A Bi-objective Analysis of the R-All-Neighbor P-Center Problem

Author information intentionally left blank

Abstract

In this paper we consider a generalization of the p-center problem called the r-all-neighbor p-center problem (RANPCP). The objective of the RANPCP is to minimize the maximum distance from a demand point to its r^{th}-closest located facility. The RANPCP is applicable to facility location with disruptions because it considers the maximum transportation distance after $(r-1)$ facilities are disrupted. While this problem has been studied from a single-objective perspective, this paper studies two bi-objective versions. The main contributions of this paper are 1) algorithms for computing the Pareto-efficient sets for two pairs of objectives (closest distance vs r^{th}-closest distance and cost vs. r^{th}-closest distance) and 2) an empirical analysis that gives several useful insights into the RANPCP. Based on the empirical results, the RANPCP produces solutions that not only minimize vulnerability but also perform reasonably well (43% from optimal, on average) when disruptions do not occur. In contrast, if disruptions are not considered when locating facilities, the consequence due to facility disruptions is about 630% higher, on average, than if disruptions had been considered. Thus, our results show the importance of optimizing for vulnerability. Therefore, we recommend a bi-objective analysis.

Keywords: Facility location; Interdiction; Stackelberg game; Mixed-integer programming; Bi-objective programming; Disruptions; Risk assessment

1. Introduction

In this paper we consider the problem of locating facilities that are subject to disruptions. In particular, we study the r-all neighbor p-center problem (RANPCP), in which the objective is to minimize the maximum distance from a demand point to its r^{th} closest facility, which is the maximum distance from a demand point to its closest non-disrupted facility after a worst-case disruption of $(r-1)$ facilities occurs. In addition to studying the maximum r^{th} closest distance objective, we also study the maximum 1^{st} closest distance objective, which is the objective of the classic p-center problem. In other words, we study both the post-disruption and pre-disruption performance of the system in order to give decision-makers a more complete picture of their system of facilities. Toward this end, we present algorithms for computing the complete Pareto-efficient set for the following pairs of objectives: (1) maximum 1^{st} closest distance vs. maximum r^{th} closest distance and (2) cost of locating facilities vs. maximum r^{th} closest distance.

The facility location problem is a fundamental problem that has been studied for a long time by researchers from many different disciplines. Over time, as researchers began to develop location models for specific applications such as locating fire stations and ambulances, location models began to include the unavailability of facilities and vehicles in order to reflect reality. For example, ambulances in large metropolitan areas are very busy and not always available for service.

In response, researchers began developing deterministic facility location models that address facility unavailability by considering backup coverage. Other researchers followed by considering the use of backup coverage as a method for mitigating against facility unavailability caused by terrorist attacks, random failures of facilities, and congestion of servers. As a result, facility location research has developed further to include backup-coverage extensions of the p-median, p-center, set covering problem, and maximal set covering problem. Most of the literature on facility location with backup coverage has focused on the degradation in overall service incurred when some facilities become unavailable and unable to serve customers. Specifically, most of the models involve locating backup facilities to minimize this potential degradation.

Some research has considered that facilities become unavailable because of random causes: natural or man-made disasters, congestion of servers, etc. Drezner (1987) was the first to consider random facility failures in the p-median model and his research was extended by others (Lee, 2001; Snyder and Daskin, 2005; Berman et al., 2007), and also modified to study facility protection instead of location (Li et al., 2013). Snyder and Daskin (2005) and Cui et al. (2011) have modeled facility failures in the fixed-charge location problem. Daskin (1982, 1983) was among the first to consider random facility unavailability in the maximal covering location problem. His work was subsequently extended by Batta et al. (1989), who explicitly included queuing in their model.

Rather than considering random failures, other research has sought to minimize the worst case degradation in service; in other words, they measure the risk of facility failures by the worst case degradation. This research is motivated by the facility interdiction problem (Aksen et al., 2012; Church et al., 2004; Zhu et al., 2013), in which an attacker seeks to cause a maximal disruption to a set of facilities. Several papers have included backup coverage in the set covering model, which ensures adequate coverage in the event of facility unavailability. Van Slyke (1982) was the first to include backup coverage in a set covering model and Church and Gerrard (2003) worked on the location set covering problem with facility failures, in which only one vehicle can be located at a potential location. One of the earliest models involving the maximal covering problem with backups was by Daskin and Stern (1981), who minimized two objectives: the total amount of demand coverage and the number of facilities located. This research has since been followed by others (see Brotcorne et al. (2003) for a survey). Aksen et al. Aksen and Aras (2012); Aksen et al. (2013) have studied the problem of locating facilities to minimize the worst case degradation in service. Rather than locating facilities, other researchers have examined the question of how to optimally allocate protection resources among a set of facilities (Scaparra and Church, 2008; Liberatore et al., 2011).

Several authors have studied a location problem called the r-neighbor p-center problem (RNPCP, an extension of the p-center problem in which the problem is to locate p facilities amongst a set of nodes in order to minimize the maximum distance from a client node, defined as a node that does not have a facility located on it, to its r^{th} closest located facility. This problem is also applicable for locating emergency vehicles that must respond to events requiring more than one vehicle. In this context he RNPCP minimizes the maximum response time of the r^{th} vehicle to a demand point. The RNPCP can also be used to minimize the maximum response time of a single vehicle when $(r-1)$ vehicles are busy. Krumke (1995) developed a 4-approximation algorithm[1] for the RNPCP. Chaudhuri et al. (1998) and Khuller et al. (2000) independently developed different 2-approximation algorithms for the RNPCP and show that a better approximation cannot be obtained in polynomial time.

Other authors have studied a version of the r-neighbor p-center problem in which all nodes are client nodes, rather than defining a client node as a node that does not have a facility located on it. Drezner

[1]An α-approximation algorithm is an algorithm that is guaranteed to find a solution with an objective function value of no more than α times the optimal objective value.

(1987) called this problem the (p, r)-center problem and described a heuristic algorithm for the version where facilities can be located anywhere in a plane. Khuller et al. (2000) named this problem the r-all-neighbor p-center problem (RANPCP) and provided approximation algorithms that guarantee an approximation factor of 3 and if $r < 4$, an approximation factor of 2.

Elloumi et al. (2004) presented a new model and an exact solution method for the p-center problem (PCP) and mentioned that their model and solution method can also be used to solve the RANPCP. They find that the LP relaxation bound of their model is at least as good as that of the standard p-center MIP model (see Daskin (1995)) and found that in many cases their bound is strictly better. They also demonstrate that a tight lower bound can be computed by solving a polynomial number of linear programs within a binary search algorithm, showing that their lower bound is at least 1/3 of the optimal objective when the distances obey the triangle inequality and at least 1/2 of the optimal objective when distances are symmetric. However, they do not prove that the approximation factors for their bounds are valid for the RANPCP. Their computational results show when they incorporate their lower bound into the standard binary search algorithm for the PCP, the binary search algorithm is able to solve PCP instances of up to 1817 nodes. Because Elloumi et al. (2004) focused on the PCP, they leave out the details needed to extend their lower bound to the RANPCP and only present empirical results for the PCP.

This article extends the existing literature on the RANPCP by studying the bi-objective version of the RANPCP. Although there has been work on the facility location problem with multiple objectives (see Current et al. (1990)), only a few studies have examined multiple objectives in the facility location problem with disruptions. Snyder and Daskin (2005) optimized a weighted combination of the system performance before and after disruptions for the p-median problem with random disruptions, and O'Hanley and Church (2011) did the same for the maximum covering location problem with interdiction. However, optimizing a weighted combination of two objective functions is only guaranteed to produce the Pareto-efficient set if the two objective functions are convex (Berube et al., 2009), which is not the case for discrete problems such as the RANPCP. Hernandez et al. (2013) perform a tri-objective analysis of the uncapacitated facility location problem. However, because they use an evolutionary algorithm, their approach is not guaranteed to find the complete Pareto-efficient set. This article is the first to describe a method for computing the complete Pareto-efficient set for a facility location problem with disruptions.

The main contributions of this article are the following. (1) An algorithm is presented for computing the Pareto-efficient set for combinations of two objectives: closest distance vs r^{th}-closest distance and cost vs. r^{th}-closest distance. (2) Empirical testing of the single- and bi-objective RANPCP suggest several decision-making insights. In addition to these main contributions, empirical testing indicates that a simple MIP formulation for the single-objective RANPCP provides computational gains over an MIP formulation by Elloumi et al. (2004).

2. Single-objective Problem

The r-all-neighbor p-center problem (RANPCP) can be defined as

> *locate p facilities amongst a set of candidate locations in order to minimize the maximum distance from a demand point to its r^{th} closest facility.*

For the sake of brevity, we will refer to the maximum distance as the *radius* (Elloumi et al., 2004). Because we are concerned with facility disruptions, the maximum r^{th} closest distance is the *post-disruption radius.* Thus, the maximum 1^{st} closest distance is the *non-disruption radius*, the objective of the classic p-center problem.

The RANPCP can be mathematically stated as follows. Let $\mathcal{N}$ be a set of points, $\mathcal{I} \subseteq \mathcal{N}$ be a set of potential facility locations and $\mathcal{J} \subseteq \mathcal{N}$ be a set of demand points. Let d_{ij} be the desirability of serving demand point $j \in \mathcal{J}$ with facility $i \in \mathcal{I}$. Because our solution methods are still valid if they are applied to a problem instance whose distances do not obey the triangle inequality, we could let $d_{ij} = h_j d^{'}_{ij}$, where $d^{'}_{ij}$ is the distance from i to j and h_j is the weight of demand point j. For simplicity, in this paper we refer to d_{ij} as the distance from i to j. Let $\mathcal{X} \subseteq \mathcal{I}$ be a set of located facilities and let $D^r_j(\mathcal{X})$ be the distance from demand point j to its r^{th} closest located facility when the facilities $\mathcal{X}$ are located. The RANPCP requires that $|\mathcal{X}| \leq p$, the number of facilities that may be located. The RANPCP can be stated as:

$$\min_{\substack{\mathcal{X} \subseteq \mathcal{I} \\ |\mathcal{X}| \leq p}} \max_{j \in \mathcal{J}} D^r_j(\mathcal{X}) \tag{1}$$

Drezner (1987) modeled the situation in which an interdictor seeks to destroy r facilities in order to maximize the maximum post-interdiction distance from a demand point to its closest available facility. He called this model the (p, r)-center problem. He noticed that the interdictor's optimal strategy is to choose a demand point and interdict the r closest located facilities to that demand point. Thus, the (p, r)-center problem is equivalent to the $(r + 1)$-all-neighbor p-center problem.

In an optimal solution to the RANPCP, each demand point is covered by at least r facilities, meaning that each demand point is within U^* distance units of r facilities, where U^* is the optimal maximum distance. Thus, the parameter r can represent either the number of covers or the number of neighbors required by each demand point.

The RANPCP model relates to several concepts in risk assessment. First, the consequence modeled in the RANPCP is the increase in the maximum distance from a demand point to its closest facility when $(r-1)$ facility disruptions have occurred. The RANPCP does not consider the likelihood of a facility disruption event; rather, it models the situation in which a facility disruption event has occurred. Further, in the RANPCP model the vulnerability of facilities is complete. That is, if a facility is affected by an event such as a natural disaster or attack, the facility is completely inoperable. Thus, the objective of the RANPCP is to minimize the worst case consequence.

In the rest of this section we describe a simple MIP formulation for the RANPCP.

2.1. Multiple-Assignment Formulation

The MIP formulation we present is a simple modification of the MIP formulation for the classic p-center problem (see Daskin (1995)). In this classic formulation, assignment variables, X_{ij}, take a value of 1 if facility i is the closest located facility to demand point j. The following constraints are included to ensure that every demand point is assigned to exactly one facility.

$$\sum_{i \in \mathcal{I}} X_{ij} = 1 \quad \forall j \in \mathcal{J} \tag{2}$$

A formulation for the RANPCP can be obtained by assigning each demand point to multiple facilities, which explicitly models backup assignments. In particular, each demand point is assigned to r facilities. Thus, the distance from a demand point j to its r^{th} closest facility is simply $\max_{i \in \mathcal{I}_j}\{d_{ij}\}$, where $\mathcal{I}_j$ is the set of r facilities that are assigned to j. Therefore, the maximum r^{th} closest distance is $\max_{j \in \mathcal{J}} \max_{i \in \mathcal{I}_j}\{d_{ij}\}$. Because each demand point is assigned to multiple facilities, this formulation can be called the multiple-assignment (MA) formulation. To allow for multiple assignments, X_{ij} is redefined as a binary variable that is 1 if demand point j is assigned to facility i at some level and 0 otherwise. In addition, the variable Y_i is 1 if a facility is located at i and 0 otherwise. The MA formulation is as follows:

$$
\begin{aligned}
\text{(MA)} \quad \min \quad & U && \text{(3a)}\\
\text{s.t.} \quad & d_{ij}X_{ij} \leq U \quad \forall i \in \mathcal{I}, j \in \mathcal{J} && \text{(3b)}\\
& \sum_{i\in\mathcal{I}} X_{ij} = r \quad \forall j \in \mathcal{J} && \text{(3c)}\\
& X_{ij} \leq Y_i \quad \forall i \in \mathcal{I}, j \in \mathcal{J} && \text{(3d)}\\
& \sum_{i\in\mathcal{I}} Y_i \leq p && \text{(3e)}\\
& Y_i \in \{0,1\} \quad \forall i \in \mathcal{I} && \text{(3f)}\\
& X_{ij} \in \{0,1\} \quad \forall i \in \mathcal{I}, j \in \mathcal{J} && \text{(3g)}
\end{aligned}
$$

The objective (3a) and Constraints (3b) ensure that the objective value is equal to the maximum value of the weighted distance between demand points and their r^{th} closest located facility, over all demand points. Constraints (3c) require that every demand point be assigned to r facilities. In conjunction with the minimization objective, Constraints (3b) and (3c) jointly require each demand point to be assigned to its r closest located facilities. Constraints (3d) only allow assignments to be made to located facilities and Constraint (3e) limits the number of facilities located. Constraints (3f) and (3g) require the decision variables to be binary.

2.2. Other Models

We also investigated several other models, which are given in Appendix A. However, the (MA) formulation outperformed all of them in a set of preliminary experiments.

2.3. Binary Search Algorithm

As an alternative to solving the MA formulation using branch-and-bound, we can also solve the RANPCP using a binary search algorithm similar to the one used to solve the p-center problem (Daskin, 1995). The binary search algorithm is an attractive alternative because 1) it solves the RANPCP faster than the MIP formulations (see Section 1) and 2) it can identify the presence of a special property called saturation (see Section 4.1.4).

The binary search algorithm for the p-center problem solves a series of set cover problems (SCPs) to find the optimal maximum distance. Our binary search algorithm for the RANPCP uses the multi-set-cover location problem (MSCLP) (Church and Gerrard, 2003) in place of the SCPs. The multi-set-cover location problem modifies the SCP because it requires that each demand point be covered by at least ℓ facilities, rather than 1. The MSCLP is formulated as the following MIP:

$$
\begin{aligned}
(\text{MSCLP}(\delta)) \quad \min \quad & \sum_{i\in\mathcal{I}} Y_i && \text{(4a)}\\
\text{s.t.} \quad & \sum_{i\in\{i:d_{ij}\leq\delta\}} Y_i \geq r \quad \forall j \in \mathcal{J} && \text{(4b)}\\
& Y_i \in \{0,1\} \quad \forall i \in \mathcal{I} && \text{(4c)}
\end{aligned}
$$

The RANPCP can be solved by using the binary search algorithm described in Algorithm 1.

Algorithm 1 Binary search algorithm for RANPCP.

1: **function** BinarySearch
2: Let $D = \{D_1, \ldots, D_{|\mathcal{I}|\times|J|}\}$ be the set of all distances, $\{d_{ij}\}_{i\in\mathcal{I}, j\in\mathcal{J}}$, arranged in increasing order.
3: $lbIndex \leftarrow 0$; $ubIndex \leftarrow |D| - 1$
4: **while** $lbIndex \neq ubIndex$ **do**
5: Set $index = lbIndex + \left\lceil \frac{ubIndex - lbIndex}{2} \right\rceil$
6: Obtain a heuristic solution to MSCLP(D_{index}), $\bar{Y}$. Let $\hat{Y} = \{i \in \mathcal{I} | \bar{Y}_i = 1\}$. ▷ Optional
7: **if** $|\hat{Y}| \leq p$ **then** $ubIndex \leftarrow index$; go to Line 5. ▷ Optional
8: Build RANPCP solution using $\hat{Y}$. Let $\hat{i}$ be the index of its post-disruption radius. ▷ Optional
9: **if** $D_{\hat{i}} < D_{ubIndex}$ **then** $ubIndex \leftarrow \hat{i}$ go to Line 5. ▷ Optional
10: Solve MSCLP(D_{index}) to optimality, obtaining solution Y^*.
11: **if** $|Y^*| > p$ **then** $lbIndex \leftarrow index + 1$.
12: **else** $ubIndex \leftarrow index$.
13: **return** Y^*

Lines 6–9 are optional steps added to speed up the algorithm by reducing the number of times that MSCLP must be solved to optimality. Step 6 of Algorithm 1 involves finding a heuristic solution to the MSCLP. One way to find such a solution is by using a heuristic algorithm for the set cover location problem (Balas and Ho, 1980), modified here for the MSCLP. First, a demand point is said to be single covered if there is at least one facility within D_{index}. A demand point is multi-covered if there are at least r facilities within distance D_{index}. Let n_i be the number of facilities that can single-cover demand point i within distance D_{index}. Proceeding through the list of demand points by increasing order of n_i, cover a demand point i by locating the facility that single-covers the maximum number of un-multi-covered demand points. Continue until all of the demand points are multi-covered. Then remove all redundant facilities, i.e., facilities for which all demand points are multi-covered after even if the facility is removed.

The method for building a heuristic solution to the RANPCP in step 8 is as follows.

1. If $|\hat{Y}| > p$, remove the facility whose removal minimizes the increase in the RANPCP objective.
2. Repeat step 1 until $|\hat{Y}| \leq p$.
3. Return the modified set $\hat{Y}$ as the RANPCP heuristic solution.

2.3.1. Bounds

We computed lower and upper bounds before using the binary search algorithm in order to reduce the set of distances over which the binary search algorithm searches. These lower and upper bounds can be added to the binary search algorithm by removing all values in D that are outside the bounds.

A simple lower bound can be obtained by locating the closest r facilities to every demand point:

$$LB_0 = \max_{j\in\mathcal{J}} \{d_{i_j^r j}\}$$

Note that when $r = 1$, this lower bound is zero.

Elloumi et al. (2004) described another lower bound for the PCP and here we describe how to modify their lower bound for the RANPCP. First, let $i'_j(r, i)$ be the r^{th} closest location to demand point j, not including location i and let $\gamma_i = \max_{j\in\mathcal{J}} d_{i'_j(r,i),j}$. Next, sort the γ_i values in increasing order $\gamma_{i_1} \leq \gamma_{i_2} \leq \cdots \leq \gamma_{j_{|\mathcal{I}|}}$. Then, $LB_1 = \gamma_{|\mathcal{I}|-p}$. Note that LB_1 is not always zero for $r = 1$. In our experimentation, we used $\max\{LB_0, LB_1\}$ as the lower bound.

Elloumi et al. (2004) also described two upper bounds for the PCP but these bounds cannot be directly extended to the RANPCP because the RANPCP requires each demand point to be covered r times while the PCP only requires each demand point to be covered once. A simple upper bound for the RANPCP can be obtained by assuming that the p furthest facilities to every demand point are located. In this case, the r^{th} closest located facility to a demand point j will be located at the $(|\mathcal{I}| - p + r)^{\text{th}}$ closest location to j. Thus, an upper bound is:

$$UB_0 = \max_{j \in \mathcal{J}} \{ d_{i_j^{|\mathcal{I}|-p+r} j} \}$$

These bounds can also be used to improve the tractability of the MIP formulations. A lower bound lb may be added to Formulation MA by adding the constraint $lb \leq V$.

2.3.2. Greedy Heuristic

One way to find an initial upper bound for the binary search algorithm is to find a good initial feasible solution. The following greedy heuristic, which is a modification of a heuristic by Mladenović et al. (2003) for the p-center problem, can be used to find an initial feasible solution and corresponding upper bound, UB_1.

1. Solve the 1-center problem ($\arg\min_{i \in \mathcal{I}} \max_{j \in \mathcal{J}} d_{ij}$) and place r facilities at the 1-center.
2. Remove a facility from the 1-center and place it at the node that minimizes the resulting objective increase; repeat until only one facility is located at the 1-center.
3. Let the set $\mathcal{I}'$ be the set of locations that do not have a facility and let $\Delta(i)$ be the objective function decrease associated with locating a facility at i. Locate a facility at $i' \in \arg\min_{i \in \mathcal{I}'} \Delta(i)$. Repeat until all p facilities have been located.

3. Bi-objective Problem: Generating Pareto-Efficient Sets

A drawback of the RANPCP model is that it only optimizes one objective: the post-disruption radius. However, the non-disruption radius is likely to be a concern of most decision-makers because it represents the operational cost without disruptions. In addition, a decision-maker might also like to include the number of facilities (p), which represents the system design cost, as an objective. Since these objectives are conflicting, it is more appropriate to present a set of solutions and let the decision-maker choose a single solution from the set. One such set of solutions is the set of Pareto-efficient points.

A Pareto efficient set for two objectives can be described as follows. If α and β are two objectives of interest to a decision maker, the set $T = \{(\alpha, \beta)\}$ may represent the set of all possible pairs of objective values. Assuming T is countable, the k^{th} point in the Pareto efficient set can be represented by (α_k, β_k). Point k_1 is said to dominate point k_2 if point k_1 is better than point k_2 in one objective and point k_1 is no worse than point k_2 in the other objective. A point that is not dominated by any other point is called a Pareto optimal point. A Pareto-efficient set, denoted here as $S \subseteq T$, is the set of all Pareto optimal points. The Pareto-efficient set for three objectives can be described in a similar manner.

Since the problems studied in this paper are discrete, the Pareto-efficient set is a set of discrete points, as shown in Figure 1. The black points represent Pareto-efficient points and the dashed lines are displayed to show that the α-objective stays constant as the β-objective decreases.

In general, the problem of finding all Pareto-efficient solutions is difficult. When α and β are convex functions of the decision variables, a weighted-sum approach can be used to generate the Pareto-efficient set.

However, the RANPCP and its bi-objective derivatives are combinatorial problems, and thus non-convex. Fortunately, the efficiency of the binary search algorithm presented in Section 2.3 facilitates the efficient generation of the set S for various combinations of objectives, as described in the following sections.

Figure 1: Pareto efficient set.

3.1. Max Closest Distance Vs. Max r^{th} Closest Distance

An efficient method for computing the Pareto-efficient set for these two objectives is to use Algorithm 1 to alternately solve for one objective with a constraint on the other objective, which we call the alternate binary-search (ABS) method. ABS similar to the ϵ-constraint approach for multi-objective combinatorial optimization problems described by Berube et al. (2009), except that ABS uses binary search to solve the single-objective problems rather than branch-and-cut.

Let RANPCP$(\cdot, \delta^{(r)})$ denote the problem of minimizing the non-disruption radius subject to a constraint that requiring the post-disruption radius to be no greater than $\delta^{(r)}$. Further, let RANPCP$(\delta^{(1)}, \cdot)$ denote the problem of minimizing the post-disruption radius subject to a constraint requiring the non-disruption radius to be no greater than $\delta^{(1)}$. The problems RANPCP$(\cdot, \delta^{(r)})$ and RANPCP$(\delta^{(1)}, \cdot)$ can both be solved using Algorithm 1 with a modified auxiliary problem. In particular, the MSCLP$(\delta^{(r)})$ auxiliary problem is modified to account for both post-disruption radius and non-disruption radius, forming the distance-constrained MSCLP (DC-MSCLP):

$$(\text{DC-MSCLP}(\delta^{(1)}, \delta^{(r)})) \quad \min \sum_{i \in \mathcal{I}} Y_i \tag{5a}$$

$$\text{s.t.} \quad \sum_{i \in \{i : d_{ij} \leq \delta^{(1)}\}} Y_i \geq 1 \quad \forall j \in \mathcal{J}, \tag{5b}$$

$$\sum_{i \in \{i : d_{ij} \leq \delta^{(r)}\}} Y_i \geq r \quad \forall j \in \mathcal{J}, \tag{5c}$$

$$Y_i \in \{0, 1\} \quad \forall i \in \mathcal{I}. \tag{5d}$$

The DC-MSCLP$(\delta^{(1)}, \delta^{(r)})$ minimizes the number of facilities located subject to the requirements that i) the r^{th} closest facility to a demand point be within $\delta^{(r)}$ distance units (5c) and ii) the closest facility to a demand point be within $\delta^{(r)}$ distance units (5c).

The DC-MSCLP$(\delta^{(1)}, \delta^{(r)})$ can be used to solve RANPCP$(\cdot, \delta^{(r)})$ by i) fixing $\delta^{(r)}$ in Constraints (5c) and ii) varying $\delta^{(1)}$ in Constraints (5b) within in a binary search algorithm (see Algorithm 1) in order to find the optimal non-disruption radius, $\delta^{(1)*}$. The DC-MSCLP$(\delta^{(1)}, \delta^{(r)})$ can also be used to solve RANPCP$(\delta^{(1)}, \cdot)$: i) fix $\delta^{(1)}$ in Constraints (5b) and ii) vary $\delta^{(r)}$ in Constraints (5c) within in a binary search algorithm in order to find the optimal non-disruption radius, $\delta^{(r)*}$.

Using RANPCP$(\cdot, \delta^{(r)})$ and RANPCP$(\delta^{(1)}, \cdot)$ as subproblems, Algorithm 2 efficiently generates the set of Pareto-efficient points. In Step 1, RANPCP$(\cdot, \delta_k^{(r)} - \epsilon)$ is solved to find the non-disruption radius for Pareto-efficient point $(k+1)$. The value $\delta_k^{(r)} - \epsilon$ is used for the post-disruption cover distance to ensure that the optimal non-disruption radius will increase from the previous iteration. In Step 2, RANPCP$(\delta_k^{(1)}, \cdot)$ is solved to obtain Pareto-efficient point k.

Algorithm 2 Constructing the Pareto-efficient set for max closest distance and max r^{th} closest distance objectives.

function ABS
  Let ϵ be a small number.
  Set $k \leftarrow 0$, $\delta_0^{(1)} = 0$, $\delta_0^{(r)} = \max_{ij}\{d_{i_j^r j}\}$, and $S \leftarrow S \cup \{(\delta_0^{(r)}, \min_{ij}\{d_{ij}\})\}$
  while $\delta_k^{(1)} \neq \max_{ij}\{d_{ij}\}$ **do**
    Solve RANPCP$(\cdot, \delta_k^{(r)} - \epsilon)$ to obtain min. non-disruption radius $\delta_k^{(1)}$ ▷ Step 1
    Solve RANPCP$(\delta_k^{(1)}, \cdot)$ to obtain min. post-disruption radius $\delta_{k+1}^{(r)}$ ▷ Step 2
    Set $S \leftarrow S \bigcup \{(\delta_{k+1}^{(r)}, \delta_k^{(1)})\}$
    $k \leftarrow k + 1$
  return S

3.2. Number of Facilities Located Vs. Maximum r^{th} Closest Distance

The optimal max r^{th} closest distance depends on the value of p because each additional facility adds more redundancy to the system. However, in most facility location problems there is a diminishing return on adding more facilities. Thus, it is useful for a decision-maker to understand the decrease in the max r^{th} closest distance that results from adding more facilities. One way to display this diminishing return is by constructing the Pareto-efficient set between the number of facilities located and the max r^{th} closest distance. This set can be constructed using Algorithm 3, which is in the same spirit as Algorithm 2. Algorithm 3 differs from Algorithm 2 in that it maintains upper bounds for the RANPCP in Step 1. Also, Step 2 consists of merely incrementing the design cost, rather than solving an optimization problem.

Algorithm 3 Constructing the Pareto-efficient set for design cost and max r^{th} closest distance objectives.

function ABS-CostVsRthClosestDistance
  RANPCP(p):= RANPCP problem with a budget of p facility locations
  $\bar{\delta}(p)$:= incumbent upper bound for RANPCP(p)
  Set $k \leftarrow 0$ and set $\bar{\delta}(p) = \infty$ for $p = 1$ to $|\mathcal{I}|$
  Solve MSCLP$(\max_{ij}\{d_{ij}\})$ to obtain p_0.
  while $p_k \leq |\mathcal{I}|$ **do**
    Set $\min\{UB_0, UB_1, \bar{\delta}(p_k)\}$ as the initial UB to RANPCP(p_k) ▷ Step 1a
    Solve RANPCP(p_k) using Algorithm 1 to obtain min. post-disruption radius $\delta_k^{(r)}$ ▷ Step 1b
      (During the execution of the Algorithm 1, continually update the function $\bar{\delta}(\cdot)$.)
    Set $S \leftarrow S \bigcup \{(\delta_k^{(r)}, p_k)\}$
    Set $p_k \leftarrow p_k + 1$
    $k \leftarrow k + 1$
  return S

Step 1 of Algorithm 3 minimizes the post-disruption radius subject to a restriction on the number of facilities located. Step 1 requires the upper bounds, $\bar{\delta}(p)$, to be updated during the execution of Algorithm 1. The upper bounds can be updated by replacing Step 4 of Algorithm 1 with the following:

Step 4 If $|Y^*| > p_k$, set $lbIndex = index + 1$ and set $\bar{\delta}(|Y^*|) \leftarrow \min\{\bar{\delta}(|Y^*|), D_{index}\}$. Otherwise, set $ubIndex = index$. Return to step 2.

4. Numerical Experimentation

This section describes experiments performed on the single- and bi-objective RANPCP. All experiments were run on a 64-bit 2.66GHz AMD processor running the Linux operating system with 16GB of memory. All MIP formulations, including the multi-set-cover location problem, were solved with CPLEX v12.1.

Before solving an instance, we first found a lower bound $LB = \max\{LB_0, LB_1\}$, an upper bound $UB = \min\{UB_0, UB_1\}$, and a feasible solution produced by the greedy heuristic in Section 2.3.2. For Formulations MA we used the upper bound to eliminate variables (see Section 2.3.1) and seeded the branch and bound algorithm with an initial feasible solution. For the binary search (Algorithm 1) and PC-SC, we used the upper and lower bounds as the initial upper and lower bounds for the algorithm.

We tested our solution methods on 12 geographically-motivated datasets from the facility location literature (see Appendix B).

4.1. Single-objective Problem

First, we study the single-objective version of the RANPCP to gain computational and decision-making insights.

4.1.1. Comparison of Run Times

In this Section the computational performance of Formulation MA is compared to the p-center set-covering formulation (PC-SC) from Elloumi et al. (2004) and the binary search algorithm (BS).

Table 1 shows the computational results for using CPLEX branch-and-bound to solve several instances of the RANPCP. Each row contains the run time from solving an instance of the RANPCP using both formulations. Each cell in the table contains the time required to solve the problem to optimality. The word "time" in a cell of the table indicates that the instance was not solved within a time limit of 72 hours; the word "memory" indicates that the CPLEX branch-and-bound algorithm ran out of memory.

Table 1: MIP results for various datasets

				Run time (s)		
No.	Dataset	p	r	PC-SC (Elloumi et al., 2004)	Form. MA	BS
1	d88	5	1	211	17	<1
2	d88	5	2	3506	74	<1
3	d88	10	1	3199	18	<1
4	d88	10	2	57	30	<1
5	d88	27	3	33	5	<1
6	d88	27	6	34	15	<1
7	d88	27	9	55	23	<1
8	d150	5	1	time	531	<1
9	d150	5	2	time	2344	<1
10	d150	10	1	time	569	<1
11	d150	10	2	memory	1762	<1
12	d150	45	5	time	81	<1
13	d150	45	9	time	100	<1
14	d150	45	14	time	145	<1

Table 1 shows that Formulation MA solved the RANPCP faster than PC-SC in all of the instances. This is likely due to the fact that while the MA and PS-SC formulations have the same number of variables, the MA formulation has $\mathcal{O}(|\mathcal{I}| \times |J|)$ constraints while the PS-SC formulation has $\mathcal{O}(|\mathcal{I}|^2|J|)$ constraints. Our results for $r = 1$ contradict the findings of Elloumi et al. (2004), who found that PC-SC outperformed Formulation MA when $r = 1$. However, it is difficult to make a fair comparison because they used CPLEX v7.1 and we used version 12.1.

The table also shows that binary search (Algorithm 1) required much less computation time than both MIP formulations.

4.1.2. Scalability of Binary Search Algorithm

In this section we examine how much the computational performance (e.g., run time, number of iterations) of Algorithm 1 is affected by changing problem parameters such as the number of locations, the number of facilities, and the number of neighbors.

Table 2 shows that datasets with more nodes usually require more computation time to solve. Each row of the table shows summary statistics for a set of instances (varying p and r) for a dataset. The table also shows that the number of nodes does not significantly influence the number of iterations. Algorithm 1 runs for $\log_2(|\mathcal{I}|^2 + 1)$ iterations in the worst case (Elloumi et al., 2004). However, the upper and lower bounds for the RANPCP and the heuristic for the MSCLP often reduce the number of iterations.

Table 2: Summary statistics for runtime and number of iterations for all instances of each dataset

	Total computation time (s)				Number of iterations			
Dataset	min	max	avg.	var.	min	max	avg.	var.
sw55	0	0.1	0.02	<0.01	0	10	4.9	19.52
lor100	0	0.26	0.06	<0.01	0	14	6.4	45.49
lon150	0.02	4.4	0.91	1.1	0	15	13	11.44
lor200	0.02	2.9	0.56	0.7	0	16	8.8	43.59
lor300a	0.04	17	1.8	16.2	0	17	6.5	51.15
lor402a	0.1	70	6.3	253	0	18	8.1	53.05
lor818	7.6	5454	861	1690336	11	19	16	5.49

Because the number of iterations does not change significantly with an increase the the number of nodes, we conclude that the time per iteration increases as the number of nodes increase. Therefore, the increased run time is due to the time required to solve a larger set cover problem at each iteration.

We also examined the affect of $|\mathcal{I}|$, p, and r on the runtime and found that the runtime increased with $|\mathcal{I}|$. However, we did not find that p or r had much affect.

4.1.3. Sensitivity of Number of Neighbors, r

Because a decision-maker may be unsure about the true value of r, the number of covers required, it is useful for them to understand the sensitivity of the optimal solution values to the choice of r. To help the decision-maker understand the sensitivity, we measured the relative error in the optimal objective value caused by an incorrect choice of r. The following notation is used to quantify this relative error. First, let $f_{(r)}(Y)$ be the maximum distance from a demand point to its r^{th} closest facility given the set of located facilities represented by the solution variable Y. Let $Y^*_{(r)}$ be the optimal solution to the r-all-neighbor p-center problem. Finally, let $\eta_{r'r}$ denote the relative objective function error that occurs when a decision-maker models the number of covers required as equal to r' when it is truly equal to r, which is calculated as

$$\eta_{r'r} = \frac{f_{(r)}(Y^*_{(r')}) - f_{(r)}(Y^*_{(r)})}{f_{(r)}(Y^*_{(r)})}. \tag{6}$$

Table 3 displays the value of $\eta_{r'r}$ for several values of r' and r for the d49 and lon150 datasets. Two main observations can be made from this table. (1) The results indicate that the RANPCP is sensitive to error in the value of r. Excluding the experiments where $\eta_{r'r} = 0$ (i.e., $r' = r$), the relative objective function increase, $\eta_{r'r}$, ranges from 0.15 to 3.5 with an average of 1.06. Thus, on average, the objective function is doubled when the value of the number of neighbors is inaccurate. (2) Surprisingly, the relative objective function increase, $\eta_{r'r}$, does not always increase with the magnitude of error, $|r' - r|$. Thus, a decision-maker is not guaranteed to obtain higher quality solutions by spending more to improve the estimate of r.

Table 3: Relative objective function error, $\eta_{r'r}$, for incorrect value of r

(a) d49 dataset

		r (Actual)			
		1	2	3	4
r' (Perceived)	1	0	1.89	0.98	1.85
	2	1.10	0	2.01	1.85
	3	3.50	0.53	0	1.61
	4	0.90	1.44	0.15	0
Average excluding zeros		1.48			

(b) lon150 dataset

		r (Actual)			
		1	2	3	4
r' (Perceived)	1	0	0.59	0.68	0.31
	2	0.63	0	0.51	0.34
	3	1.11	0.35	0	0.45
	4	1.7	0.67	0.27	0
Average excluding zeros		0.63			

4.1.4. Saturation Point

In this section we investigate a property of the RANPCP called *saturation*, which is an artifact of using the maximum distance measure along with the worst-case risk measure. An instance of the RANPCP is saturated if the r closest facilities to a given demand point are located and the distance from that demand point and its r^{th} closest located facility is equal to the optimal objective value. When an instance is saturated for a given value of p and r, locating additional facilities does not improve the objective.

The specific analysis that we present in this section is an analysis of the point at which datasets become saturated for a value of r. Let $p^*(r)$ be the *saturation point* for a dataset with r neighbors. In other words, $p^*(r)$ is the smallest value of p such that the instance of the RANPCP with a given number of facilities p and number of neighbors r is saturated.

A modeling implication of saturation is that it is possible to allow p facilities to be located in the RANPCP when in fact only $p^*(r)$ are needed to minimize the max r^{th} closest distance. A MIP formulation is unlikely to identify the saturation phenomenon unless the number of facilities located is included as an additional objective function term. However, since a binary search algorithm such as Algorithm 1 minimizes the number of facilities in each iteration (e.g., by solving MSCLP), saturation can be identified.

Figure 2 shows the saturation point vs. r for several datasets. For an instance with a given value of r, the saturation point can be found by setting $p = |\mathcal{I}|$ and using Algorithm 1. When the algorithm terminates, record the number of facilities located in the optimal solution. This is the saturation point. In the Figure, the saturation point is always greater than r. This implies that $p^*(r) \geq r$, which must be true because a saturated solution has at least r located facilities. The figure also illustrates that the saturation point is not monotonic with respect to r.

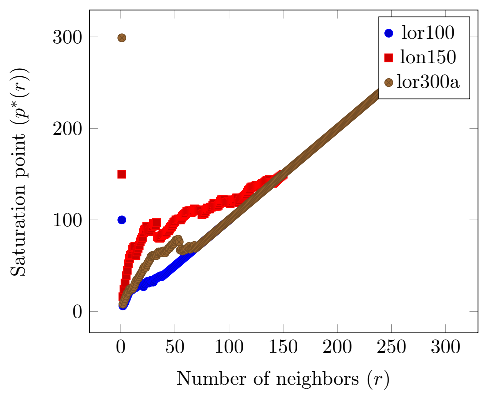


Figure 2: Saturation point vs. r for several datasets

Figure 2 also shows that the saturation curves eventually approach a point where the saturation point equals r. Let the *super-saturation point*, r_0, be a value of r such that $p^*(r) = r$ and $p^*(r_0 + n) = r_0 + n$ for $n = 1, \ldots, |\mathcal{I}| - r_0$. We observed the following about the super-saturation point:

1. Once the super-saturation point is reached, the solutions often become nested. In other words, the solution for $p^*(r_0 + n)$ is a subset of the solution for $p^*(r_0 + n + 1)$ for $n = 0, \ldots, |\mathcal{I}| - r_0 - 1$.
2. Objective values are not always the same for $p^*(r_0 + n)$ for $n = 1, \ldots, |\mathcal{I}| - r_0$. Consequently, the bottleneck pair (the facility i and demand point j for which d_{ij} equals the optimal objective) are also different for different values of r.

Table 4 contains the super-saturation point for several instances. The table shows that the ratio $\frac{r^*}{|\mathcal{I}|}$ varies across datasets. Also, all of the unweighted datasets had a super saturation point equal to $|\mathcal{I}|$. Thus, the saturation point is clearly influenced by the demand point weights.

Table 4: Super-saturation point for various datasets

	Wtd.		Unwtd.		Wtd.-Unwtd.	
Dataset	r_0	$\frac{r_0}{\lvert\mathcal{I}\rvert}$	r_0	$\frac{r_0}{\lvert\mathcal{I}\rvert}$	r_0	$\frac{r_0}{\lvert\mathcal{I}\rvert}$
sw55	22	0.45	49	1	-27	-0.55
lor100	38	0.38	100	1	-62	-0.62
lon150	140	0.93	150	1	-10	-0.07
lor200	67	0.34	200	1	-133	-0.66
lor300a	69	0.23	300	1	-231	-0.77
lor300b	69	0.23	300	1	-231	-0.77
Min	38	0.23	55	1	-231	-0.77
Max	140	0.93	300	1	-9	-0.07
Average	72	0.49	184	1	-113	-0.57

The appendix contains theoretical results related to saturation.

4.2. Bi-objective Problem

Next, we examine two bi-objective versions of the RANPCP: 1) jointly minimize both the non- and post-disruption radii and 2) jointly minimize both the number of facilities located and the post-disruption radius.

4.2.1. Scalability of Bi-objective algorithms

In Section 3, algorithms for generating the set of Pareto-efficient points for two objectives were described. For these algorithms to be useful in practice, they need to be able to facilitate scenario analysis; that is, the run time of the algorithm should be short enough to allow the decision maker to experiment with different scenarios and receive feedback within a reasonable amount of time (e.g., about 24 hours). Fortunately, Algorithms 2–3 meet the said requirements for trial-and-error analysis.

Table 5 displays the run times for computing the set of Pareto-efficient points for the non-disruption radius and post-disruption radius objectives (Algorithm 2) for several problem instances. As the results show, Algorithm 2 generates the set of Pareto-efficient points quickly for the smaller datasets. The run times for the lor818 dataset are much higher than for the other datasets, which is not surprising because the lor818 instances required much longer run times for single-objective RANPCP (see Table 2).

Table 5: Run time for max distance vs. max r^{th} distance algorithm (Algorithm 2)

Dataset	Budget, p	r	Run time (s)
d49	10	1	0.3
d49	10	2	1.0
d49	15	3	0.6
d49	15	5	0.7
d150	15	2	112
d150	15	3	17
d150	30	3	22
d150	30	6	32
lor818	164	33	9066
lor818	164	50	11857
lor818	245	25	12548
lor818	245	50	9660

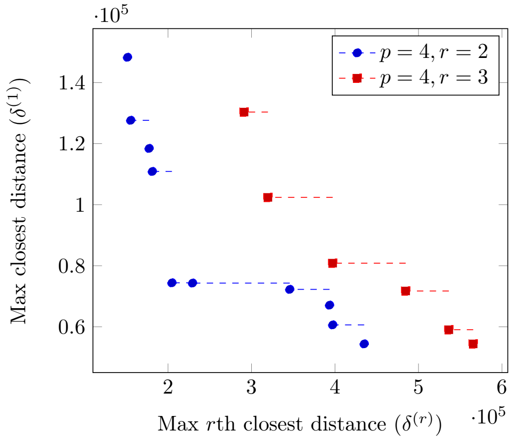
Table 6 displays the run times for computing the set of Pareto-efficient points for the cost and post-disruption radius objectives (Algorithm 3) for several problem instances. As in Table 5, the results show that Algorithm 2 generates the set of Pareto-efficient points quickly, although the run times for the lor818 dataset are much higher than for the other datasets.

Table 6: Run time for design cost vs. max r^{th} distance algorithm (Algorithm 3)

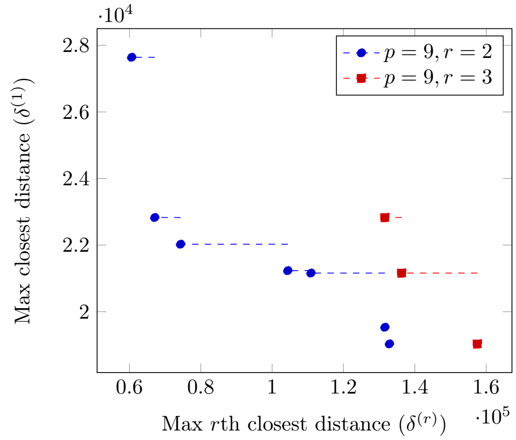
Dataset	r	computation time (s)
d49	2	1
d49	3	1
d150	5	83
d150	9	48
lor402a	5	7
lor402a	9	9
lor818	33	16672
lor818	50	13283

4.2.2. Max Closest Distance Vs. Max r^{th} Closest Distance

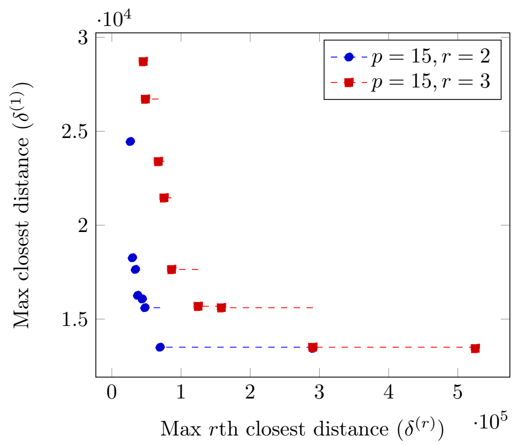
Figure 3 shows the complete Pareto-efficient set for the max closest distance and max r^{th} closest distance for several instances of the d49 and d150 datasets. Each of the sets were generated using Algorithm 2.



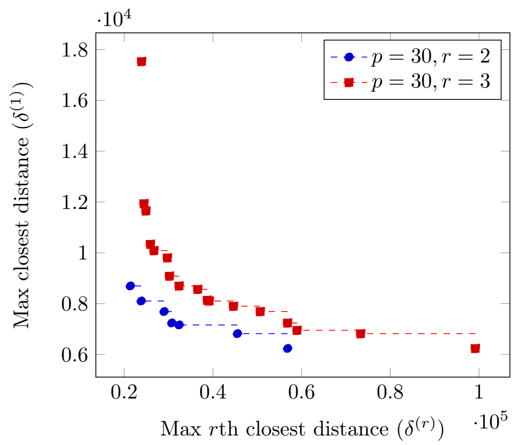
(a) Daskin 49-node dataset (d49), $p = 4$



(b) Daskin 49-node dataset (d49), $p = 9$



(c) Daskin 150-node dataset (d150), $p = 15$



(d) Daskin 150-node dataset (d150), $p = 30$

Figure 3: Max closest distance vs. max r^{th} closest distance for several datasets

Figure 3 produces two main insights. First, for many of the Pareto-efficient sets, there is significant horizontal and vertical separation between the points. This separation indicates that the algorithm for obtaining the Pareto-efficient presents more valuable information to a decision-maker than an algorithm that only optimizes a single objective in isolation. This value comes from the fact that a single-objective solution may not be Pareto-efficient. For example, if an objective is minimized in isolation, a solution could be obtained that lies on the dashed line to the right of one of the points, i.e., a non-Pareto-efficient solution. This solution is undesirable because the max r^{th} closest distance objective can still be decreased significantly without any increase in the max closest distance! Second, the curves show that if a decision-maker wishes to minimize one of the objectives by itself, the other objective will be far from its minimum value. This fact can be observed by examining the two endpoints of each of the Pareto-efficient sets. For example, for the d49 dataset with $p = 4$ and $r = 2$, the max closest distance value of the leftmost point of the Pareto set is about three times higher than the minimum max closest distance value. Because of this fact, a decision-maker may wish to choose one of the Pareto-efficient solutions that is a compromise between the two objectives.

Figure 3 demonstrates graphically that if one of the objectives is minimized in isolation, the other objective value can be much higher than optimal. In the remainder of this section, this objective function increase is demonstrated numerically. Formally, if the RANPCP model is used to optimize the max r^{th}

closest distance, the resulting solution may have a max closest distance that is much higher than the max distance of a solution optimized for max distance. This is a problem because the max distance without disruptions is usually a primary objective and potential consequence a secondary objective. We measure this *relative objective function increase* for two cases: 1) the RANPCP model is used to minimize the max r^{th} closest distance and 2) the p-center model is used to minimize the max closest distance.

We use the following notation for our relative objective function increase analysis. Define the *max closest distance objective* as the objective of minimizing the distance to from a demand point to its closest located facility. Let the RANPCP objective with r neighbors required be called the *max r^{th} closest distance* objective. For a given instance, let $Y^*_{(1)}$ be the optimal facility configuration for the max closest distance objective and let $Y^*_{(r)}$ be the optimal facility configuration for the max r^{th} closest distance objective. The functions $f_{(1)}(Y)$ and $f_{(r)}(Y)$ are the max closest distance and max r^{th} closest objective values for a location configuration Y. Let the *relative objective function increase for not considering WS-WD* be $\eta_{r,1} = \frac{f_{(1)}(Y^*_{(r)})-f_{(1)}(Y^*_{(1)})}{f_{(1)}(Y^*_{(1)})}$ and the *relative objective function increase for not considering PD-WS* be $\eta_{1,r} = \frac{f_{(r)}(Y^*_{(1)})-f_{(r)}(Y^*_{(r)})}{f_{(r)}(Y^*_{(r)})}$.

Table 7 shows summary statistics for $\eta_{r,1}$ and $\eta_{1,r}$ over all of our datasets and instances. The average relative objective function increase for not considering WS-WD is 0.43 while the average relative objective function increase for not considering PD-WS is 6.3. Hence, if only one objective is used, it should be the post-disruption radius. However, since the objectives are conflicting, a bi-objective model is more appropriate.

Table 7: Summary statistics for $\eta_{r,1}$ and $\eta_{1,r}$ for all instances of each dataset

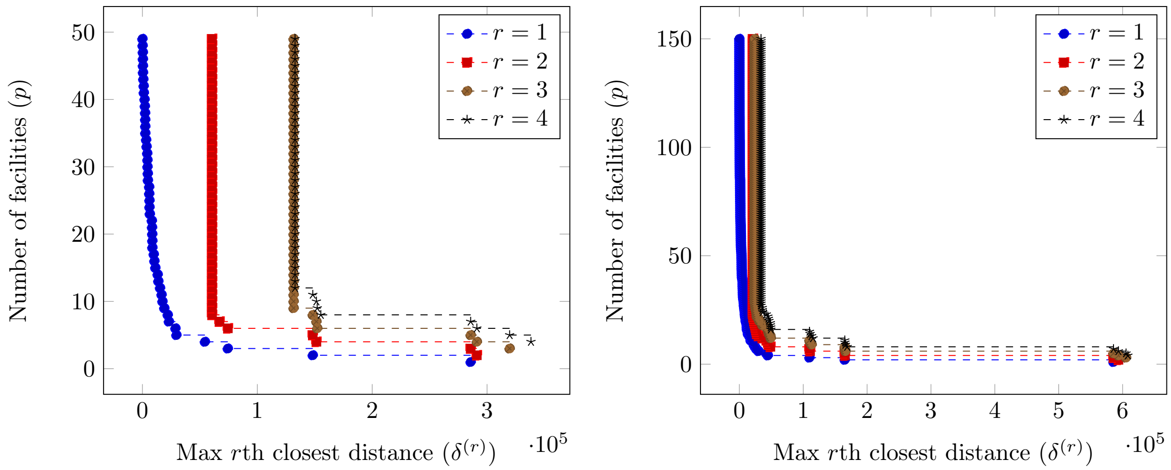
	$\eta_{r,1}$				$\eta_{1,r}$				$\eta_{1,r}/\eta_{r,1}$			
	min	max	avg.	var.	min	max	avg.	var.	min	max	avg.	var.
sw55	0.18	0.96	0.51	0.08	0.32	1.70	1.10	0.25	0.21	1.00	0.53	0.05
lor100	0.00	0.92	0.48	0.08	1.40	9.70	5.90	11.00	0.00	0.37	0.15	0.02
lon150	0.00	0.69	0.39	0.06	0.59	5.80	2.00	2.17	0.00	0.94	0.33	0.09
lor200	0.00	0.84	0.47	0.07	0.90	21.00	10.00	49.13	0.00	0.58	0.12	0.03
lor300a	0.00	1.50	0.64	0.22	0.81	26.00	13.00	75.18	0.00	1.00	0.16	0.08
lor300b	0.00	1.50	0.64	0.22	0.81	26.00	13.00	75.18	0.00	1.00	0.16	0.08
lor400a	0.00	1.30	0.52	0.19	0.85	37.00	16.00	142.91	0.00	1.30	0.15	0.12
lor400b	0.00	1.30	0.52	0.19	0.85	37.00	16.00	142.91	0.00	1.30	0.15	0.12
lor818	0.28	1.30	0.61	0.15	0.46	11.00	3.90	10.55	0.03	1.30	0.37	0.15
ALL	0.00	1.50	0.43	0.01	0.08	37.00	6.30	2625	0.00	3.90	0.29	0.18

4.2.3. Number of Facilities Located Vs. Maximum r^{th} Closest Distance

Next, we examine the tradeoff between the number of facilities located and the max r^{th} closest distance.

Figure 4 shows the complete Pareto-efficient set for the number of facilities located (p) vs. max r^{th} closest distance for several instances of datasets d49 and d150. Each set was generated using Algorithm 3.2 but not using the bounds described in Sections 2.3.1 and 2.3.2 during the execution of the Algorithm 1.

Figure 4 yields several insights. First, the Pareto-sets have a similar form for different values of r, indicating that the change in the max r^{th} closest distance is relatively insensitive to the choice of r. This insight helps a decision-maker make the decision of whether or not add an additional facility without worrying that the value of r is inaccurate. Second, many portions of the Pareto-efficient set are "flat," meaning that a slight increase in the number of facilities significantly decreases the max r^{th} closest distance. This insight is helpful to decision-makers because it shows that it is desirable to find solutions that are on the left side of the "flat" region, meaning that they achieve a relatively large reduction in the max r^{th} closest distance by locating a relatively small number of facilities. Third, the Pareto sets show that once the max r^{th} closest



(a) Daskin 49-node dataset (d49)

(b) Daskin 150-node dataset (d150)

Figure 4: Number of facilities vs. max r^{th} closest distance

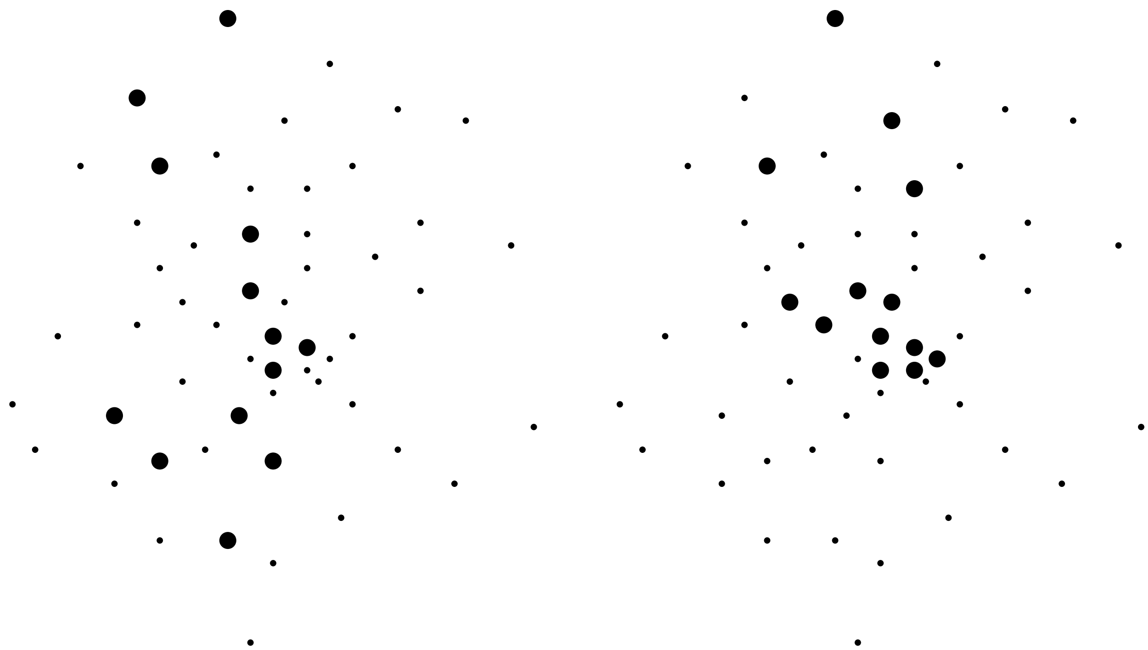
distance reaches a particular value, increases in the number of facilities no longer produce a reduction in the max r^{th} closest distance. This phenomenon is called saturation and is explained in Section 4.1.4.

5. Example

In this section we discuss the implications of the empirical results described in the previous section. We explain these insights through a detailed analysis of the classic 55-node dataset from Swain (1971) (abbreviated sw55). The nodes in this dataset represent districts in the city of Washington, D.C. The nodes each have a weight that is proportion to the population at that node.

In this case study a decision maker wishes to locate ambulances within the districts of the city. The decision maker is especially interested in the response time for emergencies requiring more than one ambulance. Each demand point represents a district and has a weight corresponding to population. The decision is where to locate 13 ambulances within the 55 city districts.

First, consider the solution to the RANPCP with $p = 13$ and $r = 6$, shown in Figure 5a. In this solution the vehicle locations are spread so that every demand point has six vehicles within a reasonably close distance. The maximum response time of the 6^{th} vehicle to an incident at a demand point is 317 time units. The maximum closest distance for this location configuration is 154.



(a) Solution to RANPCP with $r = 3$ and $p = 13$ (b) 13-center solution

Figure 5: Ambulance locations for case study

Next, consider the problem of locating p facilities to minimize the maximum time required for the first vehicle to arrive at a scene. This problem can be solved using the classic p-center model. The solution for $p = 13$ is shown in Figure 5b. In this solution, contrasted with the RANPCP solution in Figure 5a, most of the vehicles are located in the center. This solution is centralized because 1) the demand points in the center have the largest weights in the Swain dataset and 2) because $r = 1$, every demand point only needs to have one facility within a reasonably close distance. For this solution, the maximum response time for one vehicle is 72 and the maximum response time for the 6^{th} vehicle is 741.

Given the application of ambulance response, the relative objective function increases have different interpretations: the relative objective function increase for underestimating the number of vehicles needed, $\eta_{1,r}$, and the relative objective function increase for overestimating the number of vehicles needed, $\eta_{r,1}$.

In this example $\eta_{1,6} = \frac{741-317}{317} = 1.33$. This means that if the 13-center solution is chosen, the response time of the 6^{th} vehicle is 133% higher than if the RANPCP solution had been used. Further, $\eta_{6,1} = \frac{154-72}{72} = 1.13$. This means that if the RANPCP solution is chosen, the response time of the first vehicle is 113% higher than if the p-center solution had been used. Thus, a decision-maker that is concerned about incidents requiring the response of many (~ 6) vehicles would likely prefer the solution shown in Figure 5a over the solution shown in 5b.

6. Conclusions and Future Work

This paper described a study of the r-all-neighbor p-center problem (RANPCP) and makes the following contributions to the literature:

1. We described algorithms for computing the Pareto-efficient set for combinations of two objectives: closest distance vs r^{th}-closest distance and cost vs. r^{th}-closest distance.
2. We performed a series of computational tests of the single- and bi-objective RANPCP that suggest several decision-making insights.

3. We found that the solution times for a simple modification of the standard p-center MIP formulation are better that those of an existing MIP formulation.

Our experiments revealed several insights into the single- and bi-objective versions of the RANPCP. For the single-objective version, experiments showed that the RANPCP model is sensitive to changes in the number of neighbors, r. However, in our experiments the relative objective function error did not depend on the the magnitude of the change in the number of neighbors. We also discovered a structural property of the RANPCP called saturation, the point at which locating additional facilities does not improve the objective function. As we discussed, this property shows a drawback of considering only the post-disruption radius objective in isolation.

For the bi-objective version, we generated Pareto-efficient solutions for the WS-WD and PD-WS objectives. The sets of solutions demonstrated that if a a single objective is minimized in isolation, the value of the other objective will be much higher than optimal. As an alternative to single-objective minimization, the set of Pareto-efficient includes solutions that are a good compromise between the two objectives. We also measured the relative objective function increase associated with optimizing a single objective in isolation. We found that the solutions that are optimal for potential consequence have a WS-WD that is, on average, 43% more than the optimal. However, solutions that are optimal for WS-WD have a potential consequence value that is, on average, 630% more than the optimal. Thus, if only one objective is modeled, it should be the potential consequence objective. We also used our model to analyze the tradeoff between the number of facilities built and the potential consequence. We found that for several instances, significant reductions in potential consequence can be obtained by building a few additional facilities.

6.1. Future Work

Although this work focused on the maximum distance measure, there are several other distance measures that are important to consider in a facility location model with disruptions such as the total (weighted) distance measure. Considering both maximum distance and total distance simultaneously would be valuable because when the maximum distance measure is used, there are usually multiple optimal solutions due to the bottleneck structure. In addition, it would be interesting to analyze the worst-case total distance of solutions produced by the RANPCP.

The models in this paper assumed that the number of neighbors, r, is known with certainty. It may be useful for decision makers to have a model that allows them to place a probability distribution on r in order to minimize the expected loss. This could be used to model the situation where a decision maker is unsure about the amount of resources that an interdictor has. It could also be used when a decision maker is interested in emergency response to different types of incidents, each of which require a different number of vehicles. Alternatively, a robust optimization approach could be used to account for the uncertainty in the value of r. This approach would generate a facility location solution that minimizes the maximum post-disruption radius over all realizations of r.

References

Aksen, D., Aras, N., 2012. A bilevel fixed charge location model for facilities under imminent attack. Computers and Operations Research 39 (7), 1364–1381.

Aksen, D., Aras, N., Piyade, N., 2013. A bilevel P-Median model for the planning and protection of critical facilities. Journal of Heuristics 19 (2), 1–26.

Aksen, D., Şengül Akca, S., Aras, N., 2012. A bilevel partial interdiction problem with capacitated facilities and demand outsourcing. Computers and Operations Research In press.

Alp, O., Erkut, E., Drezner, Z., 2003. An efficient genetic algorithm for the P-Median problem. Annals of Operations Research 122 (1), 21–42.

Balas, E., Ho, A., 1980. Set covering algorithms using cutting planes, heuristics, and subgradient optimization: a computational study. In: Padberg, M. W. (Ed.), Combinatorial Optimization. Vol. 12 of Mathematical Programming Studies. Elsevier North-Holland, The Netherlands, pp. 37–60.

Batta, R., Dolan, J. M., Krishnamurthy, N. N., 1989. The maximal expected covering location problem: Revisited. Transportation Science 23 (4), 277–87.

Berman, O., Krass, D., Menezes, M. B. C., 2007. Facility reliability issues in network P-Median problems: Strategic centralization and Co-Location effects. Operations Research 55 (2), 332–350.

Berube, J. F., Gendreau, M., Potvin, J. Y., 2009. An exact epsilon-constraint method for bi-objective combinatorial optimization problems: application to the traveling salesman problem with profits. European Journal of Operational Research 194 (1), 39–50.

Brotcorne, L., Laporte, G., Semet, F., 2003. Ambulance location and relocation models. European Journal of Operational Research 147 (3), 451–463.

Chaudhuri, S., Garg, N., Ravi, R., 1998. The P-Neighbor K-Center problem. Information Processing Letters 65 (3), 131–134.

Church, R. L., Gerrard, R. A., 2003. The Multi-Level location set covering model. Geographical Analysis 35 (4), 277–290.

Church, R. L., Scaparra, M. P., Middleton, R. S., 2004. Identifying critical infrastructure: The median and covering facility interdiction problems. Annals of the Association of American Geographers 94 (3), 491–502.

Cui, T., Ouyang, Y., Shen, Z.-J. M., 2011. Reliable facility location under the risk of disruptions. Operations Research 58 (4-Part-1), 998–1011.

Current, J., Min, H., Schilling, D., 1990. Multiobjective analysis of facility location decisions. European Journal of Operational Research 49 (3), 295–307.

Daskin, M. S., 1982. Application of an expected covering model to emergency medical service system design. Decision Sciences 13 (3), 416–39.

Daskin, M. S., 1983. A maximum expected covering location model: Formulation, properties and heuristic solution. Transportation Science 17 (1), 48–70.

Daskin, M. S., 1995. Network and Discrete Location: Models, Algorithms, and Applications. John Wiley and Sons, Hoboken, New Jersey.

Daskin, M. S., Stern, E. H., 1981. A hierarchical objective set covering model for emergency medical service vehicle deployment. Transportation Science 15 (2), 137–152.

Drezner, Z., 1987. Heuristic solution methods for two location problems with unreliable facilities. Journal of The Operational Research Society 38 (6), 509–514.

Elloumi, S., Labbé, M., Pochet, Y., 2004. A new formulation and resolution method for the P-Center problem. INFORMS Journal on Computing 16 (1), 84–94.

Hernandez, I., Ramirez-Marquez, J. E., Rainwater, C., Pohl, E., Medal, H., 2013. Robust facility location: Hedging against failures. under second-round review for Reliability Engineering and System Safety.

Khuller, S., Pless, R., Sussmann, Y. J., 2000. Fault tolerant K-Center problems. Theoretical Computer Science 242 (1-2), 237–245.

Krumke, S. O., 1995. On a generalization of the P-Center problem. Information Processing Letters 56 (2), 67–71.

Lee, S.-D., 2001. On solving unreliable planar location problems. Computers and Operations Research 28 (4), 329–44.

Li, Q., Zeng, B., Savachkin, A., 2013. Reliable facility location design under disruptions. Computers and Operations Research 40 (4), 901–909.

Liberatore, F., Scaparra, M. P., Daskin, M., 2011. Analysis of facility protection strategies against uncertain numbers of attacks: The stochastic R-Interdiction median problem with fortification. Computers and Operations Research 38 (1), 357–366.

Lorena, L. A. N., Senne, E. L. F., 2004. A column generation approach to capacitated P-Median problems. Computers and Operations Research 31 (6), 863–876.

Mladenović, N., Labbé, M., Hansen, P., 2003. Solving the p-Center problem with tabu search and variable neighborhood search. Networks 42 (1), 48–64.

O'Hanley, J. R., Church, R. L., 2011. Designing robust coverage networks to hedge against Worst-Case facility losses. European Journal of Operational Research 209 (1), 23–36.

Scaparra, M. P., Church, R. L., 2008. A bilevel Mixed-Integer program for critical infrastructure protection planning. Computers and Operations Research 35 (6), 1905–1923.

Snyder, L. V., Daskin, M. S., 2005. Reliability models for facility location: The expected failure cost case. Transportation Science 39 (3), 400–416.

Swain, R. W., 1971. A decomposition algorithm for a class of facility location problems. Ph.D. thesis, Cornell University.

Van Slyke, R., 1982. Redundant set covering in telecommunications networks. In: Proceedings of the 1982 IEEE International Large Scale Systems Symposium. New York, NY, USA, pp. 217–222.

Zhu, Y., Zheng, Z., Zhang, X., Cai, K., 2013. The r-interdiction median problem with probabilistic protection and its solution algorithm. Computers and Operations Research 40 (1), 451–462.

Appendix A. Other Formulations

In this section we describe several other models for the RANPCP, besides the MA formulation presented in Section 2.1.

Appendix A.1. Three Index Formulation

First, we present a straightforward formulation of the RANPCP. In this formulation we keep track of the corresponding level for each demand-facility pair. The "level" at which a facility is assigned to a demand point is simply the distance rank of that facility in relation to the other located facilities.

Variables.

- Y_i is 1 if a facility is located at i and 0 otherwise.
- $X_{ij\ell}$ is 1 if the facility located at i is assigned to demand point j and i is the ℓ^{th} closest located facility to j.

$$\text{(M1)} \quad \min \quad U \tag{A.1a}$$

$$\text{s.t.} \quad \sum_{i\in\mathcal{I}} d_{ij} X_{ijr} \le U \quad \forall j \in \mathcal{J} \tag{A.1b}$$

$$\sum_{i\in\mathcal{I}} X_{ij\ell} = 1 \quad \forall j \in \mathcal{J},\ \ell = 1,\ldots,r \tag{A.1c}$$

$$\sum_{\ell=1}^{r} X_{ij\ell} \le 1 \quad \forall i \in \mathcal{I},\ j \in \mathcal{J} \tag{A.1d}$$

$$d_{ij}\, X_{ij\ell} \le d_{i'j} + M_j(1 - X_{i',j,\ell+1}) \quad \forall j \in \mathcal{J};\ \ell = 1,\ldots,r-1;\ i \ne i' \in \mathcal{I} \tag{A.1e}$$

$$X_{ij\ell} \le Y_i \quad \forall i \in \mathcal{I},\ j \in \mathcal{J},\ \ell = 1,\ldots,r \tag{A.1f}$$

$$\sum_{i\in\mathcal{I}} Y_i \le p \tag{A.1g}$$

$$X_{ij\ell} \in \{0,1\} \quad \forall i \in \mathcal{I},\ j \in \mathcal{J},\ \ell = 1,\ldots,r \tag{A.1h}$$

$$Y_i \in \{0,1\} \quad \forall i \in \mathcal{I} \tag{A.1i}$$

Constraints (A.1b), in conjunction with the minimization objective in (A.1a), ensure that the objective value is equal to the maximum value of the weighted distance between demand points and their r^{th} closest located facility, over all demand points. Constraints (A.1c) require that a demand point be assigned to one facility at each level. Constraints (A.1d) prevent a facility from being assigned to more than one level for a demand point. Constraints (A.1e) enforce an ordering of the levels for each demand point. That is, the facility assigned to demand point j at level ℓ must have a smaller value of d_{ij} than the facility assigned at level $(\ell + 1)$. The constant M_j is assigned a large value such as $\max_{i\ne i'\in\mathcal{I}} |d_{ij} - d_{i'j}|$. Constraints (A.1f) specify that a demand point may only be assigned to a facility i at a level if the facility has been located at i. Constraints (A.1g) place a restriction on the number of facilities that are located. Constraints (A.1h) define binary assignment variables for only the r most desirable levels for each facility and demand point combination. Finally, constraints (A.1i) require the location variables to be binary.

Appendix A.2. Reformulation of the RANPCP

Unfortunately, Model $M1$ has a large number of assignment variables, $|\mathcal{I}|\times|J|\times p$ to be exact. In addition, it has a disjunctive constraint (A.1e) for each pair of consecutive pair of levels $(\ell, \ell+1)$. However, some of the variables in model $M1$ are unnecessary. In finding the optimal solution to the RANPCP it doesn't matter if demand points are assigned to the correct level for levels $\ell < r+1$, because these assignments are not included in the objective function. The only requirement for the objective function to be computed correctly is that each demand point is assigned to its correct $(r+1)^{\text{th}}$ level. Thus, it is enough to require that if $X_{ijr} = 1$ and $X_{i'j\ell} = 1$ (with $\ell < r$) then i must be further to j than i'. Thus, many of the disjunctive constraints (A.1e) are unnecessary. We take advantage of this fact in formulating a more compact model.

Variables.

- X_{ij} is equal to 1 if the facility located at i is assigned to demand point j as its $(r-1)^{\text{th}}$ or closer located facility and 0 otherwise.
- Z_{ij} is equal to 1 if the facility located at i is assigned to demand point j as its r^{th} closest located facility and 0 otherwise. Since this variable represents the assignment from a demand point to one of its backup facilities, we call it the 'backup variable'.

Indices.

- i_j^ℓ is the ℓ^{th} closest facility to demand point j.

$$
\begin{aligned}
\text{(M2)} \quad \min \quad & U && \text{(A.2a)}\\
\text{s.t.} \quad & \sum_{i\in\mathcal{I}} Z_{ij} \le U \quad \forall j \in \mathcal{J} && \text{(A.2b)}\\
& \sum_{i\in\mathcal{I}} Z_{ij} = 1 \quad \forall j \in \mathcal{J} && \text{(A.2c)}\\
& \sum_{i\in\mathcal{I}} X_{ij} = r-1 \quad \forall j \in \mathcal{J} && \text{(A.2d)}\\
& d_{ij}X_{ij} \le d_{i'j} + M_j(1 - Z_{i'j}) \quad \forall j \in \mathcal{J}, i' \in \mathcal{I}, i \neq i' \in \mathcal{I} && \text{(A.2e)}\\
& X_{ij} + Z_{ij} \le Y_i \quad \forall i \in \mathcal{I}, j \in \mathcal{J} && \text{(A.2f)}\\
& \sum_{i\in\mathcal{I}} Y_i \le p && \text{(A.2g)}\\
& X_{ij}, Z_{ij} \in \{0,1\} \quad \forall i \in \mathcal{I}, j \in \mathcal{J} && \text{(A.2h)}\\
& Y_i \in \{0,1\} \quad \forall i \in \mathcal{I} && \text{(A.2i)}
\end{aligned}
$$

The objective (A.2a) and constraints (A.2b) serve the same purpose as in model $M1$. Constraints (A.2c) require that a demand point be assigned to exactly one facility at level r. Constraints (A.2d) ensure that $r-1$ facilities are assigned to levels $r-1$ or lower. Constraints (A.2e) enforce an ordering of the levels for each demand point. That is, the facilities assigned to demand point j at levels 1 through $(r-1)$ must have a smaller value of d_{ij} than the facility assigned at level r. The constant M_j is the same as in $M1$. When two facilities have the same distance to a demand point, the following constraints should be used:

$$d_{ij}X_{ij} < d_{i'j} + \epsilon + M_j(1 - Z_{i'j}) \quad \forall j \in \mathcal{J}, i' \in \mathcal{I}, i \neq i' \in \mathcal{I} \tag{A.3}$$

The quantity ϵ should take a value less than the minimum absolute difference between two values of d_{ij}. Constraints (A.2f) specify that a demand point may only be assigned to a facility i at a level if the facility has been located at i. Constraints (A.2g) place a restriction on the number of facilities located. Constraints (A.2h) define binary assignment variables for each facility and demand point combination. Finally, constraints (A.2i) require the location variables to be binary.

One may notice that in $M2$, some of the Z_{ij} variables will be 0 in an optimal solution. In particular, for a given j, Z_{ij} will be zero for all facilities closer than the r^{th} facility. To explain this formally we first need to introduce further notation. Let i_j^ℓ be the ℓ^{th} closest facility to demand point j.

Now we state our observation in the form of a remark:

Remark 1. There exists an optimal solution to model $M2$ with $Z_{i_j^l j} = 0$ for all $1 \le \ell \le r-1$ and for all $j \in \mathcal{J}$.

Proof. (By contradiction.) Suppose there exists an ℓ $(1 \le \ell \le r-1)$ such that in the optimal solution to model $M2$, $Z_{i_j^\ell j} = 1$ for some $j \in \mathcal{J}$. As a result, $\sum_{1 \le \ell' \le r-1} Z_{i_j^{\ell'} j} < r-1$ and by constraints (A.2e), $\sum_{r-1<\ell' \le |\mathcal{I}|} Z_{i_j^{\ell'} j} = 1$. Hence, there exists an ℓ' $(r-1 < \ell' \le |\mathcal{I}|)$ such that $Z_{i_j^{\ell'} j} = 1$.

Case 1: All of the values of d_{ij} are different.

By our choice of ℓ and ℓ', $d_{i_j^\ell j} < d_{i_j^{\ell'} j}$. As a result, Constraint (A.2e) is violated for j if $Z_{i_j^\ell j} = 1$ and $Z_{i_j^{\ell'} j} = 1$.

Case 2: There exists $i, i' \in I$ such that for some $j \in \mathcal{J}$, $d_{i'j} = d_{ij}$.

By our choice of ℓ and ℓ', $d_{i_j^\ell j} \le d_{i_j^{\ell'} j}$. As a result, Constraint is (A.3) violated for j if $Z_{i_j^\ell j} = 1$ and $Z_{i_j^{\ell'} j} = 1$. □

Because of Remark 1, all variables $Z_{i_j^\ell j}$ for all $1 \le \ell \le r-1$ and all $j \in \mathcal{J}$ can be removed from Model $M2$. We denote the new model that is formed by removing variables from model $M2$ as model $M2-C$.

The linear programming (LP) relaxation of $M2-C$ can be tightened by adding the following constraints:

$$rZ_{i_j^\ell j} \le \sum_{\ell'=1}^{\ell-1} Y_{i_j^\ell} \quad \forall j \in \mathcal{J}; r \le \ell \le |\mathcal{I}|. \tag{A.4}$$

These constraints require that for a given demand point j, if its ℓ^{th} closest facility, i_j^ℓ, is chosen as its safe facility (i.e., $Z_{i_j^\ell j} = 1$), then r facilities must be located that are closer to j than i_j^ℓ (i.e., $\sum_{\ell'=1}^{\ell-1} Y_{i_j^\ell} = r$).

Appendix A.3. Formulation without Constraints to Enforce Distance-Ordering

A limiting feature of Models $M2$ and $M2-C$ are the distance-ordering Constraints (A.2e), which are numerous. These constraints can be replaced with the following requirement: if a backup assignment is made from demand point j to a facility at i, then at least r facilities must be located that are at least as close to j as i. This replacement also allows the elimination of the variables X_{ij}. Because this new formulation only uses the backup variables, W_{ij}, we denote it as BACKUP. Formulation BACKUP is as follows:

$$
\begin{aligned}
\text{(BACKUP)} \quad \min \quad & U && \text{(A.5a)}\\
\text{s.t.} \quad & d_{ij}W_{ij} \leq U \quad \forall i \in \mathcal{I}, j \in \mathcal{J} && \text{(A.5b)}\\
& rW_{ij} \leq \sum_{i':d_{i'j} \leq d_{ij}} Y_{i'} \quad \forall j \in \mathcal{J}, i \in \mathcal{I} && \text{(A.5c)}\\
& \sum_{i \in \mathcal{I}} W_{ij} = 1 \quad \forall j \in \mathcal{J} && \text{(A.5d)}\\
& W_{ij} \leq Y_i \quad \forall i \in \mathcal{I}, j \in \mathcal{J} && \text{(A.5e)}\\
& \sum_{i \in \mathcal{I}} Y_i \leq p && \text{(A.5f)}\\
& Y_i \in \{0,1\} \quad \forall i \in \mathcal{I} && \text{(A.5g)}\\
& W_{ij} \in \{0,1\} \quad \forall i \in \mathcal{I}, j \in \mathcal{J} && \text{(A.5h)}
\end{aligned}
$$

The objective (A.5a) and Constraints (A.5b) are equivalent to the other models. Constraints (A.5c) are the key to this model. They require that a backup assignment can only be made between demand point j and a facility placed at location i if at least r facilities are located at locations no further to j than i. Constraints (A.5d) require every demand point to be assigned a backup facility. Although the model is still correct without Constraints (A.5e) (because of the presence of Constraints (A.5c)), they are added to tighten the linear-programming relaxation. Constraints (A.5f) limit the number of facilities located and Constraints (A.5g) and (A.5h) require the variables to be binary.

A weakness of Formulation (BACKUP) is the use of Constraints (A.5c), in which the coefficient r acts as a “Big-M”, weakening the linear-programming relaxation.

Appendix B. Datasets Used in Experimentation

The following Table contains the 18 datasets used in the experimentation. The third column of Table B.8 indicates which datasets have weighted demand points and which do not. If demand points have weights, the distance values d_{ij} usually do not obey the triangle inequality.

Table B.8: Datasets used in experimentation

no.	name	$\lvert\mathcal{I}\rvert = \lvert J\rvert$	weights	source of data	reference
1	s55	55	yes	population centers in Washington, D.C.	Swain (1971)
2	d49	49	yes	49 US state capitals and Washington, D.C.	Daskin (1995)
3	d88	88	yes	cities in US	Daskin (1995)
4	d150	150	yes	cities in US	Daskin (1995)
5	lor100	100	yes	population centers in San Jose Dos Campos, Brazil	Lorena and Senne (2004)
6	lon150	150	yes	population centers in London, Ontario	Alp et al. (2003)
7	lor200	yes	yes	population centers in San Jose Dos Campos, Brazil	Lorena and Senne (2004)
8	lor300a	300	yes	population centers in San Jose Dos Campos, Brazil	Lorena and Senne (2004)
9	lor300b	300	yes	population centers in San Jose Dos Campos, Brazil	Lorena and Senne (2004)
10	lor400a	402	yes	population centers in San Jose Dos Campos, Brazil	Lorena and Senne (2004)
11	lor400b	402	yes	population centers in San Jose Dos Campos, Brazil	Lorena and Senne (2004)
12	lor818	818	no	population centers in San Jose Dos Campos, Brazil	Lorena and Senne (2004)

Appendix C. Effect of Weights on Binary Search Algorithm

In this section we try to determine if weighted or unweighted instances require more computation time for the binary search algorithm. We used our two largest weighted datasets, lor400a and lor400b, for our analysis. Table C.9 shows the computation times and number of binary search iterations for the weighted and unweighted versions of several instances of the RANPCP. When the optimal solution is found after the lower and upper bounds stage because $lb = ub$, the number of iterations is listed as 0. Each row shows the computation time and number of iterations required for the weighted and unweighted versions of an instance. From the results in this table, it is difficult to discern if the weighted or unweighted problem requires more computation. In half of the instances, the unweighted version requires more computation time. This is also true for the number of iterations.

Table C.9: Binary search computational results for wtd. and unwtd. datasets

						Time (s)			Number of iterations		
No.	Dataset	p	r	$\frac{p}{\lvert\mathcal{I}\rvert}$	$\frac{r}{p}$	wtd	unwtd	wtd-unwtd	wtd	unwtd	wtd-unwtd
1	lor402a	5	1	0.012	0.2	3.5	3.2	0.33	18	16	2
2	lor402a	5	2	0.012	0.4	4	3.9	0.08	14	16	-2
3	lor402a	10	1	0.025	0.1	2.9	3.8	-0.86	17	15	2
4	lor402a	10	2	0.025	0.2	3.5	3.4	0.15	13	15	-2
5	lor402a	40	1	0.100	0.025	5.4	14.0	-8.5	16	13	3
6	lor402a	40	2	0.100	0.05	0.1	15.0	-15	0	14	-14
7	lor402b	5	1	0.012	0.2	3.6	3.1	0.54	18	16	2
8	lor402b	5	2	0.012	0.4	4	3.9	0.1	14	16	-2
9	lor402b	10	1	0.025	0.1	3	3.8	-0.8	17	15	2
10	lor402b	10	2	0.025	0.2	3.6	3.4	0.15	13	15	-2
11	lor402b	40	1	0.100	0.025	5.4	14.0	-8.5	16	13	3
12	lor402b	40	2	0.100	0.05	0.1	15.0	-15	0	14	-14
Min						0.1	3.1	-15	0	13	-14
Max						5.4	15	0.54	18	16	3
Average						3.3	7.2	-3.9	13	15	-1.8

Appendix D. Saturation

In this section we prove the existence of a structural property of the RANPCP called *saturation*. An instance of the RANPCP is saturated if the r closest facilities are located for a given demand point and the distance from that demand point and its r^{th} closest located facility is equal to the optimal objective value. When an instance is saturated for a given value of p and r, locating additional facilities does not improve the objective.

Let $V(p,r)$ be the optimal objective value for an instance of the RANPCP with p facilities and r neighbors. An instance of the RANPCP is said to be *saturated* for a given p and r if an optimal solution exists that has an objective value of $\max_j\{d_{i_j^r j}\} = V(p,r)$. We call the quantity $\max_j\{d_{i_j^r j}\}$ the *saturation objective*.

Lemma 1. *For an instance with r neighbors, the saturation objective is obtained when the r closest facilities to demand point j are located, where $j = arg\max_j\{d_{i_j^r j}\}$.*

Proof. By the definition of saturated, $\max_j\{d_{i_j^r j}\} = V(p,r)$. Let $j = \arg\max_j\{d_{i_j^r j}\}$. By the definition of the RANPCP, the distance from j and each of its r closest located facilities must be less than $V(p,r) = \max_j\{d_{i_j^r j}\}$. However, the distance from j and each of its r closest located facilities can only be less than $V(p,r) = \max_j\{d_{i_j^r j}\}$ if the r closest facilities to j are located. □

Theorem 1. *If an instance is saturated for a given p ($p \leq |J| - 1$) and r, then $V(p,r) = V(p+1,r)$ and the instance is also saturated for $p+1$ and r.*

Proof. By the definition of saturation, there exists a $j \in \mathcal{J}$ such that $d_{i_j^r j} = V(p,r)$. By Lemma 1, the r closest facilities to j have been located. Let $p = p+1$. Thus, one new facility can be located. Wherever the new facility is located, it will be at least as far from demand point j as facility i_j^r. As a result, this additional facility location would not change the distance from j to its r^{th} closest located facility and the optimal objective value is not changed. Thus, $V(p,r) = d_{i_j^r j} = V(p+1,r)$ and therefore the instance is saturated for $p+1$ and r. □