

# Facility Location on a Network with Unreliable Links

Refael Hassin\* R. Ravi\* F. Sibel Salman $\diamond$

*\*Department of Statistics and Operations Research, Tel Aviv University  
Tel Aviv, Israel*

*\*School of Business, Carnegie Mellon University  
Pittsburgh, USA*

*$\diamond$ College of Engineering, Koç University  
Istanbul, Turkey*

---

## Abstract

In this paper we study a simple vulnerability-based stochastic dependency model of link failures in a network prone to disasters. Under this model, we study the problem of locating  $k$  facilities to maximize the expected demand serviced within a given distance, and show its equivalence to the well-studied maximum  $k$ -facility location problem. In the special case when there is no distance constraint, we give two solutions to the  $k$ -facility location problem using dynamic programming and a greedy algorithm.

**Keywords:** *facility location, disaster response, edge failures, failure dependency*

---

## 1 Introduction

Faced with the risk of potential disasters, response agencies often establish facilities that are utilized to pre-position durable relief items and may also act as supply and distribution centers after the disaster. The distribution of the relief items from the facilities to the affected areas must be done via a transportation network, mostly on highways and roads. However, the disaster may damage the components of the network and render some of the links non-operational. It may even lead to disconnectedness. For example, after an earthquake links of the network may become non-functional due to deformation of the highways, collapses of bridges and viaducts, as well as building collapses and natural gas line explosions. Identifying and incorporating the failure risk of the network links into the facility location decision may increase the effectiveness of post-disaster response.

There exists a small but increasing number of studies on facility location for disaster response. For example, Dekle et al. [3] present a case study on locating facilities in the state of Florida for disaster preparedness, while Gormez et al. [5] locate emergency facilities for earthquake preparedness in Istanbul. Jia et al. [7] propose a formulation with scenarios and a service level requirement. In another scenario-based study with service levels, Balcik and Beamon [1] propose a model to determine the number and locations of distribution centers in a relief network.

Literature on facility location includes studies on the failure of facilities or unavailability of service at the facilities due to factors such as congestion or disruptions. For example, Snyder and Daskin [12] study the  $p$ -median and uncapacitated fixed-charge facility location models when each facility may fail with a fixed probability and the objective is to minimize the expected cost of servicing the clients after failure in addition to the operating costs. Previous work on locating facilities on a network subject to

link failures has remained limited to either single edge failure, or a single facility location on a tree network whose edges may fail. Eiselt, Gendreau and Laporte [4] have considered a single edge failure while locating  $p$  facilities with the objective of minimizing total expected demand disconnected from the facilities. Melachrinoudis and Helander [9] studied a single facility location problem on a tree with edges that fail independently with given probabilities. The objective was to maximize the expected number of demand nodes reachable by operational paths. Wolle [13] addressed the basic problem of calculating the probability of serving all demand points via facilities with given locations, under node and link failures.

The problem we study here is to locate  $k$  facilities on a network whose edges may fail with given probabilities to maximize expected demand serviced. A demand point is serviced if a facility can be reached from it within a specified distance in a surviving network. We also address the special case where the distance limit is relaxed. In both cases, we do not assume links fail independently. Instead, we consider a link failure dependency model suitable for disaster risk, namely the VB-dependency. We provide an exact Dynamic Programming algorithm and an exact greedy algorithm for the problem with no distance limit.

## 2 Problem Definition

The problem we address is locating emergency response facilities on a network whose links are subject to random failure after a potential disaster event. The input graph  $G = (V, E)$  consists of node set  $V = \{v_1, \dots, v_n\}$  and edge (link) set  $E = \{e_1, \dots, e_m\}$ . Each node  $v_i$  can be a demand point with demand value (weight)  $w_i \geq 0$ . Each link  $e_i$  has length  $l_i \geq 0$ , and may exist in either operational or non-operational state after the disaster, which we refer to as the failure or survival of the link. Let  $\xi_i = 1$ , if  $e_i$  survives and 0, otherwise. The random variable  $\xi_i$  takes the value 1 with probability  $p_i$ . That is,  $p_i$  represents the probability that link  $e_i$  survives. We assume that each node will survive and an emergency response facility can be located at any node. The problem is to find the locations of at most  $k$  facilities. The goal is to satisfy as much demand as possible within a time/distance limit, as time carries critical value. We assume that a demand node can be covered, if a facility exists within distance  $R$  to itself. The main objective we consider is to maximize the expected demand covered. We also consider the case where the distance limit is relaxed ( $R$  is set to a high number).

After the disaster, the set of edges that have survived define a "surviving network", represented by the vector  $\xi = (\xi_i)$ . In general, the vector  $\xi$  has  $2^m$  realizations where each one corresponds to a different surviving network consisting of a number of connected components. Given the set of open facilities and a surviving network, total demand covered can be calculated easily but finding the total expected demand covered requires  $O(2^m)$  such calculations. However, under statistical dependency of link failures, the number of realizations with positive probability may reduce.

When link failures occur due to a common cause such as a disaster, it is often necessary to treat the link failures as dependent events. Gunnecc and Salman [6] proposed a dependency model, which we also use in this study. They partition the set of links into sets with dependency among its elements. In each set, the links are sorted with respect to their probability of survival and the following dependency relation is defined.

### Vulnerability-Based Dependency:

**Definition 2.1.** *Given two links  $i$  and  $j$  with survival probabilities  $p_i$  and  $p_j$ , we say links  $i$  and  $j$  have Vulnerability-based dependency (VB-dependency), if  $p_i \leq p_j$  implies  $P(i \text{ fails} \mid j \text{ fails})=1$ .*

By this definition, the failure of a particular link implies failure of all links weaker than that one. The VB-dependency tries to factor in the vulnerability of the components in a link, such as the strength of a bridge and the soil type on which the link stands. Links in close geographic proximity will be prone to a similar disaster magnitude and are expected to show similar behavior, creating dependency. However, the inherent vulnerability will create differences in outcomes.

Under the VB-dependency, only  $m + 1$  surviving network realizations have positive probability. The edges are re-indexed such that  $p_{i-1} \geq p_i$ , for  $i = 2, \dots, m$ , (1 represents the strongest link and  $m$  the weakest). Then the realizations with positive probability are in the form  $\xi^a = (1, 1, 1, \dots, 1, 0, 0, \dots, 0)$ ; the

first strongest  $q$  links survive and the remaining fail, for  $q = 0, 1, \dots, m$ . When  $q = 0$  (all links fail), the realization has probability  $1 - p_1$  and when  $q = m$  (all links survive), the probability is  $p_m$ . In the other cases, the probability is  $p_q - p_{q+1}$  (as given in [6]).

## 2.1 Formulations and Hardness

We select nodes to locate at most  $k$  facilities in the input network. We assume that if a facility is established at a node, it covers the demand of all nodes that can be reached from it via a path of length  $R$  in the surviving network. If the locations of the facilities are fixed, in each possible surviving network realization, total demand covered can be evaluated by applying a shortest-path algorithm starting from each facility node. The location problem is to place at most  $k$  facilities to *maximize the expected total demand covered within distance  $R$* . We denote this problem by MAX-EXP-COVER-R. In this problem, we assume that a sufficient amount of supply will be available at the facilities and the links fail according to *VB - dependency*.

Suppose  $F \subseteq V$  represents the selected facility locations. Let  $I_v^q$  be an indicator variable such that it takes the value 1, if demand of node  $v$  can be covered in network realization  $\xi^q$  with facilities in  $F$ , and 0, otherwise. Note that  $I_v^q$  will be one if there is a path of length at most  $R$  from  $v$  to one of the facilities in  $F$  in the graph defined by  $\xi^q$ . Let  $P(q)$  be the probability that  $\xi^q$  occurs. Then, MAX-EXP-COVER-R can be formulated as follows.

$$\text{MAX-EXP-COVER-R: Find } F \subseteq V, |F| \leq k \text{ to } \max \sum_{q=0}^m \sum_{v_i \in V} P(q) w_i I_{v_i}^q.$$

**Proposition 2.1.** *The MAX-EXP-COVER-R problem is NP-Hard in the strong sense.*

*Proof.* The proof is by reduction from the maximum k-facility location problem (defined in [2]), which is known to be NP-hard. In the maximum k-facility location problem, a set of clients  $I$  and a set of potential facility locations  $J$  are given with profits  $c_{ij} \geq 0$  for each pair  $i \in I, j \in J$ . At most  $k$  facilities are located at a subset  $F$  of  $J$  to maximize  $\sum_{i \in I} \max_{j \in F} c_{ij}$ . Given an instance of this problem, let  $c_{max} = \max c_{ij}$  over all pairs  $i \in I, j \in J$ . Define an instance of MAX-EXP-COVER-R by taking the complete bipartite graph  $I \times J$  as the input graph  $G$ . For each edge  $(v_i, v_j)$ ,  $i \in I, j \in J$ , set the probability of survival (reliability) to  $c_{ij}/c_{max}$  (so that it is between zero and one) and its length to 1. Set  $R = 1$  and  $w_i = 1, \forall v_i \in I, w_j = 0, \forall v_j \in J$ . Then, any solution that maximizes expected total demand covered locates the facilities at a subset of  $J$  due to the distance limit  $R = 1$ . Furthermore, facilities will be selected to maximize the total reliability of the edges connecting each node  $v_i, i \in I$  to a facility node  $v_j, j \in F$  with maximum  $c_{ij}/c_{max}$ . Hence, this solution also maximizes the profits in the k-facility location problem.  $\square$

We next show that MAX-EXP-COVER-R reduces to the maximum k-facility location problem; hence, any solution algorithm developed for the latter can be used to solve the former by means of the transformation in the proof.

**Proposition 2.2.** *The MAX-EXP-COVER-R problem can be transformed to the maximum k-facility location problem in polynomial time.*

*Proof.* Given an instance of MAX-EXP-COVER-R with input graph  $G = (V, E)$ , we define a complete bipartite graph  $V \times V'$  by duplicating the node set  $V$  as  $V'$ . The set  $V$  corresponds to the set of clients and  $V'$  to the set of potential facility locations. For edges  $(v_j, v'_j)$ , we set  $c_{v_j, v'_j} = w_j$ , for  $v_j \in V$ . We next define the profit of the pair  $(v_i, v'_j)$  for  $i \neq j$ . For a pair of nodes  $v_i$  and  $v_j$  in  $V$ , let  $d_{ij}^q$  be the distance between the two nodes in a given network realization  $\xi^q$ . Note that if the distance  $d_{ij}^q$  exceeds  $R$ , then it will remain so in all of  $\xi^{q-1}, \dots, \xi^1, \xi^0$ . If  $d_{ij}^m > R$ , then we set the profit  $c_{v_i, v'_j}$  to 0, since a facility in  $v_j$  cannot serve  $v_i$  in any realization. Otherwise, let  $s$  be the smallest index such that  $d_{ij}^q \leq R$  for all  $q \leq s$ . Then, we set  $c_{v_i, v'_j} = w_j \sum_{q=s}^m P(q) = w_j p_s$ , as a facility in  $v_j$  can serve  $v_i$  in realizations  $s$  to  $m$ .

Suppose  $F \subset V', |F| \leq k$ , is an optimal solution to this instance of the maximum k-facility location problem. Then  $F$  gives the maximum value of  $\sum_{v_i \in V} \max_{v'_j \in F} c_{v_i, v'_j}$  by definition. As the problem is

uncapacitated, each client is serviced by one facility. If a facility is located at  $v'_j$ , it services  $v_j$  with profit equal to  $w_j$  as any other facility in  $F$  will provide a smaller profit. For a node  $v_i$  such that  $v'_i$  is not in  $F$ , it is serviced by some  $v'_j \in F$ ,  $i \neq j$ , such that  $c_{v_i, v'_j}$  is maximum over all facilities. Note that for any fixed  $F$ , for any  $i$  such that  $v'_i \notin F$ , we can assign it to some  $v'_j \in F$  such that  $d_{ij}^s \leq R$  for the maximum possible  $s$  over all such  $v'_j$ . In this way, the demand at  $i$  will be serviced for the most scenarios.

Thus under the above reduction, the set of facilities  $F^*$  that are optimal for the maximum  $k$ -facility location problem will also form a solution to the MAX-EXP-COVER-R problem with the same objective value.  $\square$

Cornuejols, Fisher and Nemhauser [2] showed that a greedy algorithm has a worst-case bound  $1 - 1/e$  for the maximum  $k$ -facility location problem. By Proposition 2.2, the same ratio will be valid for MAX-EXP-COVER-R. We next define a special case of this problem where the distance limit  $R$  is relaxed, and refer to it as MAX-EXP-COVER. In this case,  $I_v^q = 1$ , if there is a path from  $v$  to one of the facilities in  $F$  in the graph defined by  $\xi^q$ . As a result, if a facility exists in a component of the network realization, it covers the demand  $w_j$  of all nodes  $v_j$  in the component. Thus, in an optimal solution at most one facility is located in a component.

### 3 Algorithms for Maximizing Expected Demand Served without the Distance Constraint

In the MAX-EXP-COVER problem we select  $k$  facility locations to maximize the expected demand that is serviced by a facility, over all realizations. We first show that under VB-dependency of link failures, this problem can be reformulated on a rooted tree.

#### 3.1 Disconnectors and the Component Tree

Without loss of generality we may assume that the input graph  $G$  is connected. Under *VB-dependency* of link failures, the possible network realizations are  $\xi^q$ ,  $q = 0, 1, \dots, m$ , where  $\xi^m$  corresponds to  $G$ . As we go from  $\xi^m$  (all ones) to  $\xi^0$  (all zeros) one by one, each time one more link fails. Along the way, the number of components in each realization will increase until eventually we get to  $\xi^0$  that consists of  $n$  components. We can detect the links whose failure causes the number of components to go up. Start with the weakest link. Suppose it fails. Check if the graph is connected. Repeat until the graph has more than one component. At that point, the last link that failed is named  $e_{[1]}$ . Continue in this way. The next link that increases the number of connected components is named  $e_{[2]}$ , and so on. By this procedure, we obtain the links  $e_{[1]}, \dots, e_{[n-1]}$ , and call them the *disconnectors* of  $G$ . Note that  $e_{[1]}$  is the *weakest* and  $e_{[n-1]}$  is the *strongest* disconnector. As the disconnectors cannot contain a cycle, they form a spanning tree  $T$  of  $G$ .

Next, we define a new tree that represents how  $G$  is disconnected into its components as the disconnectors  $e_{[1]}, \dots, e_{[n-1]}$  fail in order. We call this tree the *Component Tree* of  $G$ , and denote it by  $CT$ . The leaves of the Component Tree are the nodes of  $G$ . The intermediate nodes are the disconnectors, and the root node is the first disconnector,  $e_{[1]}$ . Each disconnector creates two children in  $CT$ , representing the subtrees of  $T$ . We use the following procedure to define  $CT$ . Let  $e_{[1]}$  be the root node. When  $e_{[1]}$  fails, it creates two components in  $T$ . If any one of the components is a singleton, then that child is given the name of the singleton node and becomes a leaf node. Otherwise, the child is given the name of the next disconnector to fail in that component (i.e. the one with minimum index). Then, in any non-leaf node  $e_{[k]}$ , two children are created by removing  $e_{[k]}$  from  $T$  in the same way.

#### 3.2 Reformulation of MAX-EXP-COVER

We associate two quantities with each node of the Component Tree  $CT$ : a weight  $W$  and a probability  $P$ . At any leaf node corresponding to an original node  $v_i$  of the graph, the weight  $W(i) = w_i$  and the

probability  $P(i) = P(0)$ , the probability of the scenario where all links have failed. The partial expected demand/weight that accrues at this node by placing a facility here is thus  $W(i)P(i) = w_i P(0)$ . Let us denote this as *Expected Weight* or  $EW(i)$  for this leaf  $v_i$ .

At any non-leaf node corresponding to a disconnecter edge  $e_{[i]}$ , we have two child subtrees say  $L$  and  $R$ . The weight assigned to this node  $W(e_{[i]})$  is the sum of the weights of all the leaf nodes in the subtrees  $L$  and  $R$ . The probability  $P(e_{[i]})$  is set to the sum of the probabilities of all realizations in which all the leaves in subtrees  $L$  and  $R$  are exactly the nodes of a single connected component. This connected component is formed when  $e_{[i-1]}$  fails for the first time and remains to be a single component until  $e_{[i]}$  fails. Suppose  $e_{[i]} = e_q$  and  $e_{[i-1]} = e_t$ . By notation,  $q < t$ . Then,  $P(e_{[i]})$  is set to  $P(q) + P(q+1) + \dots + P(t) = p_q - p_{t+1}$ . Note that if there is a facility located at a leaf node in either  $L$  or  $R$  that will lead to an expected demand/weight of  $W(e_{[i]})P(e_{[i]})$  to accrue in the objective function of expected demand. As before we can denote this as *Expected Weight* or  $EW(e_{[i]})$  of this non-leaf node labeled by  $e_{[i]}$ . Finally at the root of the tree, we set  $P(e_{[1]})$  as follows. Let  $e_{[1]} = e_q$ . Then,  $P(e_{[1]}) = P(q) + P(q+1) + \dots + P(m) = p_q$  so that we add the probability of the realizations when all nodes are in a single component.

In the above reformulation we have split up the total probability of 1 among  $n - 1$  realizations, one for each disconnecter edge. In addition, we have set the weights of the components in the component tree so as to calculate the weights of these realizations appropriately.

With the above re-formulation, we can view the problem of locating  $k$  facilities at nodes as a simple location problem on a rooted tree  $T$  with nonnegative weights  $EW(v)$  at all nodes: the goal is to choose  $k$  leaves such that the sum of the  $EW$  values of all nodes in the union of paths from the chosen leaves to the root is maximum. Let us denote this problem as MAX-WT-K-LEAF-SUBTREE. It is for this problem that we provide two solutions - one via dynamic programming and another via a greedy algorithm.

### 3.3 A Dynamic Programming Algorithm

For every node  $v$  in the tree  $T$  and for every nonnegative integer  $t \leq k$ , we denote  $EW(v, t)$  as the maximum expected weight obtained by placing exactly  $t$  facilities in the leaves of the subtree  $T_v$  of the tree rooted at  $v$ . Note that if  $t \geq 1$ , we know that this solution has an open facility among the leaves by definition.

A simple recurrence for  $EW(v, t)$  follows. Let  $L$  and  $R$  be the two subtrees that are the children of  $T_v$  with root nodes  $v_l$  and  $v_r$  respectively. If  $t = 0$  we have  $EW(v, 0) = 0$  at all nodes  $v$  along the tree. If  $v$  is a leaf, then  $EW(v, t) = EW(v)$ , for all  $1 \leq t \leq k$ . For a non-leaf node and for  $t > 0$ , we have the following relation.

$$EW(v, t) = \max_{0 \leq t' \leq t} \{EW(v_r, t') + EW(v_l, t - t')\} + EW(v). \quad (1)$$

The recurrence corresponds to allocating  $t'$  of the  $t$  facilities optimally in the right subtree and the remaining in the left and counting in the expected profit of the root node  $v$  as long as  $t \geq 1$  since in this case some leaf will have a facility and allow the expected weight at the root node to be counted in the objective. The recursion proceeds bottom-up from leaves to the root node. As the component tree has  $2n - 1$  nodes, this yields an  $O(kn)$  exact algorithm for MAX-EXP-COVER, after the construction of the component tree. Note that the construction of the component tree takes polynomial time.

### 3.4 A Greedy Algorithm

The greedy algorithm for choosing  $k$  leaves to maximize total expected weight of the paths to the root is natural: For  $k$  steps, we pick the leaf such that the incremental addition to the total expected weight by adding this node to the solution is as large as possible.

**Proposition 3.1.** *The greedy algorithm outputs an optimal set of  $k$  leaves that maximizes the total node weight of the union of paths from the chosen leaves to the root.*

*Proof.* The proof is by a simple interchange argument. Suppose that the greedy algorithm results in a solution GREEDY that is sub-optimal and has lower weight than the optimal solution OPT. Consider the symmetric difference of the leaves added in the optimal solution OPT and the greedy solution GREEDY and let  $l$  be the leaf from GREEDY in this symmetric difference that was chosen earliest. If we add  $l$  to the current OPT and delete any current leaf say  $l'$  from OPT (but not in GREEDY), we get a solution that is closer to GREEDY. More importantly, this solution has at least as much weight as OPT since by the greedy choice the contribution of this leaf is at least as much as that of the one we removed (namely  $l'$ ). Dropping  $l'$  and adding  $l$  thus leads to no loss in the weight and thus we get one step closer to GREEDY without reducing the weight of OPT. Continuing this way we can show that GREEDY results in an optimal solution.  $\square$

The above proof generalizes the results [10, 11] known for a similar unrooted version of the problem (which can be reduced to our rooted version by trying all choices of the root). These results in turn follow also from the Greedoid framework [8]. The greedy algorithm is also  $O(kn)$  after the construction of the component tree, as in each step the nodes are traversed at most once.

## 4 Extensions

When the supply quantities are limited and a shortage can possibly occur, a capacitated problem can be formulated. If we assume that the facilities have a fixed capacity and a fixed cost for opening, we obtain an extension of the well-known capacitated fixed-charge facility location problem. The problem is to locate facilities on the nodes so that a weighted sum of facility opening costs and expected unsatisfied demand costs is minimized. Our dynamic programming algorithm can be extended to solve this capacitated problem when no distance limit is imposed.

## 5 Acknowledgement

This research is partially supported by a NATO Collaborative Linkage Grant and a TUBITAK Career Grant.

## References

- [1] B. Balcik and B. M. Beamon, *Facility location in humanitarian relief*, International Journal of Logistics Research and Applications, 11(2):101–121, 2008.
- [2] G. Cornuejols, M. Fisher and G. Nemhauser, *Location of bank accounts to optimize float: An analytic study of exact and approximate algorithms*, Management Sci. 23, 789-810, 1977.
- [3] J. Dekle, M. S. Lavieri, E. Martin, H. Emir-Farinas and R. L. Francis, *A Florida County Locates Disaster Recovery Centers*, Interfaces, 35(2): 133-139, 2005
- [4] H. A. Eiselt, M. Gendreau, and G. Laporte, *Location of facilities on a network subject to a single-edge failure*, Networks, 22:231-246, 1992.
- [5] N. Gormez, M. Koksalan and F. S. Salman, *Disaster Response and Relief Facility Location for Istanbul*, Journal of Operational Research, *submitted* August 2008.
- [6] D. Gunneç and F. S. Salman, *Assessing the Reliability and the Expected Performance of a Network Under Disaster Risk*, Proceedings of the International Network Optimization Conference (INOC), April 22-25, 2007, Spa, Belgium.
- [7] H. Jia, F. Ordonez and M. M. Dessouky, *A modeling framework for facility location of medical services for large-scale emergencies*, IIE Transactions, 39:41-55, 2007.

- [8] B. Korte, L. Lovasz and R. Schrader, *Greedoids, algorithms and combinatorics*. Springer, Berlin, 1991.
- [9] E. Melachrinoudis and M. E. Helander, *A single facility location problem on a tree with unreliable edges*, *Networks*, 27(3): 219-237, 1996.
- [10] F. Pardi and N. Goldman, *Species choice for comparative genomics: Being greedy works*, *PLoS Genetics*, 1(6): e71, 2005.
- [11] M. Steel, *Phylogenetic Diversity and the Greedy Algorithm*, *Systemic Biology*, 54(4): 527-529, 2005.
- [12] L.V. Snyder and M. S. Daskin, *Reliability Models for Facility Location: The Expected Failure Cost Case*, *Transportation Science*, 39(3): 400-416, 2005.
- [13] T. Wolle, *A Framework for Network Reliability Problems on Graphs of Bounded Treewidth*, In Proceedings of the 13th International Symposium on Algorithms and Computation, LNCS, Vol. 2518, pp. 137-149, 2002.