A Distance-Limited Continuous Location-Allocation Problem for Spatial Planning of Decentralized Systems

Kagan Gokbayrak* and Ayse Selin Kocaman†

September 25, 2016

Abstract

We introduce a new continuous location-allocation problem where the facilities have both a fixed opening cost and a coverage distance limitation. The problem might have wide applications especially in the spatial planning of water and/or energy access networks where the coverage distance might be associated with the physical loss constraints. We formulate a mixed integer quadratically constrained problem (MIQCP) under the Euclidean distance setting and present a three-stage heuristic algorithm for its solution: In the first stage, we solve a planar set covering problem (PSCP) under the distance limitation. In the second stage, we solve a discrete version of the proposed problem where the set of candidate locations for the facilities is formed by the union of the set of demand points and the set of locations in the PSCP solution. Finally, in the third stage, we apply a modified Weiszfeld's algorithm with projections, which we propose to incorporate the coverage distance component of our problem, for fine-tuning the discrete space solutions in the continuous space. We perform numerical experiments on three example data sets from the literature to demonstrate the performance of the suggested heuristic method.

Keywords— continuous location-allocation; planar set covering; multi-source Weber problem; decentralized systems

1 Introduction

Source location and allocation problems are essential components of strategic planning for sustainable development. Many problems have been studied to help decision making in this area. Some of these studies include a list of predetermined candidate locations to locate source facilities, thus solve site-selecting location problems in a discrete space. Greenfield development problems, however, involves undeveloped sites that have no existing infrastructure and facilities can be located at any point in a continuous space. This type of facility location problems are known as the site-generating problems [22].

Motivated by the popularity of decentralized systems in energy and water access networks, in this paper, we study a site-generating location-allocation problem for greenfield infrastructure planning. Our aim is to determine the number and the locations of the source facilities, which can be, for example, a solar or a wind power generation system or a water pump serving demand points as

*Department of Industrial Engineering, Bilkent University, 06800 Bilkent, Ankara, Turkey (kgokbayr@bilkent.edu.tr)

†Department of Industrial Engineering, Bilkent University, 06800 Bilkent, Ankara, Turkey (selin.kocaman@bilkent.edu.tr)

a stand-alone system. Assuming that energy or water resource availability is even over the field, the location-allocation decisions are made based on the spatial locations of demand points. Our objective is to minimize the sum of facility opening costs, which are independent of the locations of the facilities, and connection costs to serve demand points such as cable or pipe costs that are linearly increasing in their distances to the serving facilities. Facilities are assumed to be uncapacitated; however, they can only serve demand points within a specified distance. This coverage distance limitation of facilities can be associated with the constraints on voltage drop in energy systems (due to resistance on cables) as in [20] or pressure loss in water systems (due to friction in pipes) as in [13] that are both linearly increasing with distance.

We pose and study a continuous location-allocation problem with a fixed facility opening cost and a limit on the coverage distance of facilities that is related to three well-known problems in the literature: the planar set covering problem (PSCP), the uncapacitated multi-source weber problem (MWP), and the simple plant location problem (SPLP). In the special case, where there is no connection costs between demand points and their serving facilities, our problem reduces to PSCP. The original set covering problem (SCP) considers a finite collection of sets and their costs, and determines the lowest cost sub-collection whose union equals the union of the collection. This problem is known to be an NP-complete problem [19]. Several heuristic methods are proposed to solve SCP that have applications in fields such as crew scheduling (e.g., [8]) and location of emergency facilities (e.g.,[24], [26]), but in the interest of space, we mention only a few of them. A greedy heuristic for SCP is proposed by Chavatal [10]. Therein, it is shown that the ratio of the cost of the cover returned by the greedy heuristic to the cost of the optimal cover grows at most logarithmically in the size of the largest set in the collection. A non-deterministic variation of this greedy heuristic is proposed by Feo and Resende in [16]. Beasley and Chu in [1] proposed a genetic algorithm-based heuristic for SCP. Caprara et al. in [8] proposed a Lagrangian-based heuristic for large scale SCPs that outperformed all existing methods at the time. The PSCP problem considers a finite number of demand points in the Euclidean plane and determines the minimum number of facilities and their locations in the plane such that each demand point is within a certain distance to at least one of these facilities. To solve PSCP exactly, Church [9] defines the circle intersection points set (CIPS) as the locations of all demand points and the intersection points of all circles centered at demand points with a radius of predetermined coverage distance. Then, for each point in CIPS, a set is formed of all demand points that are within the coverage distance from the point. Considering the collection of all these sets, the original version of SCP is solved. It is possible to show that there exists at least one optimal solution to the PSCP in which all facilities are located in the CIPS [15].

MWP is a site-generating location-allocation problem, which is also known as continuous p-median problem. It locates p facilities in the Euclidean plane to serve a finite set of demand points, each having an associated weight. In this problem, each demand point is served by the closest facility and the objective is to minimize the weighted sum of distances to the closest facilities. MWP is known to be an NP-hard problem [23], so several heuristic solution methods are proposed in the literature: Cooper's iterative location-allocation algorithm [11, 12] is the best known algorithm developed for this problem. Starting at an arbitrary solution that divides the set of demand points into p almost-equal sized subsets, the algorithm alternates between location and allocation steps until a local optimal solution is found. In the allocation step, for fixed locations of the facilities, algorithm simply assigns each demand point to its nearest facility (breaking ties arbitrarily), and once the allocations are fixed, in the location step, the problem reduces to p independent single facility location problems that can be solved by the modified Weiszfeld's method in [27]. As the final solution depends on the initial solution, a random multi-start version of this algorithm can be applied. Another line of work is based on the idea of starting at a good initial solution. Based on the observation that the optimal solution of the continuous problem often has several facilities colocated with the demand points, Hansen et al. in [17] proposed the p-median heuristic. This heuristic first solves the p-median problem, which chooses p facility locations from the set of demand points to minimize the weighted sum of distances. Then, p independent single facility location problems are solved as in the location step of the Cooper's algorithm. Recently, Brimberg and Drezner in [3] proposed to overlay the area containing the demand points with a grid. Then, a p-median problem is solved over the nodes of the grid to obtain high-quality starting points for the Cooper's algorithm. Since there is a significant correlation between the qualities of starting and final solutions, starting at the p-median solution improves the algorithm results. In [4], Brimberg et al. proposed an alternating solution procedure where a local search is conducted in the continuous space to obtain a local optimum. The locations from the continuous problem solution is then augmented in the discrete space problem, which is solved again to obtain new starting points for the continuous space problem. This process continues until no further improvement is observed. Finally, in [14], Drezner et al. developed a distribution-based variable neighborhood search and a genetic algorithm, and a hybrid algorithm that combines these two approaches. The hybrid approach outperformed both approaches. For other heuristic, metaheuristic and exact approaches for MWP, readers can refer to a comprehensive review by Brimberg et al. [5].

SPLP is a problem in discrete space, where there are fixed facility opening costs and a finite set of possible locations for facilities. It aims to minimize the sum of facility opening costs and the weighted connection costs. The adjective 'simple' in its name is to state that the facilities are uncapacitated. This problem is widely studied in literature. Krarup and Pruzan in [21] provide a highly cited survey on this problem. It is stated in that paper that SPLP is also an NP-hard problem. The work on the continuous space version of SPLP, however, is very limited. In [6], Brimberg et al. introduced the fixed cost for facilities that is independent of the location. The problem that we consider in this paper reduces to the problem considered in [6] if the coverage distance limitation is removed. They propose a multi-stage heuristic approach for the problem without the coverage constraint. Following the path in [17] of solving the discrete version to obtain an initial solution for the continuous problem, in the first stage of this heuristic, SPLP is solved assuming that the demand points are the potential locations for facilities. Then, in the second stage, a fine tuning is performed in the continuous space using Weiszfeld's method. In [7], Brimberg and Salhi introduced zone-dependent fixed costs for facilities, where they define zones as polygons. An efficient exact solution algorithm for the single facility case is proposed, whereas, for the multi-facility case, they propose heuristic procedures.

In the distance-limited continuous location-allocation problem that we present, the number of facilities to be opened is a decision variable. For a given number of facilities and without a distance limitation, our problem reduces to MWP, which is NP-hard, so our problem is also NP-hard. Therefore, we propose a heuristic solution method, whose final solution quality highly depends on the initial solution quality we obtain from the discrete version of the problem. Employing the demand points as the only possible locations for facilities, as is done in [6], would limit the solution quality of the discrete problem. Augmenting this set of possible locations with a small number of additional promising locations is the main idea presented in this paper. Rather than overlaying the area of demand points by a fine grid, as is done in [3], we propose to solve PSCP under the distance limitation to obtain these additional locations. Note that, even though the number of our additional locations is considerably smaller than the number of nodes of a fine grid, as we present in Section 4, the discrete problem defined over the smaller set yields lower cost solutions especially for tight distance limitations and high facility costs. The main reason for the cost improvements seems to be that the number of facilities required to serve all demand points is less in the discrete problem defined over our augmented set.

We propose a multi-stage heuristic algorithm: In the first stage, we solve PSCP employing the CIPS for the demand points to obtain a set of promising locations to augment the set of demand points. In the second stage, we solve the discrete version of the problem defined over the augmented set. Finally, in the third stage of our heuristic algorithm, starting at the solution of the second stage, we apply Cooper's iterative location-allocation algorithm. Note that, for the location step, we propose a modified version of the Weizsfeld's algorithm [29] to incorporate our coverage distance constraint.

The contributions of this paper can be summarized as follows: We introduce a new problem which has wide applications in the spatial planning of decentralized energy and water distribution systems. Then, we provide the mathematical model of this problem in the continuous space. As the problem is NP-hard, we propose a three-stage heuristic solution algorithm. In order to incorporate the distance limitation constraints, we propose a Weizsfeld's algorithm with projections. We conduct computational experiments to illustrate how the proposed algorithm works under different distance limitations and cost parameters for the discrete problem.

The sections of this paper are outlined as follows: A more precise statement and the mathematical formulation of the problem are given in Section 2. Our heuristic solution method for the problem is explained in Section 3. Computational results along with the discussions are provided in Section 4. We conclude our paper in Section 5.

2 Problem Formulation

Consider a rectangular greenfield of $L \times W$ dimensions with N demand points. Each demand point i is at location (a_i, b_i) and has an associated weight $w_i > 0$. We assume that each demand point is served by a single facility; therefore, we need at most N facilities to serve all demand points.

Both electric voltage and water pressure drop with distance from the source. To prevent from exceeding the maximum allowable drop, there is a limit on the length of each connection. We incorporate this limit in our model by introducing a circular coverage region with radius D_{max} around each facility, and assuming that the demand points outside this region cannot be served by the facility. In this paper, we assume that the total demand in each coverage region can be met by a single facility, so we treat the facilities as uncapacitated. Each facility j is located at (x_j, y_j) and has a fixed opening cost of F if serving any demand points.

Our objective is to determine the number and the location of open facilities, and the assignment of demand points to these facilities to minimize the total cost composed of connection (weighted distance) and facility opening costs. Since the facilities are uncapacitated, each demand point will be served by the closest open facility to minimize its connection cost. We assume that all distances are Euclidean. Defining the decision variables

d_{ij} : the distance between demand point $i \in \{1, \ldots, N\}$ and facility $j \in \{1, \ldots, N\}$

δ_i : the distance between demand point $i \in \{1, \ldots, N\}$ and its closest open facility

$$v_j = \begin{cases} 1, & \text{if facility } j \text{ is open,} \\ 0, & \text{otherwise,} \end{cases} \quad j \in \{1, \ldots, N\};$$

$$z_{ij} = \begin{cases} 1, & \text{if demand point } i \text{ is served by facility } j, \\ 0, & \text{otherwise,} \end{cases} \quad i, j \in \{1, \ldots, N\};$$

we propose to solve the following mixed integer quadratically constrained programming (MIQCP) problem, denoted by (DLim-CLAP):

$$\min \sum_{j=1}^{N} v_j F + \sum_{i=1}^{N} w_i \delta_i \tag{1}$$

subject to

$$\sum_{j=1}^{N} z_{ij} = 1 \qquad i \in \{1,\dots,N\}; \tag{2}$$

$$z_{ij} \le v_j \qquad i,j \in \{1,\dots,N\}; \tag{3}$$

$$\delta_i \le D_{max} \qquad i \in \{1,\dots,N\}; \tag{4}$$

$$\delta_i \ge \sqrt{L^2+W^2}(z_{ij}-1) + d_{ij} \qquad i,j \in \{1,\dots,N\}; \tag{5}$$

$$d_{ij}^x = a_i - x_j \qquad i,j \in \{1,\dots,N\}; \tag{6}$$

$$d_{ij}^y = b_i - y_j \qquad i,j \in \{1,\dots,N\}; \tag{7}$$

$$d_{ij}^2 \ge (d_{ij}^x)^2 + (d_{ij}^y)^2 \qquad i,j \in \{1,\dots,N\}; \tag{8}$$

$$x_j, y_j \in \mathbb{R}, \qquad j \in \{1,\dots,N\}; \tag{9}$$

$$v_j \in \{0,1\}, \qquad j \in \{1,\dots,N\}; \tag{10}$$

$$z_{ij} \in \{0,1\}, \qquad i,j \in \{1,\dots,N\}; \tag{11}$$

$$d_{ij}^x, d_{ij}^y \in \mathbb{R}, \qquad i,j \in \{1,\dots,N\}; \tag{12}$$

$$d_{ij} \ge 0, \qquad i,j \in \{1,\dots,N\}; \tag{13}$$

$$\delta_i \ge 0, \qquad i \in \{1,\dots,N\}. \tag{14}$$

We minimize the total distribution cost in (1) that is composed of facility and connection costs. The constraint set (2) assigns a facility to each demand point. We guarantee by constraints (3) that closed facilities are not assigned to any demand points. The distances of demand points to their closest facilities are upper bounded by D_{max} in the constraint set (4). The lower bounds on these distances are presented in constraints (5). Constraints (6) and (7) define the x-coordinate difference d_{ij}^x and the y-coordinate difference d_{ij}^y, respectively, between each demand point i and each facility j. Employing these differences, the set of quadratic constraints in (8) define the Euclidean distances between demand points and facilities. The decision variables of this optimization model are defined in (9)-(14).

This optimization problem has N^2+N binary and $3N^2+3N$ continuous decision variables, and $6N^2+3N$ constraints. For a given number of facilities and without coverage distance limitations, DLim-CLAP reduces to MWP which is shown to be NP-hard by Megiddo and Supowit [23]. In the next section, we propose a three-stage heuristic method for the solution of DLim-CLAP.

3 A Three-stage Heuristic Algorithm

We follow the steps of Hansen et al. in [17], where a heuristic method to solve MWP is proposed. The discrete counterpart of MWP is the well studied p-median problem where the facility locations are chosen from a given set of candidate locations. While the p-median problem is also an NP-hard problem [18], solving a p-median problem exactly is a lot easier than solving a MWP as discussed by Hansen et al. in [17]. In addition, it is also observed in [17] that some of the optimal facility locations in MWP coincide with the demand locations. Motivated by these observations, Hansen et al., in [17], propose a heuristic for MWP where they first solve the p-median problem with demand locations as the candidate locations for the facilities, and then solve a Weber problem (the problem in [28] of finding a point minimizing the sum of weighted distances from given points) for each facility given the cluster of demand points served by the facility.

In this study, we adopt a similar approach and propose a three-stage heuristic to solve DLim-CLAP. In the first stage, we determine the minimum number of facilities and their locations to cover all demand points under the given coverage distance. In other words, in this stage, we are solving the DLim-CLAP problem with $w_i = 0$ for all demand points i.

In order to illustrate our methods graphically, we present the following running example with 12 demand points shown in Figure 1. In this example, the facility cost is given as $F = 1000$, the coverage distance is given as $D_{max} = 30$, and the weights are given as $w_i = 1$ for all $i \in \{1, \dots, 12\}$.

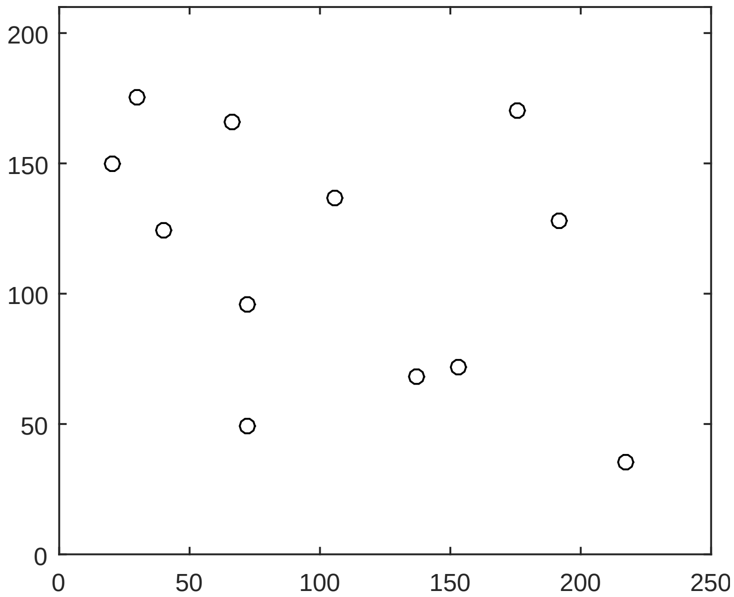

Figure 1: Demand Points

3.1 Stage 1: Solving PSCP

In order to solve the PSCP defined for our coverage distance, we first determine the intersection points of the circles centered at the demand points and with radius D_{max}. These points are suggested by Church [9] to be used to find an optimal solution to PSCP by solving a SCP. We show these points for our running example in Figure 2. Note that if the circle of a demand point does not intersect with any other circle, the center of the circle, the demand point itself, is included in the set of the circle intersection points (see the demand point in the lower right corner of Figure 2). Let us denote the cardinality of this set by C.

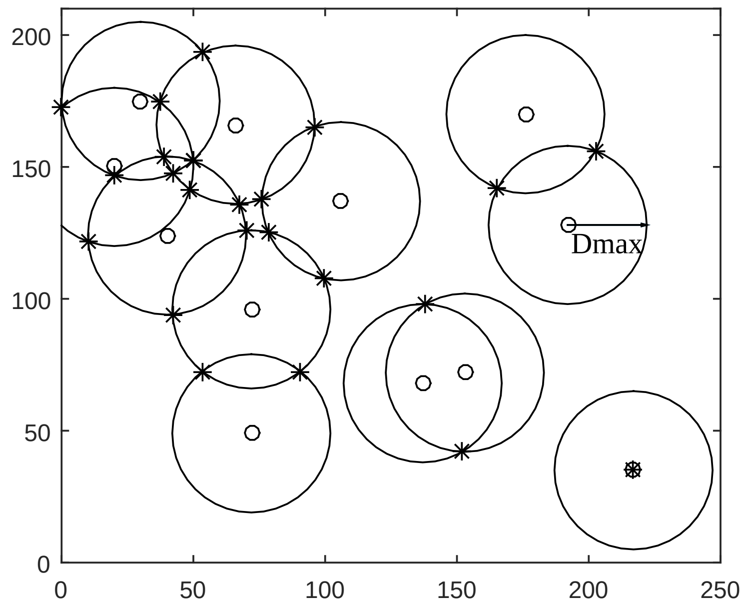

Figure 2: Circle Intersection Points

Then, we determine the coverage region for each point in the set of circle intersection points as in Figure 3.

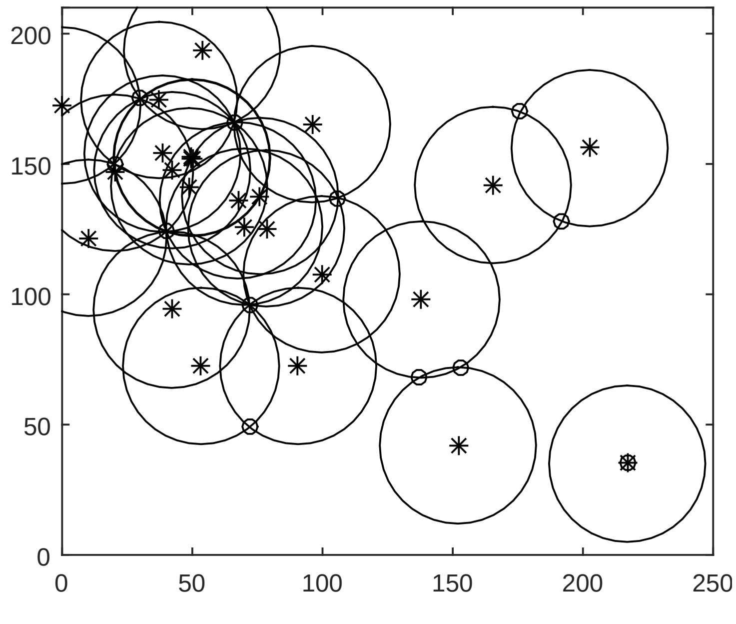

Figure 3: Coverage Regions for Circle Intersection Points

Considering the collection of sets of demand points in each coverage region, we formulate and solve the following set covering problem: For each demand point $i \in \{1, \ldots, N\}$ and for each circle intersection point $k \in \{1, \ldots, C\}$, let us define the coverage parameter

$$\alpha_{ik} = \begin{cases} 1, & \text{if } d_{ik} \leq D_{max}, \\ 0, & \text{otherwise}, \end{cases} \tag{15}$$

In this formulation, d_{ik} denotes the distance between the demand point i and the circle intersection point k.

Then, we solve (SCP) defined as:

$$\min \sum_{k=1}^{C} v_k \tag{16}$$

subject to

$$\sum_{k=1}^{C} \alpha_{ik} v_k \geq 1, \qquad i \in \{1, \ldots, N\}; \tag{17}$$

$$v_k \in \{0, 1\}, \qquad k \in \{1, \ldots, C\}. \tag{18}$$

The objective value of the optimal solution will yield the minimum number of facilities needed. The PSCP solution for our running example is shown in Figure 4.

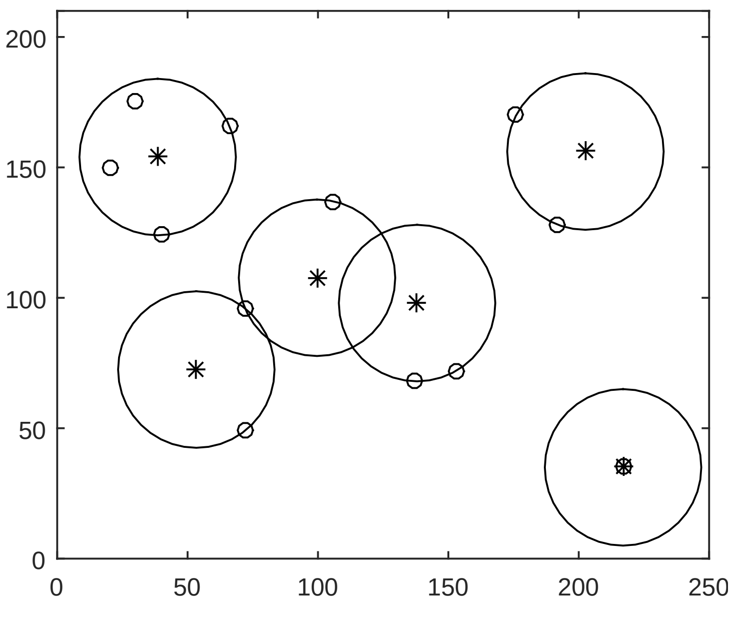

Figure 4: PSCP Solution

Once we obtain the locations of the minimum number of covering facilities ($v_k^* = 1$), we conclude the first stage of our heuristic.

3.2 Stage 2: Determining the number of facilities

In the second stage, we determine the number of facilities by solving the discrete version of DLim-CLAP, which we call distance limited 'plant' location problem, DLim-PLP, to be consistent with the literature. Rather than limiting the candidate locations for facilities to the demand locations as in [6, 7, 17], we augment the set of demand locations with the locations obtained in the first stage to form the candidate locations for facilities. Let the cardinality of this augmented set of candidate locations be denoted by M. The possible locations of facilities in our running example are shown in Figure 5. The circle intersection points in the PSCP solution are indicated by diamond shapes and the demand points are indicated by circles. Note that there exists a demand point in the lower right corner of this figure that coincides with a circle intersection point.

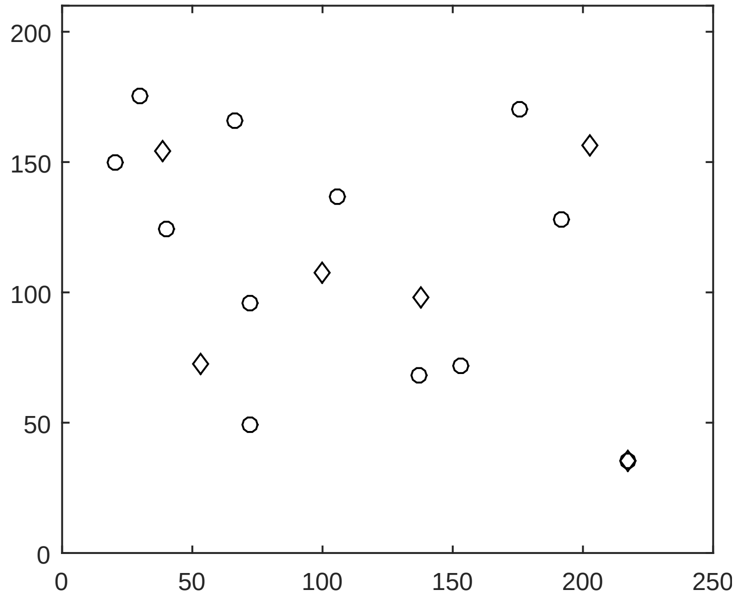

Figure 5: Possible Facility Locations

Since we provide additional candidate locations to DLim-PLP, the solution time is expected to increase but in return we may obtain a better solution. Our computational results show that PSCP solution provides a reasonable number of additional candidate locations that improve performance considerably in several instances without a major increase in the solution times.

We formulate and solve DLim-PLP (i.e, the discrete version of DLim-CLAP) as follows:

$$\min \sum_{j=1}^{M} v_j F + \sum_{i=1}^{N} \sum_{j=1}^{M} z_{ij} w_i d_{ij} \tag{19}$$

subject to

$$\sum_{j=1}^{M} z_{ij} = 1, \qquad i \in \{1, \ldots, N\}; \tag{20}$$

$$z_{ij} \leq v_j, \qquad i \in \{1, \ldots, N\}, j \in \{1, \ldots, M\}; \tag{21}$$

$$\sum_{j=1}^{M} z_{ij} d_{ij} \leq D_{max}, \qquad i \in \{1, \ldots, N\}; \tag{22}$$

$$v_j \in \{0, 1\}, \qquad j \in \{1, \ldots, M\}; \tag{23}$$

$$z_{ij} \in \{0, 1\}, \qquad i \in \{1, \ldots, N\}, j \in \{1, \ldots, M\}. \tag{24}$$

In this formulation, d_{ij} indicates the distance between demand point i and candidate location j. Constraints (20) and (21) are the constraints (2) and (3), respectively, rewritten for the augmented candidate location set. Constraints (22) follow from the constraints (4). We define the facility opening and assignment decision variables in (23) and (24).

The solution of this model yields the number of facilities $V = \sum_{j=1}^{N} v_j$ and the assignments z_{ij} of these facilities to demand points. The solution for our running example is presented in Figure 6, where the locations of the facilities are shown by squares. Note that there are three facilities in this figure that are collocated with the demand points. The rest of the demand points are connected to the facilities in a star topology.

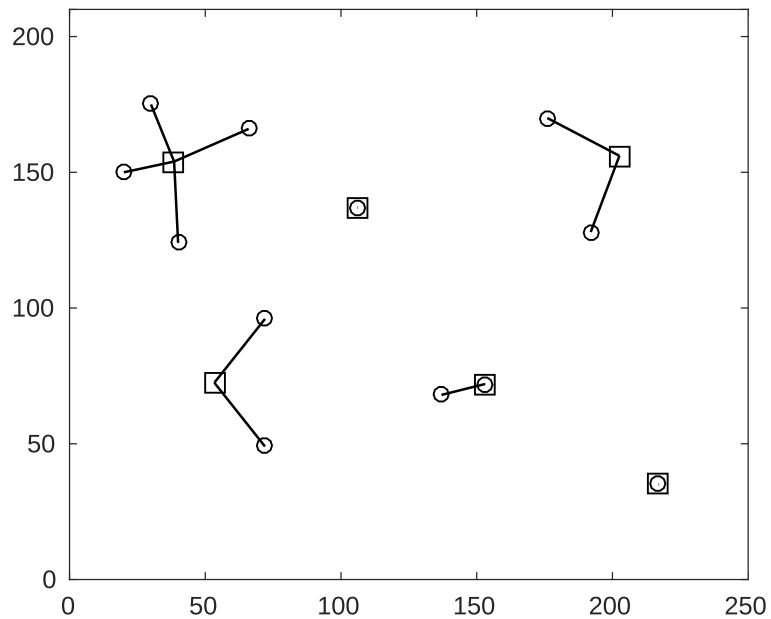

Figure 6: DLim-PLP Solution

The second stage of the heuristic method results with a number of facilities, some collocated with demand points and others possibly on circle intersection points, and the assignments of these facilities to demand points. Let us denote the cluster of demand points for each facility k by C_k defined as

$$C_k = \{i | z_{ik} = 1\} \tag{25}$$

for $k \in \{1, \ldots, V\}$.

Next, we adjust the facility locations in the continuous space to improve the objective value.

3.3 Stage 3: Determining the facility locations in the continuous space

In the third stage, starting with the facility locations obtained in the second stage, we apply Cooper's alternating location and allocation algorithm described in [12]. This algorithm iteratively reallocates demand points to the closest facilities so that clusters are updated and then relocates the facility for each cluster to minimize the weighted distance cost from each cluster, until no changes are observed in the demand point allocations and the facility locations.

At each location phase of the Cooper's algorithm, we solve the optimization problem below, denoted by DLim-Geom, to find the location of the facility for each cluster C_k:

$$min \sum_{i \in C_k} w_i d_{ik} \tag{26}$$

subject to

$$\begin{aligned}
d_{ik} &\leq D_{max}, && i \in C_k; && (27)\\
d_{ik}^x &= a_i - x_k, && i \in C_k; && (28)\\
d_{ik}^y &= b_i - y_k, && i \in C_k; && (29)\\
d_{ik}^2 &\geq (d_{ik}^x)^2 + (d_{ik}^y)^2, && i \in C_k; && (30)\\
x_k, y_k &\in \mathbb{R}, && && (31)\\
d_{ik}^x, d_{ik}^y &\in \mathbb{R}, && i \in C_k; && (32)\\
d_{ik} &\geq 0, && i \in C_k. && (33)
\end{aligned}$$

Without the set of constraints in (27), DLim-Geom reduces to the Weber problem in [28], which aims to find a point that minimizes the sum of weighted distances from the points within the cluster. Vardi and Zhang proposes a modified Weiszfeld algorithm in [27] for the Weber problem.

The set of constraints in (27) limit the feasible region for the facility location as in Figure 7, where we zoom in to the cluster in the upper left corner of Figure 6. In this figure, the facility that serves the four demand points (indicated by the little circles) has to be located within the gray area to satisfy the constraints (27). The algorithm proposed by Vardi and Zhang may locate the facility outside this region; therefore, a projection onto this region is needed.

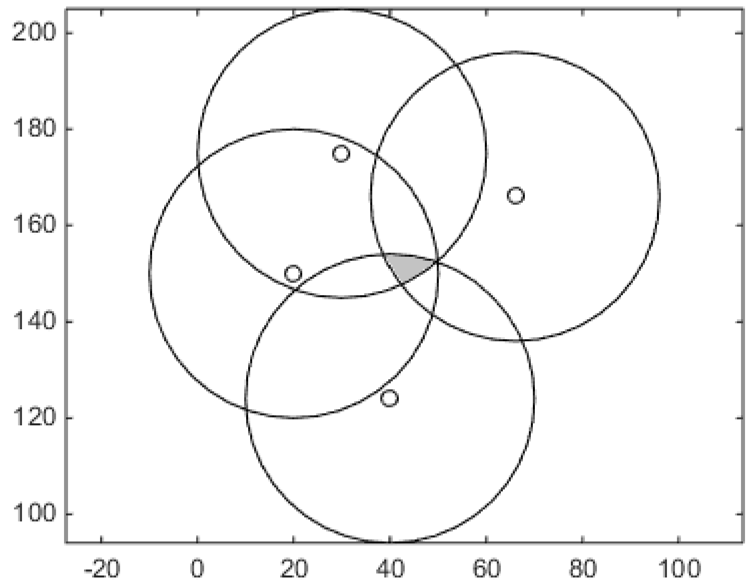

Figure 7: Feasible Region for the Facility

Next, we present our iterative solution method with projections.

3.3.1 An iterative method with projections to solve DLim-Geom

Let (x^0, y^0) and r be the center and the radius, respectively, of the minimum circle enclosing all points in the cluster C_k. In the following discussion, we assume that $r \leq D_{max}$, otherwise DLim-Geom would be infeasible. We start our algorithm at (x^0, y^0), which is a feasible location for DLim-Geom.

Our modification to the algorithm proposed by Vardi and Zhang is to project the proposed locations to the feasible set of DLim-Geom at every iteration. Let (x^t, y^t) and (x^p, y^p) be the current location and the proposed location for the next iteration, respectively. We assume that the current location is feasible for DLim-Geom as we start from a feasible location and apply projection at every iteration. If the proposed location is also feasible, i.e., if the constraints (27) are satisfied by (x^p, y^p) for all demand points in the cluster, then we accept it as the location for the next iteration so that $(x^{t+1}, y^{t+1}) = (x^p, y^p)$. If, on the other hand, the proposed location is not feasible, then we project it onto the feasible region as follows: Let $A \subset C_k$ denote the set of demand points whose distances to the proposed location (x^p, y^p) exceeds D_{max}. For each demand point (a_i, b_i) in A, we determine a location (x_i, y_i) that is both D_{max} away and on the line segment whose end points are (x^t, y^t) and (x^p, y^p) (see (x_i, y_i) and (x_j, y_j) in Figure 8 for points i and j).

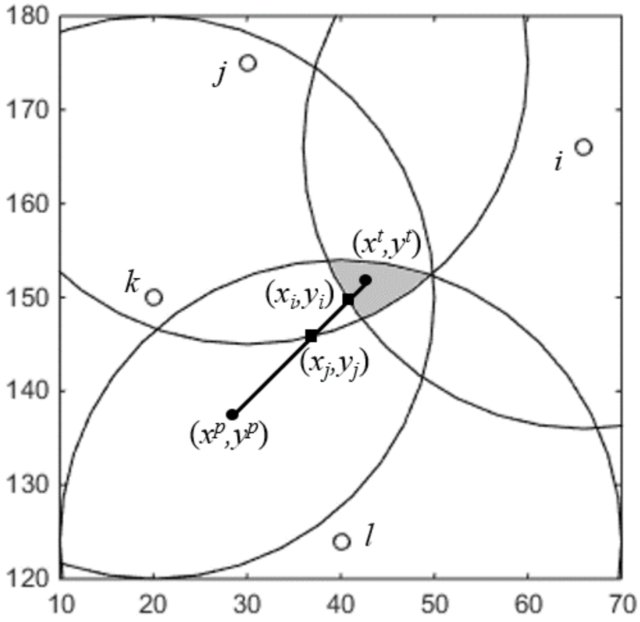

Figure 8: Projection

Among the locations (x_i, y_i) determined for all demand points in the set A, we select the closest one to (x^t, y^t) as the location for the next iteration, so that we preserve feasibility. Hence, when projection is needed, $(x^{(t+1)}, y^{(t+1)})$ is calculated as follows: Let $\beta_i \in [0,1]$ be the solution to the second order polynomial equation:

$$\big(x^p + \beta_i(x^t - x^p) - a_i\big)^2 + \big(y^p + \beta_i(y^t - y^p) - b_i\big)^2 = D^2_{max} \tag{34}$$

for all $i \in A$. We determine $\beta_{max} = \max_{i\in A} \beta_i$ and set

$$(x^{(t+1)}, y^{(t+1)}) = \beta_{max}(x^t, y^t) + (1 - \beta_{max})(x^p, y^p) \tag{35}$$

Applying the third stage on our running example, we obtain the solution shown in Figure 9 with a cost of 6210. It was possible to solve DLim-CLAP for this small example, so we verified that the solution in Figure 9 is the optimal solution.

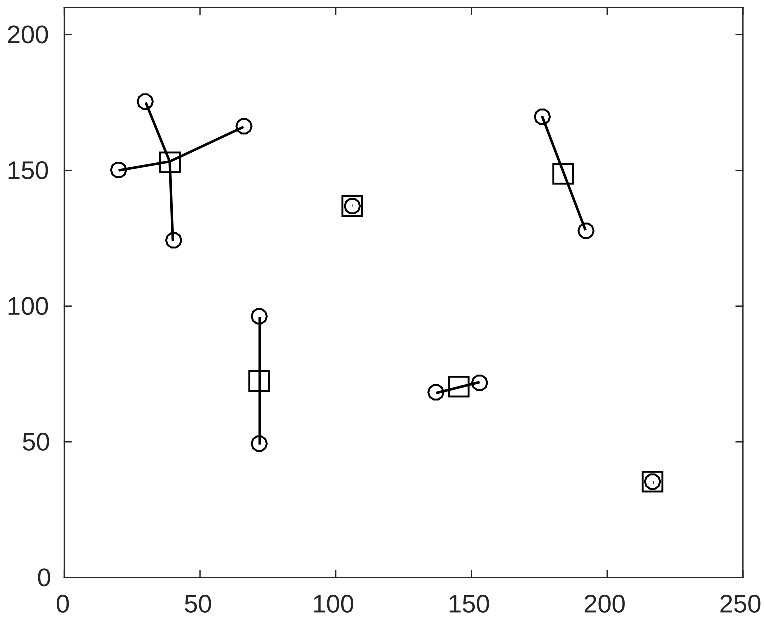

Figure 9: Final Solution

4 Computational Results

In this section, we perform experiments on three sets of data that are widely used for studying MWP [5]. These data sets include the 287 node example from [2], and 654 and 1060 customer problems from the TSP library [25]. We denote these sets by W287, U654, and U1060, and present their demand locations in Figure 10. W287 has 287 demand points with weights w_i ranging between 1 and 698. U654 and U1060, on the other hand, have 654 and 1060 demand points, respectively, each with a unit weight $w_i = 1$. These data sets are also used in [6] for the multi-source weber with constant opening cost problem.

In our experiments, the cost of deploying a facility F takes values from the sets $\{50, 100, 200, 500, 5000\}$, $\{1000, 2000, 5000, 10000, 15000\}$, and $\{1000, 2000, 5000, 10000, 15000\}$ for data sets W287, U654, and U1060, respectively. Distance limits D_{max} are selected from $\{5, 10, 15, 20, 25\}$ for W287 and from $\{200, 400, 600, 800, 1000\}$ for both U654 and U1060. Our computational experiments are performed on a dual 2.4GHz Intel Xeon E5-2630 v3 CPU server with 64GB RAM. The optimization problems that are formed in Matlab R2016a are solved using CPLEX 12.6.3 in parallel mode using upto 32 threads. We enforce a CPU time limit of ten hours on all our optimization models.

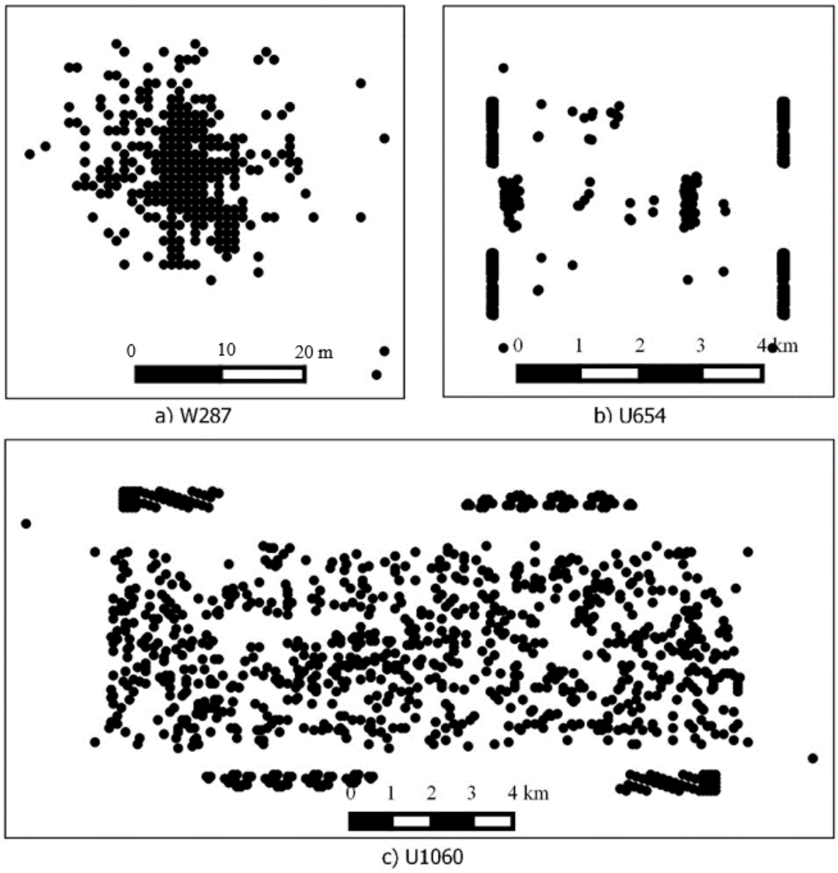

Figure 10: Demand Locations in Example Data Sets

4.1 Solving PSCP

In this section, we present some implementation details about the first stage of our algorithm on the three data sets. The first set of columns in Table 1 presents the details of each instance. In the second set of columns, we present the cardinality C of the CIPS and the minimum number of facilities needed (obtained from PSCP) to cover all demand points when their locations are selected from the CIPS, and the solution CPU times in seconds. Note that some of the demand points may also be included in CIPS; therefore, the column titled PSCP presents an upper bound on the number of additional points supplied by PSCP to the second stage.

Data	D_{max}	C	PSCP	Time
W287	5	34025	14	1
	10	71642	6	25
	15	80516	3	22
	20	81719	2	27
	25	82048	2	27
U654	200	50850	36	1
	400	74802	18	2
	600	86951	13	5
	800	97830	9	3
	1000	126800	7	7
U1060	200	8131	299	0
	400	22971	128	13
	600	45020	73	28
	800	75990	50	938
	1000	112484	35	11

Table 1: PSCP Solutions

Since the exact solution times are all less than half a minute except for one instance, we did not implement any heuristic methods for PSCP.

4.2 Effect of Using an Augmented Set of Candidate Locations

In this section, we demonstrate the benefit of adding the PSCP solutions to the set of demand points while forming the set of candidate locations for the discrete problem DLim-PLP.

In Table 2, we present our results for the instances of the data sets W287 and U654. This table is organized as follows: The first set of columns presents the details of each instance. The second set of columns presents the number of facilities p, the corresponding costs in DLim-PLP, and the CPU times in seconds for solving the discrete problem defined over the set of demand points. We also report the cost of the continuous solution obtained at the end of the third stage. The third stage takes less than a second; therefore, we do not report its solution times. The third set of columns presents the same set of results for the discrete problem defined over the set consisting of demand and PSCP locations, and its corresponding final solution. In the last set of columns, we show the cost improvements due to the additional candidate locations for both the second and third stage solutions. These improvements are calculated as the difference between the two methods' costs divided by the cost of the former method.

Table 2 indicates that including the PSCP locations in the candidate locations set lower the number of facilities in the solutions of DLim-PLP. The differences between these numbers cause substantial improvements in the cost, especially for large F values. Note that augmenting the problem with the additional candidate locations that are obtained from PSCP resulted with upto 23% improvements in the costs for instances of W287 and U654.

Since the number of additional candidate locations is small compared to the number of demand points, we did not observe a major change in the computation time for DLim-PLP. The CPU times of both models were comparable.

We report the solutions for the data set U1060 in Table 3, which is organized the same way as Table 2.

			w/o PSCP				w/ PSCP				% Diff.	
Data	F	D_{max}	p	2S Cost	Time	3S Cost	p	2S Cost	Time	3S Cost	2S	3S
		5	53	4157	1	4157	50	4055	1	4033	2.45%	2.98%
		10	46	3951	7	3951	45	3926	7	3926	0.63%	0.63%
W287	50	15	44	3890	10	3889	44	3890	9	3889	0.00%	0.00%
		20	44	3890	12	3889	44	3890	9	3889	0.00%	0.00%
		25	43	3885	13	3884	43	3885	11	3884	0.00%	0.00%
		5	42	6511	1	6499	39	6285	1	6272	3.47%	3.49%
		10	34	5932	7	5932	33	5857	6	5857	1.25%	1.25%
W287	100	15	32	5770	9	5770	32	5770	9	5770	0.00%	0.00%
		20	32	5770	11	5770	32	5770	11	5770	0.00%	0.00%
		25	31	5717	12	5717	31	5717	11	5717	0.00%	0.00%
		5	32	10236	1	10220	29	9643	1	9633	5.79%	5.74%
		10	23	8746	6	8741	22	8587	7	8586	1.82%	1.77%
W287	200	15	20	8359	13	8357	20	8359	12	8357	0.00%	0.00%
		20	20	8313	14	8305	20	8313	20	8305	0.00%	0.00%
		25	19	8171	14	8157	19	8171	16	8157	0.00%	0.00%
		5	23	17964	1	17905	18	16026	1	16004	10.79%	10.62%
		10	13	13609	11	13609	12	13194	12	13152	3.05%	3.36%
W287	500	15	10	12312	21	12290	10	12312	21	12290	0.00%	0.00%
		20	9	12134	16	12113	9	12060	26	12024	0.61%	0.73%
		25	8	11635	16	11635	8	11635	18	11635	0.00%	0.00%
		5	19	106815	2	106008	14	83889	1	82867	21.46%	21.83%
		10	8	53214	24	52903	7	47641	29	47202	10.47%	10.78%
W287	5K	15	4	35713	29	35670	4	35665	26	35603	0.13%	0.19%
		20	3	31455	40	31217	2	27618	45	27555	12.20%	11.73%
		25	2	26837	32	26837	2	26837	33	26837	0.00%	0.00%
		200	46	80602	11	79555	44	79515	16	78456	1.35%	1.38%
		400	41	77483	13	76959	37	76544	9	75520	1.21%	1.87%
U654	1K	600	34	75731	10	74966	34	75731	11	74966	0.00%	0.00%
		800	32	74471	13	73854	32	74471	13	73854	0.00%	0.00%
		1000	32	74471	12	73854	32	74471	12	73854	0.00%	0.00%
		200	42	124360	13	123359	40	121273	15	120260	2.48%	2.51%
		400	36	115820	15	115356	30	109659	8	108248	5.32%	6.16%
U654	2K	600	27	106219	14	105060	26	105419	17	103660	0.75%	1.33%
		800	25	103187	13	102522	25	103187	12	102522	0.00%	0.00%
		1000	24	103126	13	102145	24	103126	12	102145	0.00%	0.00%
		200	38	247657	7	246695	36	238570	14	237596	3.67%	3.69%
		400	28	212358	21	211994	21	184903	17	183633	12.93%	13.38%
U654	5K	600	20	175888	22	174681	19	171092	27	169085	2.73%	3.20%
		800	15	163606	26	162996	14	163118	27	161256	0.30%	1.07%
		1000	13	156020	14	155270	13	156020	15	155270	0.00%	0.00%
		200	38	437657	8	436695	36	418570	15	417596	4.36%	4.37%
		400	28	352358	14	351994	20	284992	22	283833	19.12%	19.36%
U654	10K	600	17	266005	28	265132	15	252508	30	249556	5.07%	5.87%
		800	15	238606	34	237996	14	233118	36	231256	2.30%	2.83%
		1000	12	220637	15	219505	12	220637	14	219505	0.00%	0.00%
		200	38	627657	8	626695	36	598570	14	597596	4.63%	4.64%
		400	28	492358	16	491994	18	380864	28	378263	22.64%	23.12%
U654	15K	600	17	351005	31	350132	15	327508	36	324556	6.69%	7.30%
		800	15	313606	38	312996	12	296921	44	295641	5.32%	5.54%
		1000	11	280420	15	279336	11	280420	16	279336	0.00%	0.00%

Table 2: Method Comparisons on W287 and U654

		w/o PSCP				w/ PSCP				% Diff.	
F	D_{max}	p	2S Cost	Time	3S Cost	p	2S Cost	Time	3S Cost	2S	3S
1K	200	510	576862	5	575950	299	447587	6	431613	22.41%	25.06%
	400	195	375929	3	373518	184	372645	3	369876	0.87%	0.98%
	600	164	365866	3	363891	165	365789	3	363824	0.02%	0.02%
	800	158	364061	3	362693	158	364061	4	362693	0.00%	0.00%
	1000	158	364061	7	362693	158	364061	8	362693	0.00%	0.00%
2K	200	510	1086862	9	1085950	299	746587	6	730613	31.31%	32.72%
	400	175	558629	3	550432	140	532503	4	521681	4.68%	5.22%
	600	108	492312	5	489608	107	491587	5	489118	0.15%	0.10%
	800	96	485589	6	483775	96	485589	5	483775	0.00%	0.00%
	1000	96	485315	8	483655	96	485315	8	483655	0.00%	0.00%
5K	200	510	2616862	6	2615950	299	1643587	7	1627613	37.19%	37.78%
	400	174	1081232	3	1072291	128	920491	3	906865	14.87%	15.43%
	600	93	782911	16	775602	80	758307	7	749739	3.14%	3.33%
	800	66	718288	11	714008	65	715394	11	711085	0.40%	0.41%
	1000	62	708087	17	705836	62	708087	16	705836	0.00%	0.00%
10K	200	510	5166862	8	5165950	299	3138587	8	3122613	39.26%	39.55%
	400	174	1951232	3	1942291	128	1560491	3	1546865	20.03%	20.36%
	600	92	1246614	20	1238409	77	1146893	6	1137596	8.00%	8.14%
	800	59	1025468	816	1017621	56	1009890	107	1004476	1.52%	1.29%
	1000	47	971768	31	967574	46	968577	37	964311	0.33%	0.34%
15K	200	510	7716862	7	7715950	299	4633587	7	4617613	39.96%	40.15%
	400	174	2821232	3	2812291	128	2200491	3	2186865	22.00%	22.24%
	600	92	1706614	24	1698409	73	1517581	6	1503049	11.08%	11.50%
	800	58	1315502	1618	1306441	52	1281530	136	1274254	2.58%	2.46%
	1000	44	1200859	26482	1193058	41	1183629	15339	1176580	1.43%	1.38%

Table 3: Method Comparisons on U1060

Table 3 also indicates that the solution times of both models are comparable. As also observed in the instances of W287 and U654, assuming PSCP solutions as possible locations for facilities decreased the number of facilities needed considerably. In the instances of U1060 that we present, we observe cost differences upto 40%.

In order to explain the big differences in cost, we present in Table 4 the minimum number of facilities needed to cover all demand points under both candidate location sets. As the F value is increased, the cost difference percentages approach to the difference percentages presented in this table. Hence, we view the decrease in the minimum number of facilities needed as the main reason for cost improvements.

Additional candidate locations in DLim-PLP is expected to lower the costs. In the following analysis, we show that the number of additional locations obtained from the PSCP solution is small; however, the cost improvement is substantial compared to the size of the additional locations set.

In Table 5, we compare our proposed method with two alternative methods in terms of cost improvements on some of U1060 instances. The first alternative method solves DLim-PLP after adding 4000 random locations from the minimum rectangle covering the demand locations to the set of demand points while forming the set of candidate locations. Since the result would depend on the random locations generated, we report the best cost of 100 replications for each instance. The second alternative method overlays a grid of 40x100 on the area containing the demand points and adds 4000 grid nodes to the set of demand points while forming the set of candidate locations. Then, DLim-PLP is solved with the augmented set of candidate locations.

Table 5 is organized as follows: The first set of columns presents the parameters of the instances. The third column presents the baseline cost, which is determined by solving DLim-PLP with demand locations as the only candidate locations for facilities. The third set of columns present the number of added candidate locations, the resulting costs, and the percentage improvements. The number of

Data	D_{max}	w/o CIPS	w/ CIPS	%Diff
W287	5	19	14	26.32%
	10	7	6	14.29%
	15	4	3	25.00%
	20	3	2	33.33%
	25	2	2	0.00%
U654	200	38	36	5.26%
	400	28	18	35.71%
	600	16	13	18.75%
	800	11	9	18.18%
	1000	8	7	12.50%
U1060	200	510	299	41.37%
	400	174	128	26.44%
	600	92	73	20.65%
	800	58	50	13.79%
	1000	43	35	18.60%

Table 4: Minimum Number of Facilities

additional candidate locations for the other two methods are 4000 for each instance, hence we do not include this information in the table. In the fourth and fifth set of columns, we present the costs and the percentage improvements for methods adding random locations and grid nodes, respectively. The percentage improvements are calculated as the decrease in cost divided by the cost in the third column. For each instance, we indicate the best method by a boldface entry.

Data		Baseline	w/ PSCP			w/ Random		w/ Grid	
F	Dmax	Cost	# Added	Cost	%Diff	Cost	%Diff	Cost	%Diff
1000	200	575950	299	431613	**25.06%**	492857	14.43%	490428	14.85%
	400	373518	128	369876	0.98%	368398	1.37%	368390	**1.37%**
	600	363891	73	363824	0.02%	363177	**0.20%**	363554	0.09%
	800	362693	50	362693	0.00%	362263	**0.12%**	362579	0.03%
	1000	362693	35	362693	0.00%	362263	**0.12%**	362583	0.03%
2000	200	1085950	299	730613	**32.72%**	887857	18.24%	884428	18.56%
	400	550432	128	521681	**5.22%**	524131	4.78%	521913	5.18%
	600	489608	73	489118	0.10%	486240	**0.69%**	487664	0.40%
	800	483775	50	483775	0.00%	483130	**0.13%**	483515	0.05%
	1000	483655	35	483655	0.00%	483061	**0.12%**	483396	0.05%
5000	200	2615950	299	1627613	**37.78%**	2072857	20.76%	2066428	21.01%
	400	1072291	128	906865	**15.43%**	969012	9.63%	958903	10.57%
	600	775602	73	749739	**3.33%**	750358	3.25%	755796	2.55%
	800	714008	50	711085	0.41%	709724	**0.60%**	710146	0.54%
	1000	705836	35	705836	0.00%	705121	**0.10%**	705362	0.07%
10000	200	5165950	299	3122613	**39.55%**	4047857	21.64%	4036428	21.86%
	400	1942291	128	1546865	**20.36%**	1709012	12.01%	1683903	13.30%
	600	1238409	73	1137596	**8.14%**	1163261	6.07%	1175928	5.05%
	800	1017621	50	1004476	1.29%	994491	**2.27%**	1003110	1.43%
	1000	967574	35	964311	0.34%	955930	**1.20%**	957044	1.09%

Table 5: Cost Improvements in U1060 instances

Table 5 indicates that, even though the number of additional candidate locations is a lot smaller, our proposed method outperforms the other two alternatives when D_{max} is small and F is large. In the instances where the other methods outperformed our method, their costs were at most 1% lower.

5 Conclusion

We introduced a new continuous location-allocation problem with a distance limitation that is applicable to water and energy distribution systems. We presented an NP-hard MIQCP formulation and proposed a heuristic solution method. Our heuristic method is based on solving a discrete version of the problem to obtain an initial solution for the Cooper's algorithm that obtains a local optimum solution in the continuous space. The candidate facility locations of discrete version of the problem included not only the demand locations but also the locations in the PSCP solution under the distance limitation. Even though the number of additional candidate locations is small, we observed substantial improvements in cost, as it was feasible to serve all demand points with fewer facilities.

The location step of Cooper's algorithm, which utilized Weiszfeld's method, was also modified to incorporate the distance limitation. We proposed a projection method to preserve feasibility at every iteration.

In this study, we utilized optimization solvers in the first two steps of our three-step heuristic method. Other heuristic methods based on agglomerative and divisive clustering that do not utilize optimization solvers are subjects of our ongoing research.

References

[1] J. E. Beasley and P. C. Chu. A genetic algorithm for the set covering problem. *European Journal of Operational Research*, 94(2):392–404, 1996.

[2] I. Bongartz, P. H. Calamai, and A. R. Conn. A projection method for lp norm location-allocation problems. *Mathematical Programming*, 66(1-3):283–312, 1994.

[3] J. Brimberg and Z. Drezner. A new heuristic for solving the p-median problem in the plane. *Computers & Operations Research*, 40(1):427–437, 2013.

[4] J. Brimberg, Z. Drezner, N. Mladenović, and S. Salhi. A new local search for continuous location problems. *European Journal of Operational Research*, 232(2):256–265, 2014.

[5] J. Brimberg, P. Hansen, N. Mladenovic, and S. Salhi. A survey of solution methods for the continuous location-allocation problem. *International Journal of Operations Research*, 5(1):1–12, 2008.

[6] J. Brimberg, N. Mladenovic, and S. Salhi. The multi-source weber problem with constant opening cost. *Journal of the Operational Research Society*, 55:640–646, 2004.

[7] J. Brimberg and S. Salhi. A continuous location-allocation problem with zone-dependent fixed cost. *Annals of Operations Research*, 136(1):99–115, 2005.

[8] A. Caprara, M. Fischetti, and P. Toth. A heuristic method for the set covering problem. *Operations research*, 47(5):730–743, 1999.

[9] R. L. Church. The planar maximal covering location problem. *Journal of Regional Science*, 24(2):185–201, 1984.

[10] V. Chvatal. A greedy heuristic for the set-covering problem. *Mathematics of operations research*, 4(3):233–235, 1979.

[11] L. Cooper. Location-allocation problems. *Operations research*, 11(3):331–343, 1963.

[12] L. Cooper. Heuristic methods for location-allocation problems. *Siam Review*, 6(1):37–53, 1964.

[13] J. Douglas, J. Gasiorek, and J. Swaffield. *Fluid Mechanics. 1995.* Longman Group Ltd., 1979.

[14] Z. Drezner, J. Brimberg, N. Mladenović, and S. Salhi. New heuristic algorithms for solving the planar p-median problem. *Computers & Operations Research*, 62:296–304, 2015.

[15] H. A. Eiselt and C.-L. Sandblom. *Decision analysis, location models, and scheduling problems.* Springer Science & Business Media, 2013.

[16] T. A. Feo and M. G. C. Resende. A probabilistic heuristic for a computationally difficult set covering problem. *Operations Research Letters*, 8:67–71, 1989.

[17] P. Hansen, N. Mladenović, and E. Taillard. Heuristic solution of the multisource weber problem as a p-median problem. *Operations Research Letters*, 22(2):55–62, 1998.

[18] O. Kariv and S. L. Hakimi. An algorithmic approach to network location problems. ii: The p-medians. *SIAM Journal on Applied Mathematics*, 37(3):539–560, 1979.

[19] R. M. Karp. *Reducibility among combinatorial problems.* Springer, 1972.

[20] A. S. Kocaman, W. T. Huh, and V. Modi. Initial layout of power distribution systems for rural electrification: A heuristic algorithm for multilevel network design. *Applied Energy*, 96:302–315, 2012.

[21] J. Krarup and P. M. Pruzan. The simple plant location problem: survey and synthesis. *European Journal of Operational Research*, 12(1):36–81, 1983.

[22] R. Love, J. Morris, and G. Wesolowsky. *Facilities Location: Models and Methods.* North-Holland, New York, 1988.

[23] N. Megiddo and K. J. Supowit. On the complexity of some common geometric location problems. *SIAM journal on computing*, 13(1):182–196, 1984.

[24] H. K. Rajagopalan, C. Saydam, and J. Xiao. A multiperiod set covering location model for dynamic redeployment of ambulances. *Computers & Operations Research*, 35(3):814–826, 2008.

[25] G. Reinelt. Tspliba traveling salesman problem library. *ORSA journal on computing*, 3(4):376–384, 1991.

[26] C. Toregas, R. Swain, C. ReVelle, and L. Bergman. The location of emergency service facilities. *Operations Research*, 19(6):1363–1373, 1971.

[27] Y. Vardi and C.-H. Zhang. A modified weiszfeld algorithm for the fermat-weber location problem. *Mathematical Programming*, 90(3):559–566, 2001.

[28] A. Weber. *Über den Standort der Industrien. Tübingen: JCB Mohr. English translation: The Theory of the Location of Industries.* Chicago: Chicago University Press, 1929.

[29] E. Weiszfeld. Sur le point pour lequel la somme des distances de n points donnés est minimum. *Tohoku Math. J*, 43(355-386):2, 1937.