	296.43	2
	2	90.68	33.35	2.01	0.008	5	372.89	4
	3	66.38	30.91	2.37	0.008	6	366.93	5
	4	50.22	29.08	2.57	0.004	6	559.1	5
	5	42.64	29.06	3.74	0.005	9	427.2	8
	6	36.56	28.42	4.04	0.004	9	370.18	6
36	1	270.69	66.33	3.21	0.013	4	697.49	2
	2	176.51	63.73	5.45	0.012	7	1272.07	6
	3	139.9	63.58	7.35	0.014	7	966.63	7
	4	111.91	60.11	8.13	0.011	9	937.38	6
	5	95.53	58.48	6.91	0.011	8	952.4	6
	6	86.06	57.74	18.86	0.012	17	1212.22	10
54	1	388.48	94.28	3.43	0.015	3	1257.49	2
	2	279.05	107.71	12.55	0.014	8	2650.29	6
	3	222.12	99.37	12.68	0.013	8	2525.77	9
	4	181.24	90.05	12.53	0.016	11	1653.59	7
	5	159.52	89.98	16.85	0.017	13	2965.54	13
	6	141.09	90.61	16.87	0.018	11	1890.73	9
72	1	538.08	140.56	4.65	0.02	3	2153.84	2
	2	395.52	153.47	9.21	0.02	6	2801.23	5
	3	314.2	151.79	17.75	0.019	9	2990.04	7
	4	256.98	128.24	13.52	0.017	8	3521.19	8
	5	224.57	133.44	22.68	0.018	14	LIMIT	0
	6	196.95	119.61	20.33	0.014	9	LIMIT	0

The results shown in Table 7 indicate a clear dominance of BDA over both CPLEX 12.5 and PBDA in terms of the solution time.

Table 8 presents average CPU time statistics for solving instances under columns $\hat{t}_C$, $\hat{t}_D$, and $\hat{t}_P$ for CPLEX 12.5, BDA and PBDA, respectively, the average savings (in percent) between CPLEX 12.5 and BDA (SAVC), and between PBDA and BDA (SAVP), as well as the average number $\hat{\eta}_i$ of iterations in BDA and the average number $\hat{\eta}_p$ of iterations in PBDA.

Table 8: Average savings gained by BDA over CPLEX 12.5 and PBDA

$\vert\mathsf{M}\vert$	$\hat{t}_C$	$\hat{t}_D$	$\hat{t}_P$	SAVC	SAVP	$\hat{\eta}_i$	$\hat{\eta}_p$
18	29.97	2.66	398.79	91.14%	99.33%	6.333	5
36	61.83	8.32	1006.37	86.55%	99.17%	8.667	6.167
54	95.33	12.49	2157.24	86.90%	99.42%	9	7.167
72	137.85	14.69	–	89.34%	–	8.167	–

The results in Table 8 show that the average solution CPU times for instances obtained by BDA are significantly less than those for CPLEX 12.5 and PBDA. The average reduction in computation time by BDA are more than 85% and 99% over CPLEX 12.5 and PBDA, respectively. These results confirm that the performance of the BDA is superior to the other two techniques. Another interesting point in Table 8 is that the stronger cuts generated by PBDA require fewer iterations as compared to the BDA, however this comes at the expense of a significant increase in the computational time as the results indicate. The instances with 72 demand points are not included in the columns related to PBDA, since those with five or six new facilities could not be solved by PBDA due to the limitations posed by the available computer memory.

Table 9 shows summary statistics for all instances, namely the average time $\hat{t}_M$ to solve the MP, the average total time $\hat{T}_M$ to solve the MP within the BDA, and the ratio of the total time required to solve the MP as a fraction of the total solution time.

Table 9: MP average solution times

$\lvert\mathbf{M}\rvert$	$\hat{t}_M$	$\hat{T}_M$	Ratio
18	0.006	0.038	1.42%
36	0.012	0.106	1.28%
54	0.015	0.139	1.11%
72	0.018	0.147	1%

As can be seen in Table 9, solving the relaxed MP in BDA consumes significantly less time compared to the other steps of the algorithm.

5.4 Stochastic Restricted Location Problem Instances

In this section, we report the computational results for the 360 stochastic restricted location problem instances with up to six facilities using BDA. Detailed computational results for the whole set of instances are presented in the Appendix as supplementary material.

For illustrative purposes, we depict the solution of an instance with 36 demand points, six new facilities and two scenarios using probability Set 1. Figure 6a shows the locations of the new facilities and the paths from demand points to these facilities when the barriers in scenario 1 occur, whereas Figure 6b shows the resulting solution under scenario 2.

A summary of the results for 18 and 36 demand point instances is presented in Tables 10 and 11,

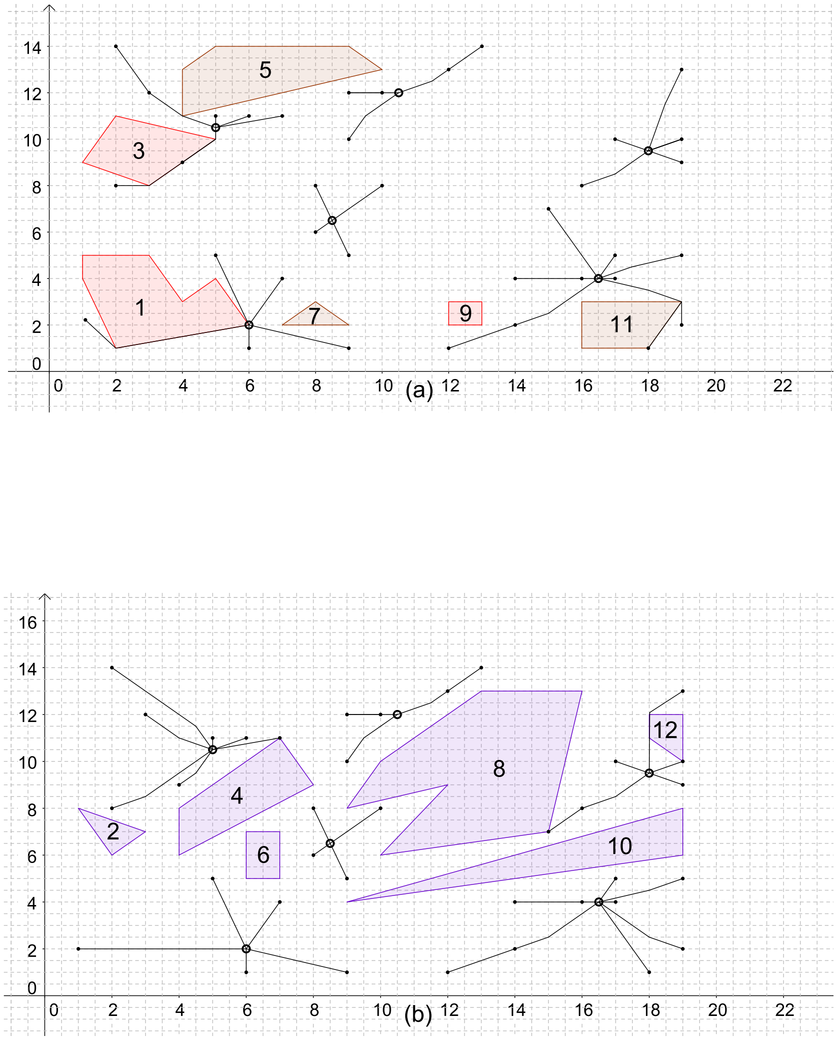

Figure 6: Solutions for the instance with 36 demand points and 6 new facilities using probability set 1

respectively, with results averaged across the scenarios for each probability set. The Ratio column presents the time spent to solve the MP in the algorithm as a fraction of the overall solution time.

Table 10: Summary of results for 18 demand points instances

	θ	$\hat{v^\star}$	$\hat{t}_C$	$\hat{t}_D$	SAVC	$\hat{t}_M$	$\hat{\eta}_i$	Total $\hat{t}_M$	Ratio
Set 1	1	113.21	181.1	5.27	97.09%	0.026	2.8	0.073	1.38%
	2	83.18	212.06	11.41	94.62%	0.024	6.4	0.15	1.32%
	3	62.76	194.11	14.53	92.52%	0.021	7.8	0.166	1.14%
	4	48.68	169.69	11.40	93.28%	0.021	6	0.127	1.12%
	5	41.36	164	16.24	90.1%	0.047	8.6	0.401	2.47%
	6	35.56	153.01	19.78	87.07%	0.023	9.4	0.217	1.09%
Set 2	1	113.5	184.51	5.55	96.99%	0.022	3	0.066	1.19%
	2	83.58	219.89	11.25	94.88%	0.022	6.4	0.14	1.25%
	3	62.91	191.94	13.48	92.98%	0.023	7.4	0.168	1.25%
	4	48.93	166	16.16	90.26%	0.021	9	0.19	1.18%
	5	41.48	162.88	15.7	90.36%	0.02	8.6	0.17	1.08%
	6	35.62	154.6	23.63	84.71%	0.021	11.2	0.238	1.01%
Set 3	1	112.95	189.06	5.55	97.07%	0.023	3	0.069	1.24%
	2	83.48	222.52	10.83	95.13%	0.022	6.2	0.134	1.24%
	3	62.40	194.3	13.12	93.25%	0.028	7.4	0.208	1.59%
	4	48.92	165.83	13.93	91.60%	0.021	7.4	0.155	1.11%
	5	41.57	162.95	16.04	90.16%	0.021	8.8	0.186	1.16%
	6	35.71	154.63	27.18	82.42%	0.02	13.6	0.276	1.02%

Table 11: Summary of results for 36 demand points instances

	θ	$\hat{v^\star}$	$\hat{t}_C$	$\hat{t}_D$	SAVC	$\hat{t}_M$	$\hat{\eta}_i$	Total $\hat{t}_M$	Ratio
Set 1	1	249.14	471.17	10.74	97.72%	0.038	3	0.114	1.06%
	2	161.93	534.27	22.38	95.81%	0.043	6.4	0.275	1.23%
	3	130.57	478.45	36.66	92.34%	0.041	10.2	0.416	1.13%
	4	107.12	430.67	31.67	92.65%	0.039	8.6	0.335	1.06%
	5	91.42	364.85	28.2	92.27%	0.038	7.6	0.289	1.03%
	6	83.136	362	50.69	86%	0.04	14.4	0.569	1.12%
Set 2	1	249.53	475.25	11.04	97.68%	0.042	3	0.126	1.14%
	2	162.56	502.19	19.38	96.14%	0.04	5.8	0.233	1.2%
	3	130.97	478.98	26.16	94.54%	0.04	7.2	0.29	1.11%
	4	107.5	423.11	31.93	92.45%	0.039	8.6	0.334	1.05%
	5	91.66	356.95	26.53	92.57%	0.039	7.6	0.293	1.11%
	6	83.38	364.03	47.55	86.94%	0.04	10.8	0.427	0.9%
Set 3	1	246.62	451.97	10.54	97.67%	0.04	3	0.12	1.14%
	2	160.54	495.16	18.46	96.27%	0.04	5.4	0.217	1.18%
	3	130.03	473.4	28.94	93.89%	0.041	7.8	0.319	1.1%
	4	107.71	381.33	31.72	91.68%	0.041	8.8	0.359	1.13%
	5	91.9	362.38	53.45	85.25%	0.043	11.8	0.511	0.96%
	6	83.58	370.38	100.14	72.96%	0.046	23	1.066	1.06%

The results shown here confirm once again that BDA is superior to CPLEX 12.5 for the stochastic instances in terms of computational time. The time reduction by BDA for 18 and 36 demand points instances on probability sets 1, 2 and 3 for up to six new facilities are presented in Figures 7a and 7b, respectively. The highest levels of savings in computational time are in single facility problem instances, varying between 96.99% and 97.72%. When the number of new facilities increases, the performance gap between BDA and CPLEX 12.5 decreases, but the BDA is still faster by at least 72.96%, which is attained for the instances with 36 demand points, six new facilities, and under probability set 3.

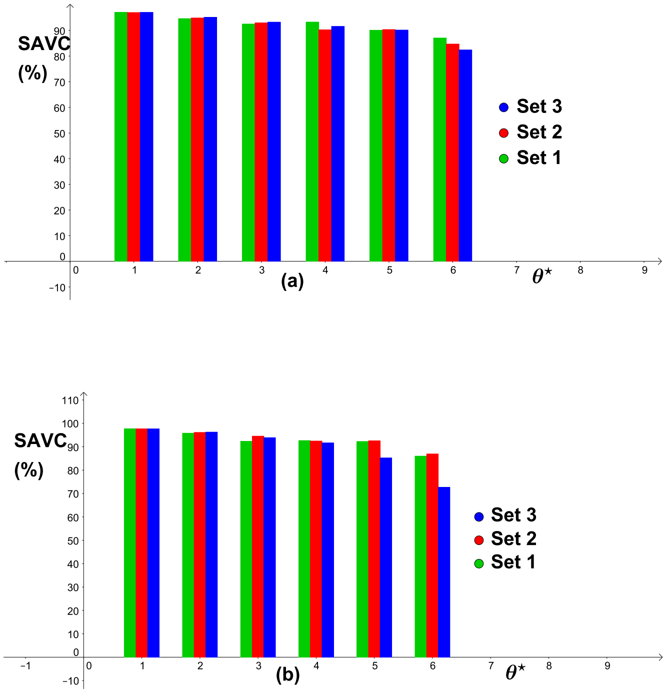

Figure 7: Savings (%) vs. number of demand points for 18 and 36 demand points instances

To solve the instances with 54 and 72 demand points using CPLEX 12.5 a large amount of memory is needed, for which reason RAM was increased to 256 GB. Table 12 shows the total number NV of variables and the total number NC of constraints of DESM$_L$ for 54 and 72 demand points instances for each scenario.

Table 12: Total number of variables and constraints for 54 and 72 demand points Instances

$\|\mathbf{M}\|$	NS	NV	NC
54	2	276,654,400	347,201
	3	414,980,800	520,001
	4	553,307,200	692,801
	5	691,633,600	865,601
	6	829,960,000	1,038,401
72	2	368,872,000	462,401
	3	553,307,200	692,801
	4	737,742,400	923,201
	5	922,177,600	1,153,601
	6	1,106,612,800	1,384,001

Using CPLEX 12.5, only 39 of the 90 instances with 54 demand points could be solved, but it was unable to solve any of the instances with 72 demand points. BDA optimally solved all 54 and 72 demand points instances even with no increase in RAM.

Table 13: Summary of results for 54 demand points instances

	θ	$\hat{v^\star}$	$\hat{t}_D$	$\hat{t}_M$	$\hat{\eta}_i$	Total $\hat{t}_M$	Ratio
Set 1	1	359.95	23.28	0.063	3.8	0.238	1.02%
	2	253.15	31.02	0.059	6.2	0.366	1.18%
	3	204	46.32	0.059	8.4	0.497	1.07%
	4	171.6	57.34	0.061	10.2	0.627	1.09%
	5	150.04	159.23	0.079	19.8	1.555	0.98%
	6	133.61	67.30	0.056	11	0.614	0.91%
Set 2	1	360.25	17.7	0.057	3.2	0.183	1.04%
	2	254.34	34.15	0.06	6.6	0.397	1.16%
	3	204.77	49.04	0.06	8.8	0.526	1.07%
	4	172.24	49.66	0.057	9	0.514	1.04%
	5	150.74	99.07	0.061	15.6	0.952	0.96%
	6	134.42	53.17	0.059	9.6	0.565	1.06%
Set 3	1	357.52	21.1	0.059	3.6	0.213	1.01%
	2	250.78	34.74	0.064	6.6	0.419	1.21%
	3	203.47	82.04	0.067	13.6	0.917	1.12%
	4	172.28	66.75	0.062	12.2	0.752	1.13%
	5	151.15	136.84	0.069	18.4	1.278	0.93%
	6	134.96	60.55	0.059	10.2	0.597	0.99%

Table 14: Summary of results for 72 demand points instances

	θ	$\hat{v^\star}$	$\hat{t}_D$	$\hat{t}_M$	$\hat{\eta}_i$	Total $\hat{t}_M$	Ratio
Set 1	1	493.31	21.60	0.076	3	0.228	1.06%
	2	363.17	59.97	0.087	8.2	0.716	1.19%
	3	290.89	90.96	0.085	11.4	0.97	1.07%
	4	239.54	71.55	0.076	10.2	0.771	1.08%
	5	208.37	176.24	0.093	17.2	1.592	0.9%
	6	185.45	104.43	0.076	12	0.908	0.87%
Set 2	1	493.63	36.39	0.083	4.6	0.382	1.05%
	2	364.53	62.17	0.088	8.8	0.778	1.25%
	3	291.99	87.89	0.083	11.2	0.926	1.05%
	4	240.21	133.76	0.081	10.8	0.875	0.65%
	5	209.26	146.14	0.084	15.6	1.308	0.89%
	6	186.57	102.81	0.078	11.4	0.889	0.87%
Set 3	1	489.48	21.53	0.076	3	0.228	1.06%
	2	360.60	70.81	0.086	9.4	0.804	1.14%
	3	290.47	394.71	0.145	37.2	5.395	1.37%
	4	240.01	79.76	0.075	10	0.753	0.94%
	5	209.20	132.18	0.08	14.8	1.187	0.9%
	6	187.17	148.40	0.081	15.2	1.232	0.83%

As in the deterministic case, Tables 10–11 and 13–14 show that the solution times of each master problem in each iteration is significantly smaller. The ratios between average total time for solving MPs and average solution time for BDA varies from 0.65% to 2.47% with an overall average of 1.11%. This shows that decomposing the model and solving each $\mathrm{D_{sm}}(\bar{Y})$ with a shortest path algorithm leads to a significant gain in performance in terms of computational time. Large sized models cannot be solved by CPLEX 12.5 to optimality even with 256 GB of memory. These results along with those in the previous section indicate that BDA is an efficient algorithm to solve both deterministic and stochastic restricted location problems that have been discretized using the proposed modeling framework.

5.5 The Value of the Stochastic Solution

One question of concern is the value of solving the stochastic restricted facility location problem over solving the deterministic counterpart, if all the random variables were replaced by their expected values. The deterministic counterpart would be computationally less difficult to solve, compared to stochastic version, and would lead to implementable first stage results, which, in our case, are the decisions on the locations of the new facilities. This problem is called expected value problem or mean value problem (EV) (Birge and Louveaux, 1997).

Birge and Louveaux (1997) introduced the concept of the value of the stochastic solution (VSS), which aims to measure the value of solving stochastic problem by comparing the result obtained from stochastic problem (RP) and the expected result of using the EV solution (EEV). VSS is computed as follows,

$$\text{VSS} = v^{\star}_{EEV} - v^{\star}_{RP}, \tag{5.1}$$

where $v^{\star}_{RP}$ is the value of the stochastic problem. To overcome the dependency of VSS on the parameters of a given problem instance, we instead use the relative value of the stochastic solution computed as:

$$\text{PVSS} = 100 \times \frac{v^{\star}_{EEV} - v^{\star}_{RP}}{v^{\star}_{EEV}}. \tag{5.2}$$

The $v^{\star}_{EEV}$ and PVSS values of all 360 stochastic instances are presented in the Appendix, according to which 95% of the instances have a positive PVSS, whereas only 18 instances have a PVSS equal to zero.

A summary of these results across all the different probability sets from 1 to 3 is presented in Table 15, which shows the minimum, maximum and average PVSS. For example, in instances with

$|\mathsf{M}| = 18$ and the probability set is 1, the percentage minimum saving is 0.11%, the percentage maximum saving is 3.89% and the percentage average saving is 1.53%.

Table 15: Savings of the stochastic instances

	PVSS								
	Probability Set 1			Probability Set 2			Probability Set 3		
$\lvert\mathsf{M}\rvert$	Min.	Max.	Avg.	Min.	Max.	Avg.	Min.	Max.	Avg.
18	0.11	3.89	1.53	0.18	4.80	1.59	0.20	4.73	1.81
36	0.00	2.50	0.88	0.00	3.24	0.95	0.00	3.21	1.17
54	0.00	3.78	1.43	0.00	4.07	1.42	0.00	4.07	1.66
72	0.49	4.67	2.05	0.42	4.77	2.00	0.66	4.36	2.17

Considering the number of demand points and the probability sets, Table 15 indicates a consistency in the results for all stochastic instances. For all probability sets, the least minimum, maximum and average PVSS are obtained for instances with 36 demand points. In contrast, the highest values are seen for instances with 72 demand points.

Even if the PVSS seems rather low at a first glance, they might lead to significant savings in terms of the cost values of the problem. Table 16 shows, as an example, the PVSS for two instances from probability set 2 with 18 and 72 demand points.

Table 16: PVVs and VSSs of two chosen instances

$\lvert\mathsf{M}\rvert$	θ	NS	PVSS	VSS (distance)
18	2	5	4.80%	4.06
72	6	6	4.77%	10.29

In particular, the VSS of the instance with 72 demand points is more than double the VSS of the instance with 18 demand points, even though they have a similar PVSS. It is also obvious that an increase in the unit flow costs will lead to an increase in VSS, but not in PVSS. From a managerial perspective, even if the PVSSs are not large, the VSS can still be significant.

6 Conclusions

This paper described a model and a solution algorithm for stochastic and deterministic restricted facility location problems originally defined on the plane. The model is of a multicommodity network flow formulation with unknown destinations constructed on a discretization of the continuous plane, and provides an approximate representation of the original problem. Computational results on the instances in the literature indicated that the percentage gaps between the best-known optimal solutions and the approximate results were less than 1% for all instances, which showed that the quality of the network approximation was satisfactory in solving the original problem.

The algorithm itself is based on Benders Decomposition and provides an exact solution for the model. It exploits the structure of the discrete network by decomposing the subproblem over the number of demand points and the scenarios. Each subproblem can be solved through the use of a shortest path algorithm. The algorithm proved effective in solving the proposed model, yielding significant time reductions over an off-the-shelf solver for over a total 384 instances. The performance of the latter dropped significantly with an increase in the number of demand points. In particular, 56.67% of the instances with 54 demand points and none of the instances with 72 demand points could be solved by the off-the-shelf solver, whereas all of these instances could be solved by the proposed solution technique within at most a few hundred seconds.

In this paper, we have tested the model and the algorithm in solving both stochastic and deterministic restricted facility location problems with forbidden regions and barriers, with single and multiple facilities, and with various distance metrics. One extension of this work would be to investigate the use of the approach taken here to solve problems arising in emergency logistics using risk-averse models (e.g., minimax-with-regret or conditional value-at-risk as in Alem et al., 2016).

Acknowledgments The authors thank two anonymous reviewers whose comments have been helpful in improving an earlier version of the paper.

References

Alem D, Clark A, Moreno A (2016) Stochastic network models for logistics planning in disaster relief. *European Journal of Operational Research* 255(1):187–206.

Alfieri L, Salamon P, Bianchi A, Neal J, Bates P, Feyen L (2014) Advances in pan-European flood hazard mapping. *Hydrological Processes* 28(1):4067–4077.

Amiri-Aref M, Javadian N, Tavakkoli-Moghaddam R, Aryanezhad MB (2011) The center location problem with equal weights in the presence of a probabilistic line barrier. *International Journal of Industrial Engineering Computations* 2(1):793–800.

Amiri-Aref M, Javadian N, Tavakkoli-Moghaddam R, Baboli A (2013) A new mathematical model for the Weber location problem with a probabilistic polyhedral barrier. *International Journal of Production Research* 51(20):6110–6128.

Aneja YP, Parlar M (1994) Technical note algorithms for weber facility location in the presence of forbidden regions and/or barriers to travel. *Transportation Science* 28(1):70–76.

Balakrishnan A, Magnanti TL, Wong RT (1987) A dual-ascent procedure large-shape uncapacitated network design. Working paper, OR 161-87, Massachusetts Institute of Technology, Cambridge.

Bazaraa MS, Jarvis JJ, Sherali HD (1990) *Linear programming and network flows*. John Wiley & Sons, Canada.

Benders J (1962) Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik* 4(3):238–252.

Birge JR, Louveaux F (2007) *Introduction to Stochastic Programming*. Springer, New York.

Bischoff M, Klamroth K (2007) An efficient solution method for weber problems with barriers based on genetic algorithms. *European Journal of Operational Research* 177(1):22–41.

Bischoff M, Fleischmann T, Klamroth K (2009) The multi-facility location-allocation problem with polyhedral barriers. *Computers & Operations Research* 36(1):1376–1392.

Butt SE, Cavalier TM (1996) An efficient algorithm for facility location in the presence of forbidden regions. *European Journal of Operational Research* 90(1):56–70.

Canbolat MS, Wesolowsky GO (2010) The rectilinear distance weber problem in the presence of a probabilistic line barrier. *European Journal of Operational Research* 202(1):114–121.

Carrizosa E, Plastria F (1993) Planar minquantile and maxcovering location problems *Studies in Locational Analysis* 4:29–33.

Cordeau JF, Stojkovic G, Soumis F, Desrosiers J (2001) Benders decomposition for simultaneous aircraft routing and crew scheduling. *Transportation Science* 35(4):375–388.

Correia I, Saldanha da Gama F (2015) Facility location under uncertainty. In: G. Laporte, S. Nickel and F. Saldanha da Gama (eds.) Location Science, Springer, Cham.

Cumbria County Council (2011) Cumbria floods november 2009 learning from experience recovery phase debrief report . ⟨http://www.cumbria.gov.uk/eLibrary/Content/Internet/533/561/4066711109.pdf⟩ (accessed 15.03.2017).

Dijkstra EW (1959) A note on two problems in connexion with graphs. *Numerische Mathematik* 1:269–271.

Foulds LR, Hamacher HW (1993) Optimal bin location and sequencing in printed circuit board assembly . *European Journal of Operational Research* 66:279–290.

Gallucci M (2015a) Texas And Oklahoma Floods 2015: How Global Warming Makes The Flooding More Destructive. *International Business Times* 26 May. Available from: ⟨http://www.ibtimes.com/texas-oklahoma-floods-2015-how-global-warming-makes-flooding-more-destructive-1938597⟩ (accessed 15.03.2017)

Gallucci M (2015b) Texas Flooding 2015: Severe Storms In Texas Cause At Least $27M In Infrastructure Damage, Continuing Trend Of Increasing Natural Disasters. *International Business Times* 31 May. Available from: ⟨http://www.ibtimes.com/texas-flooding-2015-severe-storms-texas-cause-least-27m-infrastructure-damage-1945594⟩ (accessed 15.03.2017)

Hamacher HW, Nickel S (1994) Combinatorial algorithms for some 1-facility median problems in the plane. *European Journal of Operational Research* 79(2):340–351.

Javadian N, Tavakkoli-Moghaddam R, Amiri-Aref M, Shiripour S (2014) Two meta-heuristics for a multi-period minisum location-relocation problem with line restriction. *The International Journal of Advanced Manufacturing Technology* 71(1):1033–1048.

Katz IN, Cooper L (1981) Facility location in the presence of forbidden regions, I: Formulation and the case of Euclidean distance with one forbidden circle. *European Journal of Operational Research* 6(2):166–173.

Klamroth K (2001) A reduction result for location problems with polyhedral barriers. *European Journal of Operational Research* 130(3):486–497.

Klamroth K (2004) Algebraic properties of location problems with one circular barrier. *European Journal of Operational Research* 154(1):20–35.

Magnanti T, Wong R (1981) Accelerating Benders decomposition: algorithmic enhancement and model selection criteria. *Operations Research* 29(3):464–484.

Milly PCD, Wetherald RT, Dunne KA, Delworth TL (2002) Increasing risk of great floods in a changing climate. *Nature* 415:514–517.

Moradi S, Raith A, Ehrgott M (2015) A bi-objective column generation algorithm for the multi-commodity minimum cost flow problem. *European Journal of Operational Research* 244(2):369–378.

Nemhauser GL, Wolsey LA (1988) Integer and Combinatorial Optimization. *Integer and Combinatorial Optimization*. Wiley, New York.

Oğuz M, Bektaş T, Bennell J, Fliege J (2016) A modeling framework for solving restricted planar location problems using phi-objects. *Journal of the Operational Research Society* 67(8):1080–1096.

Papadakos N (2008) Practical enhancements to the Magnanti-Wong method. *Operations Research Letters* 36(1):176–195.

Papadakos N (2009) Integrated airline scheduling. *Computers & Operations Research* 36(1):176–195.

Shiripour S, Mahdavi I, Amiri-Aref M, Mohammadnia-Otaghsara M, Mahdavi-Amiri N (2012) Multi-facility location problem in the presence of a probabilistic line barrier: a mixed integer quadratic programming model. *International Journal of Production Research* 50(15):3988–4008.

Schumacher C, Chandler PR, Rasmussen SR (2002) Task allocation for wide area search munitions via iterative network flow. In: *American Control Conference*, Anchorage, 8-10 May 2002. pp. 1917–1922.

Snyder LV (2006) Facility location under uncertainty: a review. *IIE Transactions* 38(7):537–554.

Toothill J (2002) Central European Flooding August 2002. Technical Report. EQECAT, London.

Ulbrich U, Brücher T, Fink AH, Leckebusch GC, Krüger A, Pinto JG (2003) The central European floods of August 2002: Part 1 - Rainfall periods and flood development. *Weather* 58:371–377.