Management Science and Financial Engineering
Vol 19, No 2, November 2013, pp.19-29
ISSN 2287-2043 | EISSN 2287-2361 |
http://dx.doi.org/10.7737/MSFE.2013.19.2.019

Designing Refuse Collection Networks under Capacity and Maximum Allowable Distance Constraints

Ji-Su Kim
Department of Industrial Engineering, Hanyang University
Dong-Ho Lee*
Department of Industrial Engineering, Hanyang University

(Received: September 9, 2013 / Revised: October 21, 2013 / Accepted: October 27, 2013)

ABSTRACT

Refuse collection network design, one of major decision problems in reverse logistics, is the problem of locating collection points and allocating refuses at demand points to the opened collection points. As an extension of the previous models, we consider capacity and maximum allowable distance constraints at each collection point. In particular, the maximum allowable distance constraint is additionally considered to avoid the impractical solutions in which collection points are located too closely. Also, the additional distance constraint represents the physical distance limit between collection and demand points. The objective is to minimize the sum of fixed costs to open collection points and variable costs to transport refuses from demand to collection points. After formulating the problem as an integer programming model, we suggest an optimal branch and bound algorithm that generates all feasible solutions by a simultaneous location and allocation method and curtails the dominated ones using the lower bounds developed using the relaxation technique. Also, due to the limited applications of the optimal algorithm, we suggest two heuristics. To test the performances of the algorithms, computational experiments were done on a number of test instances, and the results are reported.

Keywords: Reverse Logistics, Refuse Collection Network Design, Maximum Allowable Distances

* Corresponding Author, E-mail: leman@hanyang.ac.kr

1. INTRODUCTION

Due to environmental, economic and social reasons, reverse logistics has been paid much attention among researchers and practitioners for the last decades. Reverse logistics, an opposite direction of the conventional forward logistics, is defined as the logistics activities all the way from end-of-use/life products no longer required by the user to further recovery or disposal (Fleischmann *et al.*, 1997; Bloemhof-Ruwaard *et al.*, 1999; Srivastava, 2007; Lee *et al.*, 2008; Sasikumar and Kannan 2008a, b). Compared with the forward logistics that has the divergent structure, the reverse logistics has the convergent structure from many sources to collection, recovery or disposal facilities (Fleischmann *et al.*, 2000). Also, designing and operating reverse logistics systems is complicated by several factors, such as the needs to test and grade end-of-use/life products, the needs to address uncertainty in terms of return quantity, quality and timing (Dowlatshahi, 2000).

The major activities in reverse logistics can be described in Figure 1, which is adopted from Lee *et al.* (2008). In this figure, recycling implies material recovery without conserving any product structure, e.g. metal recycling from scrap and plastic recycling, while remanufacturing is the transformation of end-of-use/life products into those that satisfy exactly the same quality as new or other standards. While the specific activities differ per case, the typical reverse logistics activities include collection, disassembly, sorting, testing, recovery (repair, reuse and remanufacturing), recycling, disposal, etc.

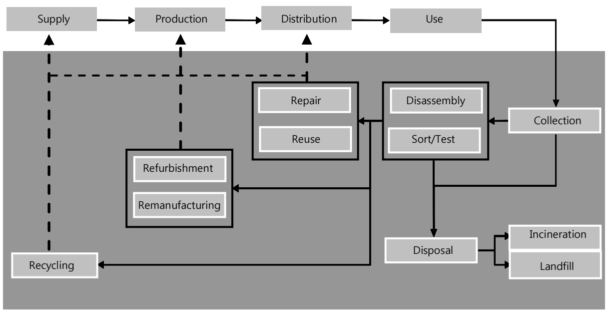

Figure 1. Activities in Reverse Logistics

Among the activities explained above, this study focuses on collection, i.e. gathering end-of-use/life products and moves them to the facilities where further recovery or disposal activity is done. Here, end-of-use/life products are called refuses in solid waste management (Vesilind *et al.*, 2002). In fact, refuse collection is one of indispensable activities in reverse logistics since refuses must be collected for further product recovery or disposal.

There are various decision problems in refuse collection systems, which can be classified into design and operational problems (Mansini and Speranza, 1998). The design decisions include the number and locations of collection points, the types of collection facilities, the number and types of collection vehicles, the number of workers, etc. Among them, the number and locations of collection points are important decisions since they work as the most basic structural elements in collection systems. Here, the collection point is the place where refuses are gathered and handled (sorting, testing and storing) for further treatments. On the other hand, the operational decisions include vehicle routing, vehicle scheduling and control, workforce scheduling, etc.

This study focuses on the problem of determining the number and locations of collection points as well as the allocations of demand points to the opened collection points while satisfying capacity at each collection point and maximum allowable distance constraints between demand and collection points. Here, the maximum allowable distance constraint is additionally considered for the purpose of avoiding impractical solutions in which collection points are located too closely. In fact, most refuse collection networks have physical distance limits between demand and collection points. Unlike the previous research on designing the entire reverse logistics network, we narrow the scope to the collection system, so that a basic but more concrete model can be derived. The model suggested in this study is generic in the sense that it includes the most essential features of refuse collection networks, and hence it can be used in any applications after certain customizations. Note that most previous research articles on refuse collection network design are case studies on specific network types (See the next section for a literature review).

To represent the problem mathematically, an integer programming model is developed. Then, two types of solution algorithms, optimal and heuristic, are suggested. The optimal algorithm is based on the branch and bound technique that generates all feasible solutions by simultaneous location and allocation and curtails the dominated solutions using the upper and lower bounds developed in this study. Here, the lower bounds are developed using the relaxation technique. Due to the limited applications of the optimal solution algorithms, two heuristic algorithms are also suggested. To test the performances of the solution algorithms, computational experiments were done on various test instances and the results are reported.

The remainder of this paper is organized as follows. In the next section, we review the previous research articles on network design in reverse logistics. In Section 3, the problem is described in more detail with an integer programming model. The optimal and heuristic algorithms and their test results are presented in Sections 4 and 5, respectively. Finally, Section 6 concludes the paper with a summary and provides some areas for further research.

2. LITERATURE REVIEW

This section summarizes the previous studies on network design in reverse logistics. After the case studies are reviewed, the theoretical ones are explained. Other studies related to integating forward and reverse logistics network design are also reviewed. For the literature reviews on various reverse logistics network design and operational problems, see Dekker *et al.* (1998), Srivastava (2007), Akçali *et al.* (2008), Melo *et al.* (2009), and Ilgin and Gupta (2010).

2.1 Case Studies

The case studies on reverse logistics network design can be classified by the major activities in reverse logistics, i.e. reuse, remanufacturing, recycling, and disposal. See Fleischmann *et al.* (2000) for a classfication scheme. Note that most case studies suggest mathematical models for specific network types and give solutions using the existing algorithms or commercial software packages.

Kroon and Vrijens (1995) considered a reverse logistics network that collects and reprocesses returnable containers, and suggest an integer programming model that determines the number of containers required, the number and locations of container depots, and the appropriate service, distribution and collection fees. Krikke *et al.* (1999) developed a mixed integer programming model to determine the location of shredding and melting facilities for remanufacturing end-of-life automobiles, and Jayaraman *et al.* (1999) considered the problem of determining the locations of remanufacturing and distribution facilities together with production quantities and inventory of remanufactured products. Lee and Dong (2008) suggested a location-allocation model for designing an integrated forward and reverse logistics network in which end-of-life computers are collected, recovered, distributed, and disposed. See Caruso *et al.* (1993), Spengler *et al.* (1997), Barros *et al.* (1998), Louwers *et al.* (1999), Kara *et al.* (2007), Lu and Bostel (2007), and Pati *et al.* (2008) for other case studies on designing various reverse logistics networks.

2.2 Theoretical Studies

To overcome the limitations of case studies, theoretical studies have been done. In this paper, we classify them into: (a) the studies on the entire reverse logistics networks; (b) the studies on integrated forward and reverse logistics networks; (c) the studies on collection networks.

2.2.1 Studies on the Entire Reverse Logistics Networks

Jayaraman *et al.* (2003), who extended the earlier case study of Jayaraman *et al.* (1999), suggested a two-level hierarchical facility location model that determines the numbers and locations of collection points and refurbishing facilities for hazardous products, and Min *et al.* (2006a) suggested genetic algorithms that determine the numbers, locations and sizes of collection points and centralized return centers in a multi-echelon reverse logistics. Lieckens and Vandaele (2007) proposed a mixed integer non-linear programming model for the stochastic single-period, single-echelon and single-product facility location-allocation problem in a reverse logistics network with stochastic lead times. See Min *et al.* (2006b), Aras *et al.* (2008), Cruz-Rivera and Ertel (2009) and Lee and Dong (2009) for other theoretical studies on designing reverse logistics networks.

2.2.2 Studies on Integrated Forward and Reverse Logistics Networks

Fleischmann *et al.* (2001) suggested a mathematical programming model to determine the numbers and locations of plants, warehouses and disassembly centers in multi-echelon forward distribution and reverse recovery networks, and later, Salema *et al.* (2007) generalize the earlier model by considering multiple products, capacity constraints, and uncertainty in the demand and return flows. Sim *et al.* (2004) suggested a genetic algorithm for designing recovery networks with hybrid manufacturing and remanufacturing facilities, distribution and collection centers, and Ko and Evans (2007) considered the problem of designing networks with manufacturing facilities and co-located distribution/collection centers. See Sahyouni *et al.* (2007) and Wang and Yang (2007) for other models.

2.2.3 Studies on Collection Networks

Unlike the articles that consider the entire reverse (and forward) logistics networks, some articles limit their scope to the collection activity. Bautisa and Pereira (2006) suggested meta-heuristics for the problem of locating collection areas in urban waste management networks, and reported a case study on a metropolitan area of Barcelona. Wojanowski *et al.* (2007) suggested a framework for designing drop-off retail-collection networks under deposit-refund policy, and developed an analytical network design model, i.e. determining the locations of identical retail-collection facilities and retail (sales) price so as to maximize the expected profit. Aras and Aksen (2008) provided a mixed integer nonlinear facility location-allocation model to determine the locations of collection centers and the incentive values for each return type so as to maximize the profit obtained from the returns. Also, Kim and Lee (2008) suggested heuristic algorithms for the problem of determining the locations of collection points and the allocation of refuses at demand points to collection points under the capacity restriction at each collection point, and later, Kim *et al.* (2010) suggested various meta-heuristics.

3. PROBLEM DESCRIPTION

The refuse collection network design problem considered in this study can be briefly described as follows. *For a given set of potential sites, the problem is to determine the locations of collection points and the allocations of refuses at demand points to the opened collection points for the objective of minimizing the sum of fixed costs to open collection points and variable costs to transport refuses at demand points to collection points while satisfying the capacity and the maximum allowable collection distance constraints at each collection point.*

Figure 2 shows a schematic description of the collection network considered in this study. As can be seen in the figure, the example network has 3 collection points (triangles) and 9 demand points or potential sites (circles). Of the two decision variables, the location decision is done by selecting the collection points from the set of potential sites. It is assumed that only one collection point is opened at each potential site. Also, in the allocation decision, each demand point is allocated to exactly one collection point, i.e. no demand splitting is allowed. In this study, we assume that the potential sites are identical to demand points. In the objective function, the fixed costs, which may defer at different collection points, are assumed to be deterministic and given in advance, and the variable costs are assumed to be directly proportional to the distances and the refuse amounts. It is assumed that refuse amounts at demand points and distances between demand and collection points are deterministic and given in advance.

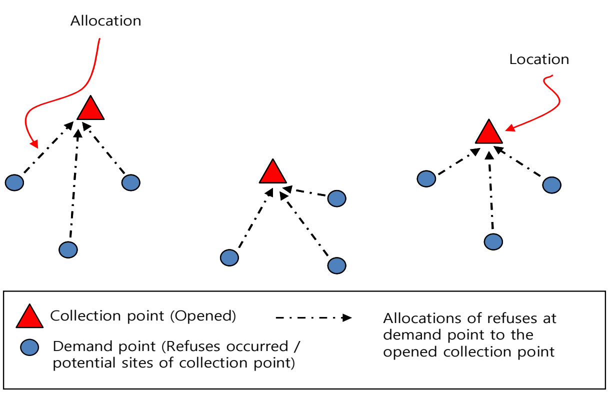

Figure 2. Collection Network Considered in This Study: Example

The problem considered here has two main constraints: (a) capacity; and (b) maximum allowable distance. The capacity constraint at each collection point implies that there is an upper limit on the amount of refuses allocated to the collection point. Also, the maximum allowable distance for a collection point implies that no demand points beyond its distance limit can be allocated to the collection point. See Klose and Drexl (2005) and Rosemary (2007) for the distance constraints in other facility location models. It is assumed that the capacities and the maximum allowable collection distances are assumed to be deterministic and given in advance.

To represent the problem more clearly, an integer programming model is suggested in this study. Define a graph $G = (N, A)$, where $N = \{1, 2, \cdots, n\}$ denotes the set of nodes representing the potential sites and $A = \{(i, j): i \neq j\}$ denotes the set of arcs that connect potential sites. Before presenting the model, the notations used are summarized below.

Parameters

- f_j fixed cost to open a collection point at node j
- c_{ij} variable cost to transport a unit of refuse per unit distance from node (demand point) i to collection point located at node j
- w_i amount of refuse at node (demand point) i
- d_{ij} distance between nodes i and j
- Q_j capacity of the collection point opened at node j
- S_j maximum allowable distance of the collection point opened at node j

Decision variables

- y_j = 1 if a collection point is opened at node j, and 0 otherwise
- x_{ij} = 1 if the refuse at demand point i is allocated to the collection point opened at node j, and 0 otherwise

Now, the integer programming model is given below:

[P] Minimize $\sum_{j \in N} f_j \cdot y_j + \sum_{(i,j) \in A} c_{ij} \cdot w_i \cdot d_{ij} \cdot x_{ij}$

subject to

$$\sum_{j=1}^{n} x_{ij} = 1 \quad \text{for all } i \in N \quad (1)$$

$$\sum_{i=1}^{n} w_i \cdot x_{ij} \leq Q_j \cdot y_j \quad \text{for all } j \in N \quad (2)$$

$$d_{ij} \cdot x_{ij} \leq S_j \cdot y_j \quad \text{for all } (i, j) \in A \quad (3)$$

$$y_j \in \{0, 1\} \quad \text{for all } j \in N \quad (4)$$

$$x_{ij} \in \{0, 1\} \quad \text{for all } (i, j) \in A \quad (5)$$

The objective function denotes minimizing the sum of fixed costs to open collection points and variable costs to transport refuses at demand points to the opened collection points. Constraint (1) ensures that each demand point is assigned to only one collection point, i.e. demand splitting is not allowed. Constraints (2) and (3) represent the capacity and the maximum allowable distance at each collection point. Also, the two constraints ensure that demand points cannot be assigned to a potential site unless there is a collection point opened at the corresponding potential site. Finally, constraints (5) and (6) represent the conditions of decision variables.

We can easily see that the problem [P] is NP-hard since the relaxed problem without the maximum allowable distance constraint is the well-known capacitated facility location problem. Also, the problem is a special case of the capacitated facility location problem with single source constraints. In fact, this study is an extension of Kim and Lee (2008, 2010) in that the maximum allowable distance constraint is additionally considered, and hence one can obtain the practical solutions in which collection points are not located closely.

4. SOLUTION ALGORITHMS

This section explains the branch and bound (B&B) algorithm that gives the optimal solutions for small-sized instances and the heuristic algorithms for practical large-sized instances.

4.1 Optimal Algorithms

The B&B algorithm is an enumeration based technique that generates all feasible solutions, i.e. branching scheme, and finds the optimal one after curtailing dominated solutions using upper and lower bounds, i.e. bounding scheme. The branching and bounding schemes of the B&B algorithm suggested in this study are explained below.

4.1.1 Branching Scheme

The branching scheme that generates all feasible solutions by simultaneous location and allocation is explained using a B&B tree. In the B&B tree, each node (except for the root node) denotes a demand point (potential site) and each arc represents the collection point to which a demand point is allocated. Also, each level corresponds to the number of considered potential sites and hence the largest level number is the number of potential sites.

The branching starts at the root node (level 0) in which no demand points are allocated to any collection point. One node is branched at the root node and n nodes are branched at the nodes from the second level. After a node is branched, the corresponding arcs are connected to the node. In this way, a complete solution can be obtained after a leaf node is branched, where a leaf node denotes the one with the largest level number. In other words, there is a one-to-one correspondence between a solution and a complete path from the root to a leaf node. Using this method, we can generate all feasible solutions more efficiently. Note that the existing branching method separates location and allocation, i.e. allocations are done for all possible location alternatives.

Figure 3 shows a B&B tree for an instance with three potential sites (n = 3). In this example, the leftmost path corresponds to a possible solution that demand points 1, 2 and 3 are allocated to the collection point opened at potential site 1, while no collection points are opened at potential sites 2 and 3. According to the definition of the B&B tree, the example has 3^3 (= 27) possible solutions. However, the number of feasible solutions may be much smaller than the total number of possible solutions due to the capacity and the maximum allowable distances constraints.

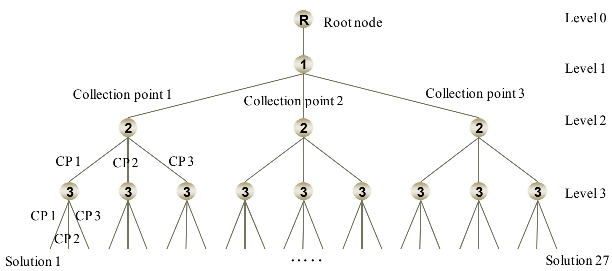

Figure 3. B&B Tree: an Example with 3 Potential Sites

Each node (except for the leaf nodes) of the B&B tree corresponds to a partial solution, which can be found by tracing a path from that node to the root node. For example, the second node (level 2) at the leftmost path represents a partial solution that demand point 1 is allocated to the collection point opened at potential site 1 while the others remain unfixed. In the branching scheme, the branch and backtrack method, alternatively called the depth-first search in the literature, is used for node selection. More specifically, if the current node is not the leaf node, the next node to be considered is one of its child nodes. Otherwise, we go back on the path from this node toward the root node until we find the first node with a child node that has not been considered yet.

4.1.2 Bounding Scheme

As in other B&B algorithms, the purpose of the bounding scheme is to reduce the search space while curtailing dominated solutions. The bounding scheme suggested in this study consists of three methods. They are: (a) eliminating infeasible solutions; (b) curtailing dominated solutions using upper and lower bounds; and (c) using a dominance property.

(*a*) *Eliminating Infeasible Solutions*

The infeasible solutions can be eliminated by checking if the current partial solution satisfies both the capacity and the maximum allowable distance constraints. In other words, after a node is branched, we can easily compute the amount of refuses allocated to the collection point corresponding to the node and the distances between the collection and the demand points associated with the branching. If a partial solution is infeasible, the node associated with the partial solution can be deleted from further considerations, i.e. the ones after the node need not be considered.

(*b*) *Curtailing Dominated Solutions Using upper and Lower Bounds*

Upper and lower bounds can also be used to reduce the search space. In other words, each node of the B&B tree can be deleted from further considerations if the lower bound at the node is greater than equal to the incumbent solution value, i.e. the smallest upper bound obtained so far.

The initial upper bound at the root node is calculated using a greedy method. More specifically, among unconsidered potential sites, the collection point is selected that has the minimum fixed cost and then demand points are allocated to the selected collection point as many as possible in the non-decreasing order of variable costs while checking if the capacity and the maximum allowable distance constraints are satisfied. Note that the upper bound is updated whenever a better feasible solution is obtained at a leaf node of the B&B tree.

Two lower bounds, which are based on the relaxation technique, are suggested in this study. The first lower bound (LB1), calculated at each node of the B&B tree, can be represented by:

$$Z_{PS} + Z_R,$$

where Z_{PS} is the cost value obtained from the partial solution and Z_R is the cost value derived from the points not included in the partial solution. More specifically, Z_{PS} can be represented as:

$$\sum_{j \in C} f_j \cdot y_j + \sum_{\{(i,j) \in A | i \in D \text{ and } j \in C\}} c_{ij} \cdot w_i \cdot d_{ij} \cdot x_{ij} \ ,$$

where C and D denote the set of potential sites selected in the partial solution and the set of demand points allocated to the selected collection points, respectively. The second term Z_R is obtained from solving the reduced problem defined for the potential sites that are not included in the partial solution, i.e. $N \backslash (C \cup D)$. The reduced problem [P′] can be formulated as follows.

[P′] Minimize $\sum_{j \in N \backslash C} f_j \cdot y_j + \sum_{\{(i,j) \in A | i \in N \backslash D \text{ and } j \in N \backslash C\}} c_{ij} \cdot w_i \cdot d_{ij} \cdot x_{ij}$

subject to

$$\sum_{j \in N \backslash C} x_{ij} = 1 \qquad \text{for all } i \in N \backslash D \qquad (6)$$

$$\sum_{i \in N \backslash D} w_i \cdot x_{ij} \le Q_j \cdot y_j \qquad \text{for all } j \in N \backslash C \qquad (7)$$

$$d_{ij} \cdot x_{ij} \le S_j \cdot y_j \qquad \text{for all } i \in N \backslash D \text{ and } j \in N \backslash C \qquad (8)$$

$$y_j \in \{0, 1\} \qquad \text{for all } j \in N \backslash C \qquad (9)$$

$$x_{ij} \in \{0, 1\} \qquad \text{for all } i \in N \backslash D \text{ and } j \in N \backslash C \qquad (10)$$

To obtain a lower bound, the reduced problem [P′] is solved after the maximum allowable distance constraint (10) is relaxed. Note that the relaxed problem must be solved optimally to obtain a valid lower bound. For this purpose, we solve the relaxed problem for all possible sets of collection points. For a given set Y^* of collection points, i.e., $Y^* = \{y_j^* | j \in N \backslash C\}$, the relaxed problem can be reformulated as follows. Note that the first term (fixed cost) in the objective function is a constant and hence can be removed in the reformulated problem.

[RP′] Minimize $Z_{RP'}(Y^*) = \sum_{\{(i,j) \in A | i \in N \backslash D \text{ and } j \in N \backslash C\}} c_{ij} \cdot w_i \cdot d_{ij} \cdot x_{ij}$

subject to

$$\sum_{i \in N \backslash D} w_i \cdot x_{ij} \le Q_j \cdot y_j^* \qquad \text{for all } j \in N \backslash C \qquad (11)$$

$$x_{ij} \in \{0, 1\} \qquad \text{for all } i \in N \backslash D \text{ and } j \in N \backslash C \qquad (12)$$

To solve the reformulated problem [RP′] optimally, we decompose the entire problem into the subproblems for opened collection points, i.e., $\{j | \ y_j^* = 1 \text{ for } j \in N \backslash C\}$, and each subproblem is solved using the dynamic programming algorithm for the 0-1 knapsack problem. (See Martello and Toth (1990) for the dynamic programming algorithm.) The second term Z_R of the LB1 can be represented as follows.

$$Z_R = \min_{\forall \mathbf{y}^*} \{ \sum_{j \in N \backslash C} f_j \cdot y_j^* + Z_{RP'}(\mathbf{y}^*) \}$$

The second lower bound (LB2) is the same as LB1 except that the problem [RP′] is solved after the integrality constraint is relaxed. Note that the problem [RP′] without the integrality constraint is the fractional knapsack problem which can be solved optimally using the greedy algorithm, i.e. assign demand points to collection points in the non-decreasing order of $c_{ij} \cdot w_i \cdot d_{ij} / w_i$ (= $c_{ij} \cdot d_{ij}$). Note that LB2 is a valid lower bound since the solution value of the minimization version of the fractional knapsack problem gives a valid lower bound of the corresponding 0-1 knapsack problem. Although LB2 is always less than or equal to LB1, LB2 has a certain merit in computation time and hence it may work well for the entire B&B algorithm.

(*c*) *Using a Dominance Property*

The search space can be reduced further using a simple dominance property. The basic idea is that each node of the B&B tree can be fathomed if the solution value of the partial solution corresponding to the node is greater than or equal to the upper bound obtained so far. This is a valid dominance property since the B&B tree has the monotonic non-decreasing property, i.e. the solution values of partial and complete solutions do not decrease as the level of the B&B tree increases.

4.2 Heuristic Algorithms

Since the optimal B&B algorithm has limited applications due to its heavy computational burden, we suggest two heuristics that give solutions for large-sized instances. They are: (a) B&B heuristic; and (b) two-stage heuristic. The two heuristics are based on decomposing the entire problem into the location and the allocation sub-problems.

4.2.1 B&B Heuristic

The B&B heuristic is the same as the optimal B&B algorithms except that location and allocation are done separately. In other words, location is done using a binary tree and allocation is done using the dynamic programming algorithm for the 0-1 knapsack problem.

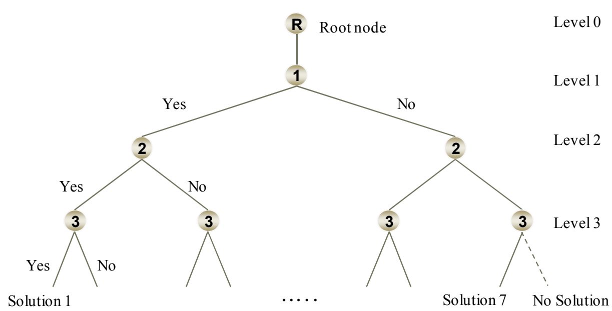

Figure 4. Binary Tree: an Example with 3 Potential Sites

In the binary tree, each level corresponds to the index of potential sites. There is a single root node (level 0) and a (parent) node of level l ($l < n$) has two child nodes of level l+1 that correspond to solutions with $y_{l+1} = 1$ and $y_{l+1} = 0$. The node on the left corresponds to a subproblem in which y_{l+1} is fixed to be 1, while $y_{l+1} = 0$ for the node on the right. Recall that $y_l = 1$ if a collection point is opened at potential site l, and 0 otherwise. Each node at level l represents a partial solution that corresponds to a set of opened collection points, which can be found by tracing a path from that node to the root node. As in the optimal B&B algorithm, the depth-first search is used for node selection. Figure 4 shows the binary tree for a problem with three potential sites (n = 3) in which there are 2^3-1 (= 7) possible locations of collection points.

At each node of the binary B&B tree, the allocation problem can be formulated as the binary integer programming model given below. In the formulation, C_{rl} denotes the set of collection points opened after node r of level l is branched.

[AP$_{rl}$] Minimize $\sum_{i \in N} \sum_{j \in C_{rl}} c_{ij} \cdot w_i \cdot d_{ij} \cdot x_{ij}$

subject to

$$\sum_{j \in C_{rl}} x_{ij} = 1 \quad \text{for all } i \in N \quad (13)$$

$$\sum_{i \in N} w_i \cdot x_{ij} \le Q_j \quad \text{for all } j \in C_{rl} \quad (14)$$

$$d_{ij} \cdot x_{ij} \le S_j \quad \text{for all } i \in N \text{ and } j \in C_{rl} \quad (15)$$

$$x_{ij} \in \{0, 1\} \quad \text{for all } i \in N \text{ and } j \in C_{rl} \quad (16)$$

The allocation problem [AP$_{rl}$] at each node of the binary tree is solved heuristically using the decomposition method after the maximum allowable collection distance constraint is relaxed. As in the optimal B&B algorithm, the decomposition is done for each of the collection points specified at each node of the binary tree. Then, we can easily see that each decomposed problem is the 0-1 knapsack problem that can be solved using the dynamic programming algorithm.

Since the maximum allowable collection distance constraint is relaxed, it is needed to check the feasibility of the solution obtained. If the current solution is infeasible, the demand points violating the distance constraint are reallocated to the collection point that gives the minimum variable cost. Finally, the solution is improved further by reallocating each demand point to the collection point that gives the maximum improvement. The procedure for the B&B heuristic is omitted here since it is similar to the optimal B&B algorithm.

4.2.2 Two-Stage Heuristic

The second heuristic, an extension of the algorithm of Kim and Lee (2008), consists of two stages: (a) locating collection points using a clustering method; and (b) allocating demand points to collection points using a heuristic for the generalized assignment problem (GAP) with the additional maximum allowable distance constraint. Note that the GAP with the additional constraint is the same as the problem [AP$_{rl}$] except that the location is done for all potential sites, not partial potential sites. In overall, the two-stage heuristic has a certain merit over the B&B heuristic since it is very useful when the computation time is critical.

The two-stage heuristic algorithm works as follows. First, the potential sites are clustered in the non-decreasing order of the adjacency index while considering the capacity constraint, where the adjacency index for a pair of potential sites (i, j) is defined as $c_{ij} \cdot w_i \cdot d_{ij}$. Then, the collection point is selected for each cluster in such a way that it gives the minimum total variable cost. Finally, the allocation problem for the given set of collection points, i.e. the GAP with the additional maximum allowable distance constraint, is solved using the modification of the MTHG algorithm. See Martello and Toth (1990) for more details on the MTHG algorithm.

Now, the detailed procedure for the two-stage heuristic can be summarized as follows.

Procedure (Two-stage heuristic)

Step 1: (Locating collection points)

(a) For each pair of potential sites (i, j), compute the adjacency index. Then, sort the pairs in the non-decreasing order, and make a list according to this order.

(b) Starting from the top of the list, determine whether the two potential sites in the current pair can be merged without violating the capacity constraint. If such points exist, merge them. Otherwise, consider the next one.

(c) For each cluster, determine the collection point that gives the minimum total variable cost.

Step 2: (Allocating demand points to collection points)

Step 2.1 (Initialization)

(a) For all unallocated demand points, find the one (denoted by point i) that has the maximum difference between the largest and the second largest values of g_{ij}, where $g_{ij} = w_i \cdot d_{ij}$.

(b) Allocate demand point i to collection point j for which g_{ij} has the minimum value without violating the capacity and the maximum allowable collection distance constraints. If collection point j cannot be allocated, do this step for the collection point that has the second minimum value of the desirability measure, and so on.

(c) If all demand points are allocated, save the initial solution and go to Step 2. Otherwise, go to step (a).

Step 2.2 (Improvement)

(a) For each demand point, find the set A_i of collection points that improves the initial solution while satisfying their capacity and maximum allowable distance constraints. Let demand point i be allocated to collection point j^* in the initial solution, i.e. $x_{ij^*} = 1$. Then, the set A_i can be described as follows.

$$A_i = \{ j \ne j^* \mid g_{ij} < g_{ij}^* \text{ and } w_i \le R_j \text{ and } d_{ij} \le S_j \},$$

where R_j and S_j denote the remaining capacity and the maximum allowable collection distance of collection point j, respectively. If $A_i = \varnothing$ for all i, i.e. no improvement can be made, stop the algorithm. Otherwise, go to Step (b).

(b) Perform the reallocation that gives the maximum amount of improvement after evaluating all possible candidates. Here, reallocation implies that demand point i is assigned to the set A_i of collection points. Repeat this step until there is no improvement.

5. COMPUTATIONAL RESULTS

To test the performances of the optimal and heuristic algorithms suggested in this study, a series of computational experiments were done on various test instances and the results are reported in this section. The algorithms were coded in C and run on a workstation with a Xeon processor operating at 3.2 GHz clock speed.

In the test, the algorithms were tested in two aspects. For the optimal B&B algorithm, we tested two types, i.e. B&B1 with LB1 and B&B2 with LB2, and report: (a) the number of instances that the algorithm gave the optimal solutions within a time limit of one hour (NOS); and (b) CPU seconds. For the heuristic algorithms, the solution qualities are shown with: (a) percentage gaps from the optimal solutions for the instances that can be solved by the optimal B&B algorithms (GAP); (b) relative performance ratio (RPR) for large sized instances. The relative performance ratio of heuristic a for an instance is defined as.

$$[(Z_a - Z_{best}) / Z_{best}] \cdot 100 \ (\%),$$

where Z_a is the objective function value obtained from heuristic a, i.e. one of the two heuristics suggested in this study and Z_{best} is the better one of the objective values obtained from the two heuristics. Also, we report the number of instances that the heuristic algorithms gave the optimal solutions.

The algorithms were tested on the instances with different number of potential sites. For the test on the optimal B&B algorithm, 40 instances were generated randomly, i.e. 10 instances for each of four levels of the number of potential sites (10, 20, 30, and 40). For the comparison of the two heuristics, 60 additional instances were generated, i.e. 10 instances for each of six levels of number of potential sites (50, 100, 200, 300, 400, and 500). The location of each potential site was randomly generated on a square-shaped area with a size of 100×100 unit length, and the distances between two potential sites were obtained by calculating its rectilinear (Manhattan) distance. The amount of refuse at each demand point was generated from DU (10, 50), where DU (a, b) is the discrete uniform distribution with range [a, b]. The capacity and the maximum allowable collection distance of each collection point were generated from DU (40, 100) and DU (140, 200), respectively. Finally, the fixed cost to open a collection point and the unit variable cost for transporting one unit of refuse were set to DU (100000, 150000) and 10, respectively.

Test results for the optimal B&B algorithms are shown in Table 1 which summarizes: (a) the number of instances that the B&B algorithms gave the optimal solutions; (b) the number of branched nodes; and (c) CPU seconds (average, minimum and maximum). It can be seen from the table that both algorithms give the optimal solutions for most test instances up to 30 potential sites. However, the computation time increases sharply when the number of potential sites becomes 30. In this test, the optimal B&B algorithms were not compared with CPLEX 10.1, a commercial software package, since it could not solve the instances even with 15 potential sites. Of the two algorithms, B&B2 was slightly faster than B&B1 although LB2 is always less than or equal to LB1 (and hence B&B2 branched more nodes). This is because the time to obtain LB2 is much shorter than LB1 and hence the overall CPU seconds could be reduced. Although it is not needed to report here, the initial upper bound was not very effective since it was obtained by a simple greedy algorithm. Also, the dominance property was slightly effective for reducing the search space. In fact, we could reduce the computation time about 10% in overall average.

Table 2 summarizes the test results on two heuristics (B&B and two-stage heuristics). Table 2(a) shows their performances for the small sized instances that the B&B algorithms gave the optimal solutions. Of the two heuristics, the B&B heuristic outperformed the two-

Table 1. Performances of the Optimal Branch and Bound Algorithms

Number of potential sites	B&B1 (B&B with LB1)			B&B2 (B&B with LB2)		
	NOS[1]	NBN[2]	CPU[3]	NOS	NBN	CPU
10	10	179.3	1.1 (0.3 , 3.8)	10	272.6	0.7 (0.1 , 5.0)
20	10	37776.7	184.6 (35.3 , 1153.6)	10	47904.8	143.1 (33.3 , 1069.6)
30	9	665043.1	2953.5 (2876.9 , 3544.1)	10	7587442.0	2638.3 (535.3 , 3186.6)
40	4	776117.7	3535.8 (3454.8 , 3577.2)	5	9584232.6	2942.1 (2636.7, 35780.3)
Average	8.3	369779.2	1668.8 (1591.8, 2069.7)	8.8	4304963.0	1431.1 (801.4, 10009.3)

1 number of instances that the B&B algorithm gave the optimal solutions out of 10 instances.
2 number of branched nodes (averaged out of 10 instances).
3 average CPU seconds (minimum, maximum).

stage heuristic since it considers all possible alternatives for locating collection points. Also, we found that both algorithms give near optimal solutions for small sized test instances. In fact, the overall average gaps of the B&B and the two-stage heuristics were 2.04% and 2.84%, respectively. However, the B&B heuristic required much longer CPU seconds than the two-stage heuristic.

Test results for medium-to-large sized instances with 50 to 500 potential sites are summarized in Table 2(b). In overall, the test results were very similar to those for small sized instances. In other words, the B&B heuristic outperformed the two-stage heuristic in overall average. In particular, the B&B heuristic dominated the two-stage heuristic for the instances with more than 100 potential sites, which can be seen from the number of instances for which each heuristic found the better solutions. However, as in the results on small sized instances, the B&B heuristic required much longer CPU seconds. Therefore, we recommend the two-stage heuristic for large sized practical applications in which the computation time is very critical.

Table 2. Performances of the Heuristic Algorithms
(a) Results for Small Sized Instances

Number of potential sites	B&B heuristic			Two-stage heuristic		
	N_{opt} [1]	GAP [2]	CPU [3]	N_{opt}	GAP	CPU
10	9	0.16	0.0	6	0.64	0.0
20	6	1.40	0.0	2	1.55	0.0
30	4	2.34	0.0	3	3.64	0.0
40	1	4.26	0.4	0	5.55	0.0
Average	4.5	2.04	0.1	2.5	2.84	0.0

[1] number of instances that the heuristic algorithm found the optimal solutions out of 10 instances (This value was obtained only for instances for which optimal solutions are obtained).
[2] average percentage gap from the optimal solutions out of 10 instances (This value was obtained only for instances for which optimal solutions are obtained).
[3] average CPU seconds out of 10 instances.

(b) Results for Medium-to-Large Sized Instances

Number of potential sites	B&B heuristic			Two-stage heuristic		
	N_{best} [1]	RPR [2]	CPU [3]	N_{best}	RPR	CPU
50	7	0.29	0.6	3	0.44	0.0
100	10	0.00	4.1	0	1.04	0.0
200	10	0.00	32.3	0	1.44	0.1
300	10	0.00	125.9	0	1.44	0.2
400	10	0.00	289.1	0	1.26	0.4
500	10	0.00	415.6	0	1.43	0.8
Average	9.5	0.05	75.4	0.5	1.17	0.3

[1] number of instances that the heuristic algorithm gave the better solutions out of 10 instances.
[2] average relative performance ratio out of 10 instances.
[3] average CPU seconds out of the 10 instances.

6. CONCLUDING REMARKS

In this study, we considered the problem of designing refuse collection networks, one of important decisions in reverse logistics. The problem is to determine the locations of collection points and the allocations of refuses at demand points to the opened collection points for the objective of minimizing the sum of fixed costs to open collection points and variable costs to transport refuses while satisfying the capacity and the maximum allowable distance constraints at each collection point. As an extension of the previous models, the maximum allowable distance constraint is additionally considered, so that one can obtain practical solutions in which collection points are not located too closely. After an integer programming model is provided, we suggested optimal and heuristic algorithms. The optimal algorithm was based on the branch and bound technique that incorporates a new branching scheme with simultaneous location and allocation, and two heuristics were also suggested to solve large sized instances. Computational experiments were done on various test instances and the results showed that the branch and bound algorithm gave the optimal solutions for the test instances up to 30 potential sites within 3600 seconds. Of the two heuristics, the branch and bound heuristic was better than the two-stage heuristic. However, the two-stage heuristic is more practical as the problem size increases due to its shorter computation time.

There are several directions to extend this research. First, it is needed to extend the model by considering multi-level reverse logistics networks that consist of demand points, collection points, and recycling/remanufacturing/disposal facilities. In this case, the model suggested in this study can be used as a building block for designing the entire network. Second, a case study on a refuse collection system is needed to validate the collection network design model suggested in this study. Finally, the model can be integrated with the operational problem such as vehicle routing and scheduling.

ACKNOWLEDGEMENTS

This work was supported by the Korea Science and Engineering Foundation (KOSEF) granted funded by Korea government (MEST) (Grant Code: 2009-0074736).

REFERENCES

Ackali, E., S. Çetinkaya, and H. Üster, "Network design for reverse and closed-loop supply chains: an annotated bibliography of models and solution approaches," *Network*s 53 (2008), 231-248.

Aras, N. and D. Aksen, "Locating collection centers for distance and incentive-dependent returns," *Interna-*

tional Journal of Production Economics 11 (2008), 316-333.

Aras, N., D. Aksen, and A. G. Tanugur, "Locating collection centers for incentive dependent returns under a pick-up policy with capacitated vehicles," *European Journal of Operational Research* 191 (2008), 1223-1240.

Barros, A. I., R. Dekker, and V. Scholten, "A two-level network for recycling sand: a case study," *European Journal of Operational Research* 110 (1998), 199-214.

Bautista, J. and J. Pereira, "Modeling the problem of locating collection areas for urban waste management: an application to the metropolitan area of Barcelona," *Omega* 34 (2006), 617-629.

Berger, R. T., C. R. Coullard and M. S. Daskin, "Location-routing problems with distance constraints," *Transportation Science* 45 (2007), 29-43.

Bloemhof-Ruwaard, J. M., M. Fleischmann, and J. A. E. E. van Nunen, "Reviewing distribution issues in reverse logistics," *Lecture Notes in Economics and Mathematical Systems* 480 (1999), 23-44.

Caruso, C., A. Colorni, and M. Paruccini, "The regional urban solid waste management system: a modeling approach," *European Journal of Operational Research* 70 (1993), 16-30.

Cruz-Rivera, R. and J. Ertel, "Reverse logistics network design for the collection of end-of-life vehicles in Mexico," *European Journal of Operational Research* 196 (2009), 930-939.

Dekker, R., J. M. Bloemhof-Ruwaard, M. Fleischmann, J. Van Nunen, E. Van Der Laan, and L. N. Van Wassenhove, "Operational research in reverse logistics: some recent contributions," *International Journal of Logistics Research and Applications* 1 (1998), 141-155.

Dowlatshahi, S., "Developing theory of reverse logistics" *Interfaces* 30 (2000), 143-151.

Fleischmann, M., J. M. Bloemhof-Ruwaard, R. Dekker, E. van der Laan, J. A. E. E. van Nunen, and L. N. van Wassenhove, "Quantitative models for reverse logistics: a review," *European Journal of Operational Research* 103 (1997), 1-17.

Fleischmann, M., H. R. Krikke, R. Dekker, and S. D. P. Flapper, "A characterization of logistics networks for product recovery," *Omega* 28 (2000), 653-666.

Fleischmann, M., P. Beullens, J. M. Bloemhof-Ruwaard, and L. N. van Wassenhove, "The impact of product recovery on logistics network design," *Production and Operations Management* 10 (2001), 156-173.

Ilgin, M. A. and S. M. Gupta, "Environmentally conscious manufacturing and product recovery (ECMPRP): a review of the state of the art," *Journal of Environmental Management* 91 (2010), 563-591.

Jayaraman, V., V. D. R. Guide Jr., and R. Srivastava, "A closed-loop logistics model for remanufacturing," *Journal of the Operational Research Society* 50 (1999), 497-508.

Jayaraman, V., R. A. Patterson, and E. Rolland, "The design of reverse distribution networks: models and solution procedures," *European Journal of Operational Research* 150 (2003), 128-149.

Kara, S., F. Rugrungruang, and H. Kaebernick, "Simulation modeling of reverse logistics networks," *International Journal of Production Economics* 106 (2007), 61-69.

Kim, J.-S. and D.-H. Lee, "Heuristic algorithms for capacitated collection network design in reverse logistics," *International Journal of Management Science* 12 (2008), 45-66.

Kim, J.-S., H.-S. Choi, and D.-H. Lee, "Search heuristics for capacitated refuse collection network design in reverse logistics," *Journal of Korea Industrial and Systems Engineering* 33 (2010), 41-50.

Klose, A. and A. Drexl, "Facility location models for distribution system design," *European Journal of Operational Research* 162 (2005), 4-29

Ko, H. J. and G. W. Evans, "A genetic algorithm-based heuristic for the dynamic integrated forward/reverse logistics network for 3PLs," *Computers and Operations Research* 34 (2007), 346-366.

Krikke, H. R., E. J. Kooi, and P. C. Schuur, "Network design in reverse logistics: a quantitative model," *Lecture Notes in Economics and Mathematical Systems* 480 (1999), 45-62.

Kroon, L. and G. Vrijens, "Returnable containers: an example of reverse logistics," *International Journal of Physical Distribution and Logistics Management* 25 (1995), 56-68.

Lee, D. H. and M. Dong, "A heuristic approach to logistics network design for end-of-lease computer products recovery," *Transportation Research Part E: Logistics and Transportation Review* 44 (2008), 455-474.

Lee, D. H. and M. Dong, "Dynamic network design for reverse logistics operations under uncertainty," *Transportation Research Part E: Logistics and Transportation Review* 45 (2009), 61-71.

Lee, D.-H., H.-J. Kim, and J.-S. Kim, "Reverse logistics: research issues and literature review," *Journal of the Korean Institute of Industrial Engineers* 34 (2008), 270-288.

Lieckens, K. and N. Vandaele, "Reverse logistics network design with stochastic lead times," *Com-*

puters and Operations Research 34 (2007), 395-416.

Louwers, D., B. J. Kip, E. Peters, F. Souren, and S. D. P. Flapper, "A facility location allocation model for reusing carpet materials," *Computers and Industrial Engineering* 36 (1999), 855-869.

Lu, Z. and N. Bostel, "A facility location model for logistics systems including reverse flows: the case for remanufacturing activities," *Computers and Operations Research* 34 (2007), 299-323.

Mansini, R. and M. G. Speranza, "A linear programming model for the separate refuse collection service," *Computers and Operations Research* 25 (1998), 659-673.

Martello, S. and P. Toth, *Knapsack Problems: Algorithms and Computer Implementations*, John Wiley and Sons, 1990.

Melo, M. T., S. Nickel and F. Saldanha-da-Gama, "Facility location and supply chain management-a review," *European Journal of Operational Research* 196 (2009), 401-412.

Min, H., H. J. Ko, and C. S. Ko, "A genetic algorithm approach to developing the multi-echelon reverse logistics network for product returns," *Omega* 34 (2006a), 56-69.

Min, H., C. S. Ko, and H. J. Ko, "The spatial and temporal consolidation of returned products in a closed-loop supply chain network," *Computers and Industrial Engineering* 51 (2006b), 309-320.

Pati, R. K., P. Vrat, and P. Kumar, "A goal programming model for paper recycling system," *Omega* 36 (2008), 405-417.

Sahyouni, K., R. C. Savaskan, and M. S. Daskin, "A facility location model for bidirectional flows," *Transportation Science* 41 (2007), 484-499.

Salema, M. I. G., A. P. Barbosa-Povoa, and A. Q. Novais, "An optimization model for the design of a capacitated multi-product reverse logistics network with uncertainty," *European Journal of Operational Research* 179 (2007), 1063-1077.

Sasikumar, P. and G. Kannan, "Issue in reverse supply chains, part I: end-of-life product recovery and inventory management - an overview," *International Journal of Sustainable Engineering*, 1 (2008a), 154-172.

Sasikumar, P. and G. Kannan, "Issue in reverse supply chains, part II: reverse distribution issues-an overview," *International Journal of Sustainable Engineering* 1 (2008b), 234-249.

Sim, E. S., S. W Jung, H. J. Kim, and J. W. Park, "A generic network design for a closed-loop supply chain using genetic algorithm," *Lecture Notes in Computer Science* 3103 (2004), 1214-1225.

Spengler, Th., H. Püchert, and P. O. Rentz, "Environmental integrated production and recycling management," *European Journal of Operational Research* 97 (1997), 308-326.

Srivastava, S. K., "Green supply chain management: a state-of-the-art literature review," *International Journal of Management Review* 9 (2007), 53-80.

Vesilind, A. P., W. Worell, and D. Reinhart, *Solid Waste Engineering*, Brook/Cole, 2002.

Wang I. L. and W. C. Yang, "Fast heuristics for designing integrated e-waste reverse logistics networks," *IEEE Transactions on Electronics Packaging Manufacturing* 30 (2007), 147-154.

Wojanowski, R., V. Verter, and T. Boycai, "Retail-collection network design under deposit-refund," *Computers and Operations Research* 34 (2007), 324-345.