International Journal of Management Science
Vol 14, No 2, November 2008

Heuristic Algorithms for Capacitated Collection Network Design in Reverse Logistics[*]

Ji-Su Kim
Department of Industrial Engineering Hanyang University
Seongdong-gu, Seoul 133-791 Korea

Dong-Ho Lee[**]
Department of Industrial Engineering Hanyang University
Seongdong-gu, Seoul 133-791 Korea

(Received: May 25, 2008 / Revised: September 2, 2008 / Accepted: September 4, 2008)

ABSTRACT

Refuse collection, one of important elements in reverse logistics, is an activity rendering recyclables or wastes and moving them to some points where further treatment is required. Among various decisions in the collection activity, we focus on network design, which is the problem of locating collection points as well as allocating refuses at demand points to collection points while satisfying the capacity restriction at each collection point. Here, the collection point is the place where recyclables or wastes near the point are gathered, and locating the collection points is done by selecting them from a given set of potential sites. The objective is to minimize the sum of fixed costs to open collection points and transportation costs to move refuses from demand points to collection points. An integer programming model is developed to represent the problem mathematically and due to the complexity of the problem, two types of heuristics, one with simultaneous and the others with separate location and allocation, are suggested. Computational experiments were done on test problems up to 500 potential sites, and the results are reported. In particular, some heuristics gave near optimal solutions for small-size test problems, i.e., 2% gaps in average from the optimal solution values.

Keywords: Reverse Logistics, Refuse Collection, Network Design, Heuristics

[*] This work was supported by Brain Korea 21 Grant funded by Korean Government. This is gratefully acknowledged.
[**] Corresponding author, E- mail: leman@hanyang.ac.kr

1. Introduction

There have been various environmental issues for manufacturing firms and municipalities due to legislation pressures and customer recognitions to protect the environment, while imposing the obligations to collect and upgrade used or end-of-life products in an environmentally conscious way. One of fundamental directions to deal with environmental issues is the originator principle, i.e., he (she) who inflicts harm on the environment should pay for fixing the damage. For example, see EuP (Eco-design requirements for Energy using Products), WEEE (Waste Electrical and Electronic Equipment), RoHS (Restriction of the use of certain Hazardous Substance in electric and electronic equipment), and ELV (Directives for End-of-Life Vehicle).

Among the environmental issues, those for used products and wastes are increasingly important because of shortages of dumping sites and waste incineration facilities, shortened lifetime of products, etc. This gives rise to additional material flows of collecting and reprocessing used or end-of-life products and wastes. To manage those flows, the concept of reverse logistics has been emerged. Reverse logistics, which is the opposite direction of the conventional forward logistics, can be defined as the logistics activities all the way from used or end-of-life products no longer required by the user to products again usable in a market or waste disposal [9].

A brief description of the reverse logistics is shown in Figure 1, adopted from Bloemhof-Ruwaard *et al.* [2]. In this figure, collection implies an activity that gathers used or end-of-life products for recovery or disposal, recycling implies material recovery without conserving product structures, and remanufacturing is transformation of used or end-of-life products into units that satisfy the quality of new products or other standards. For more details on reverse logistics, see Fleischmann *et al.* [9, 10], Dowlatshahi [6], Guide *et al.* [12], Ferguson and Browne [8], and Lee *et al.* [22]. While the specific activities involved differ per case, the typical functions of reverse logistics include collection, transportation, testing, disassembly, recovery, and disposal. Also, unlike the forward logistics, reverse logistics has the convergent structure from many sources to few demand points and high degree of uncertainty in supply both in terms of quantity and quality of used products returned by consumers [9].

Among various activities in reverse logistics, this paper focuses on refuse collection, i.e., an activity rendering used products or wastes and moving them to some points where further treatment is required. Note that refuse collection, a reverse of distribution in forward logistics, is one of important activities in reverse logistics since product recovery or waste disposal cannot be performed without collecting used or end-of-life products or waste. Therefore, we can see that constructing an efficient refuse collection system is one of fundamental issues in reverse logistics for manufacturing and service firms or municipalities.

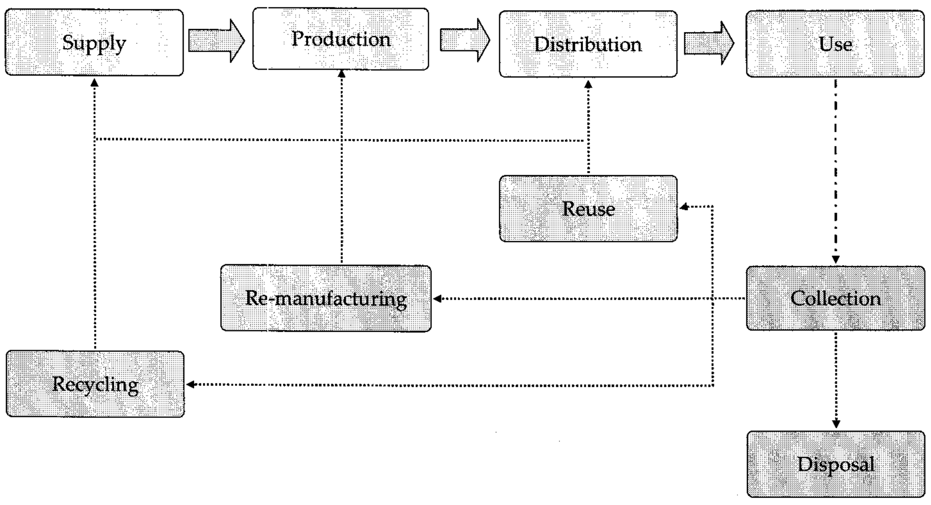

Figure 1. A simple description of forward and reverse logistics

There are various decision problems in refuse collection systems, which can be classified into those in strategic, tactical and operational levels [25]. The decisions in the strategic level include the choice on the types of facilities and their locations, while those in the tactical level include the decisions on the type of services offered to customers. Also, the operational decisions are concerned with the routing and scheduling of collection vehicles. Note that the three types of decisions should be integrated as a whole in order to construct an efficient refuse collection system.

Among the decision problems in refuse collection systems, this paper addresses the problem of designing collection networks, i.e., locating collection points as well as allocating refuses at demand points to collection points while satisfying the capacity restriction at each collection point. The collection points, where recyclables or wastes near the point are gathered, are determined by selecting them from a given set of potential sites. As in the distribution network design problem for forward logistics, the collection network design problem is one of core elements in designing reverse logistics. To represent the problem mathematically, we first suggest an integer programming model. Then, due to the complexity of the problem, two types of heuristic algorithms, one with simultaneous and the others with separate location and allocation, are suggested. To show the performances of the heuristics, computational experiments were done on test problems up to 500 potential sites, and the results are reported.

This paper is organized as follows. In the next section, the literature review is done on the relevant research articles. In Section 3, the problem is described in more detail with an integer programming model. A comparison with the traditional distribution network design problems is also explained. The two types of heuristics are suggested in Section 4, and Section 5 reports the computational results. Finally, Section 6 gives conclusions with summarizing the paper and describing some areas for further research.

2. Literature Review

Most previous research articles on designing reverse logistics networks are case studies on reusing returnable containers, remanufacturing used or end-of-life products, recycling materials, disposing waste, etc. Caruso *et al.* [4] consider a multi-objective capacitated location-allocation problem for waste service users, processing plants, and sanitary landfills for an urban solid waste management system in an Italian region, and suggest heuristic algorithms after formulating it as an integer programming model, and Kroon and Vrijens [19] suggest an integer programming model for a network design problem for returnable containers and solve it using an existing optimal solution algorithm for an uncapacitated facility location model. Spengler *et al.* [32] consider a capacitated network design problem for by-product recycling of steel and iron industry in Germany, and suggest a mixed integer programming model, and Barros *et al.* [1] suggest heuristic algorithms for the capacitated two-level location problem for sand recycling in Netherlands. Louwers *et al.* [24] consider a design problem for carpet recycling networks, and suggest an optimal solution algorithm using their non-linear mathematical model. They also perform case studies occurred in Europe and USA. Krikke *et al.* [20] develop an uncapacitated multi-echelon network design model for reverse logistics, and suggest a mixed integer programming model. In their paper, a case study was done on collecting, processing, and delivering end-of-life automobiles. See Jayaraman *et al.* [13], Krikke *et al.* [21], Lee and Dong [23], and Realff *et al.* [29] for other case studies.

Unlike the above case studies, some articles consider the theoretical points on network design problems in reverse logistics. Jayaraman *et al.* [14], as an extension of Jayaraman *et al.* [13], suggest heuristic algorithms for the two-level hierarchical location problem that determines the numbers and locations of collection points and refurbishing facilities for hazardous products. Min *et al.* [27] consider the problem of determining the numbers, locations, and sizes of collection points and centralized return centers under capacity limits and service requirements, and suggest genetic algorithms after providing a mixed integer nonlinear programming model. Fleischmann *et al.* [11] consider the problem of designing the forward distribution and reverse recovery networks, and suggest a mixed integer linear programming model that extends the traditional warehouse location problem. Recently, Salema *et al.* [31] generalized the model of Fleischmann *et al.* [11] by considering multiple product types with uncertainty in demand and return flows, and suggested a branch and bound algorithm that minimizes the sum of relevant costs. Also, Ko and Evans [18] suggested a genetic algorithm-based heuristic for designing forward and return networks in an integrated aspect. In fact, their problem is a class of the multi-period, two-echelon, multi-commodity, capacitated location model.

3. Problem Description

This section describes the collection network design problem considered in this paper, together with an integer programming model. A comparison with the traditional distribution network design problems is also done.

Before presenting the problem, we describe the refuse collection system. Figure 2, adopted from Kim *et al.* [16], shows a refuse collection area of a district of Seoul, South Korea. The district office operates its refuse collection system in such a way that its region is decomposed into several areas and a vehicle, together with two or three workers, is assigned to each area. As can be seen in the figure, the area has fourteen collection points and demand points near each collection point. As defined earlier, each collection point is the place where the refuses at the demand points are gathered, and the vehicle collects gathered refuses by visiting the collection points.

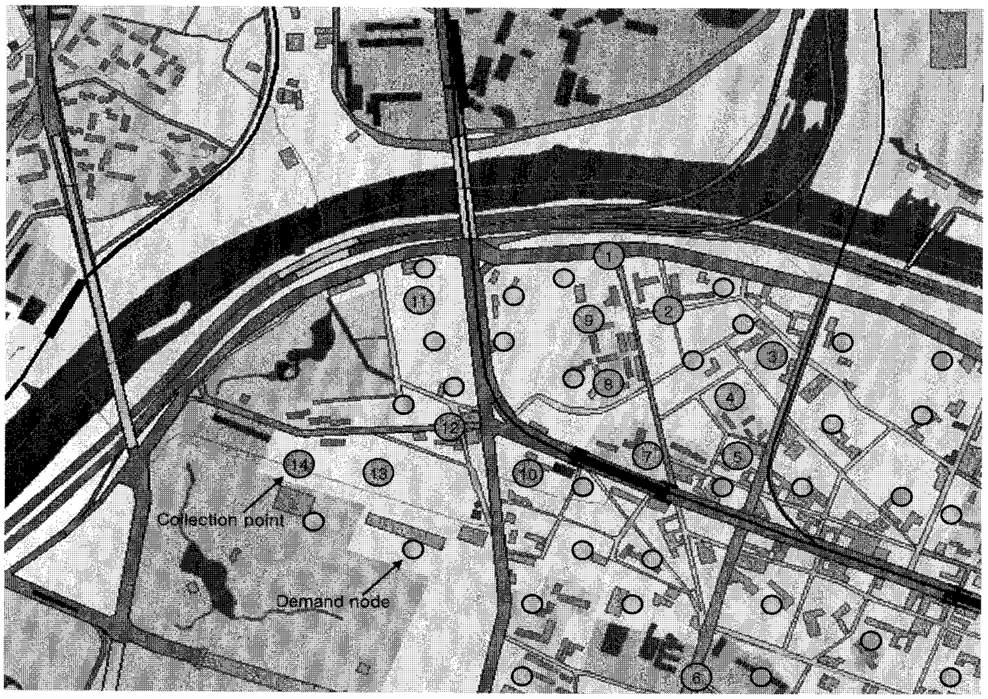

Figure 2. A refuse collection system: example

There are two decisions in the collection network design problem considered in this paper. They are: (a) locating collection points; and (b) allocating refuses at demand points to collection points. As stated earlier, the location is done by selecting them from a given set of potential sites, and the refuse demand occurs at each potential site. (Note that the potential sites correspond to the demand points.) It is assumed that the amount of refuse at each demand point is deterministic and given in advance. The objective is to minimize the sum of fixed costs to open collection points and transportation costs to move refuses at demand points to collection points. It is assumed that the fixed and transportation costs are deterministic and given in advance.

The main constraint is the capacity restriction at each collection point, i.e., the limit in the amount of refuse assigned to the collection point. In other words, there is an upper limit to assign refuse demands to each collection point, and the demand at each demand point consumes a portion of the available capacity of the corresponding collection point. It is assumed that the available capacity at each collection point is given in advance. Other assumptions made in the problem are summarized as follows: (a) each demand point is allocated to exactly one collection point, i.e., no demand splitting is allowed; (b) it is not possible to assign two or more collection points to a potential site; (c) distances are symmetric; (d) each potential site has the same fixed cost for opening a collection point; and (e) transportation cost is proportional to the amount of refuse.

Compared with the previous research articles on designing the entire reverse logistics networks that give ambiguous network structures, this paper narrows the system scope to the refuse collection activity so that a more concrete model is provided for designing collection networks. Also, in the theoretical aspect, we define the most basic form of the collection network design problem. Therefore, our model can be used as a building block for designing the entire reverse logistics network since it includes the most essential features of collection networks, i.e., locating collection points and allocating refuse demands to collection points.

To represent the problem more clearly, an integer programming model is suggested. First, we define a complete graph $G = (N, A)$, where $N = \{1, 2, \cdots, n\}$ and $A = \{(i, j): i \neq j\}$ denote the sets of nodes and arcs, respectively. Note that each node in the graph corresponds to a potential site for opening a collection point or generating a refuse demand. Before presenting the model, the notations used are summarized below.

• ***Parameters***

w_i	amount of refuse to be collected at node i
d_{ij}	distance between nodes i and j
c_{ij}	transportation cost (per unit and unit distance) from node i to j
f_j	fixed cost for opening a collection point at node j
Q_j	capacity at the collection point opened at node j

• ***Decision variables***

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

Now, the integer programming model for the collection network design problem is given below.

[P] Minimize $\sum_{j\in N} f_j \cdot y_j + \sum_{i\in N}\sum_{j\in N} w_i \cdot d_{ij} \cdot c_{ij} \cdot x_{ij}$ (1)

subject to

$\sum_{j\in N} x_{ij} = 1$ for all i (2)

$x_{ij} \le y_j$ for all i and j (3)

$\sum_{i\in N} w_i \cdot x_{ij} \le Q_j \cdot y_j$ for all j (4)

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

$y_j \in \{0,1\}$ for all j (6)

The objective function denotes the sum of the fixed costs to open collection points and the transportation costs to move refuses at demand points to collection points. Here, the transportation costs are computed using the amounts of refuses at demand points and the distances between collection points and demand points. Constraint set (2) ensures that each demand point be assigned to one and only one collection point, and constraint set (3) implies that demand point i can be assigned to collection point j only if there is an opened collection point at potential site j. In other words, it states that no demand point can be assigned to collection point j unless there is a collection point opened there. Constraint set (4) denotes the capacity restriction at each collection point, i.e., the limit in the amount of refuse that can be assigned to the collection point. Finally, constraint sets (6) and (7) specify the conditions of decision variables.

The collection network design problem [P] formulated above has the characteristics of the p-median problem in that refuse demands occur at potential sites (See Kariv and Hakimi [15] and Mladenović *et al.* [28] for the p-median problem). Also, it is similar to the hub location problem since the refuses at demand points are allocated to collection points with fixed costs and capacity restrictions (See Campbell [3], Ebery *et al.* [7], ReVelle and Eiselt [30] and Klose and Drexel [17] for the hub location problem). However, the p-median problem does not consider the fixed costs to open the locations, the weights in the transportation costs, and the capacity restrictions. Also, in the hub location problem, the transportation costs consist of those from the origin to hubs, between hubs, and from hubs to the destination. In fact, the hubs, which connect the origin and the destination, have only the switching or sorting function. However, the collection network design problem considered in this paper is similar to the capacitated facility location problem. Nevertheless, this paper has a certain contribution in that we defined a basic collection network design problem for reverse logistics, which can be extendable to more generalized problems.

The problem [P] can be solved directly using a commercial integer programming software package. However, this requires an excessive amount of time (Results of computational tests will be reported later). In fact, the problem considered in this paper is NP-hard, which can be easily seen from the fact that the formulation contains the knapsack constraint. In addition, compared with the distribution network design problems, the problem considered in this paper is similar to the p-median problem and the hub location problem, but not exactly the same as each of them.

4. Solution Approach

This section presents the solution algorithms for the problem considered in this paper. Before explaining the detailed algorithms, we first suggest the method to determine the number of collection points to be opened.

4.1 Determining the number of collection points

To determine the number of collection points to be opened, we suggest a clustering method. The basic idea is that the potential sites are clustered according to the non-decreasing order of their adjacencies while considering the capacity restriction of each collection point. Here, the adjacency is defined as the transportation cost between two potential sites. A detailed procedure is given below.

• **Procedure 1:** Determining the number of collection points

Step 1: For all pairs of potential sites (i, j), compute the transportation cost h_{ij} as

$$\sum_{i \in N} \sum_{j \in N} w_i \cdot d_{ij} \cdot c_{ij} \quad \text{for all } i \text{ and } j,$$

where N denotes the set of potential sites. Then, sort the transportation costs in the non-decreasing order, and make a list of pairs of potential sites according to this order.

Step 2: Starting from the top of the list obtained in step 1, do the following steps (Repeat this step until all the potential sites are clustered).

(a) Given a transportation cost h_{ij} $(i \neq j)$, determine whether the two potential sites i and j can be merged without violating the capacity of collection point.

(b) If such points exist, merge them.

Step 3: Set the number of collection points to the number of clusters obtained in step 2.

4.2 Heuristic algorithms

As stated earlier, we suggest two types of heuristics, one with simultaneous and five with separate location and allocation. Detailed explanations of the six heuristics are given below.

4.2.1 Algorithm with simultaneous location and allocation

One simultaneous type algorithm, denoted by the *Consolidation Location and Allocation (CLA) algorithm,* is suggested. Given the clusters obtained from Procedure 1, the CLA algorithm selects the collection point for each cluster in such a way that it gives the minimum total transportation cost. Here, the total transportation cost for each candidate collection point in a cluster is calculated by letting the candidate be the collection point and allocating the other points within the cluster to the candidate collection point. A detailed procedure of the CLA algorithm is given below.

• **Procedure 2:** CLA algorithm

Step 1: Find a set of clusters using the steps given in procedure 1.

Step 2: For each cluster, determine a collection point using the following steps

(a) Set the initial transportation cost to a large number.

(b) Select a candidate collection point among those not considered yet and calculate the total transportation cost. If there is an improvement, set the current candidate to the collection point of the current cluster.

(c) If all candidate points are considered, stop. Otherwise, go to (b).

Step 3: For each cluster, allocate the other points to the selected collection point.

4.2.2 Algorithms with separate location and allocation

The separate type algorithms decompose the entire problem into two subproblems and then solve the two subproblems iteratively. Here, the two subproblems are: (a) selecting collection points from potential sites (location); and (b) allocating refuses at demand points to the selected collection points (allocation). For a given set of collection points, we can see that the allocation problem can be transformed into the generalized assignment problem (GAP). Recall that the GAP assigns multiple tasks to an agent subject to the availability of resource consumed by the agent in performing these tasks. In our case, each demand point can be represented by a task, and each collection point can be represented by an agent.

To solve the allocation problem, we use the modified version of the MTHG algorithm of Martello and Toth [26] since it gives fast and good solutions (Note that the allocation problem must be solved for each candidate of locating collection points). The modified MTHG algorithm, based on the desirability measure g_{ij}, consists of initialization and improvement steps. Here, $g_{ij} = w_i \cdot d_{ij}$ for all i and j. (Recall that w_i and d_{ij} denote the amount of refuse at node i and the distance between nodes i and j, respectively). A detailed procedure of the modified MTHG algorithm is given as follows.

• **Procedure 3:** Modified MTHG algorithm for allocation

Step 1: Initialization step

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

(b) Allocate demand point i to collection point j for which the desirability measure g_{ij} has the minimum value without violating the capacity of collection point j. If demand point j cannot be allocated to the current collection point, do this step for the next 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: Improvement step

(a) For each demand point, find the set A_i of collection points that give improvements without violating their capacity restrictions. Suppose that demand point i is allocated to collection point j^* in the initial solution, i.e., $x_{ij^*} = 1$. Then, the set A_i can be described as follows.

$$A_i = \{j \neq j^* \mid g_{ij} < g_{ij^*} \text{ and } w_i \leq R_j\},$$

where R_j denotes the remaining capacity of collection point j. 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, a reallocation implies the action that demand point i is assigned to the set A_i of collection points. Repeat this step until there is no improvement.

Based on the allocation method explained above, five separate type algorithms are suggested in this paper. Detailed explanations of the five heuristics are given below.

• *RL-G (Random Location and GAP) algorithm*

In this algorithm, collection points are selected randomly and then demand points are allocated to the selected collection points using the modified MTHG algorithm. This algorithm stops if no improvement has been made for a certain number of consecutive iterations. From a preliminary test, the number of consecutive iterations (for the termination condition) was set to 100.

• *DRL-G (Double Random Location and GAP) algorithm*

This algorithm is the same as the RL-G algorithm except that it generates the initial alternatives for collection points by two times of the number of collection points given by the method to determine the number of collection points (Procedure 1). The final collection points, i.e., a half of the initial ones, are determined by selecting the ones with larger refuse amounts, so that one may reduce the transportation cost. As in the RL-G algorithm, the DRL-G algorithm stops if no improvement has been made for a certain number of consecutive iterations. From a preliminary test, it was set to 1000.

- *PFL-G (Preprocessing Fixed Location and GAP) algorithm*

The PFL-G algorithm is similar to the CLA algorithm in that it selects the collection point for each cluster using the total transportation cost. However, for a given set of collection points, the allocation is done with the modified MTHG algorithm while ignoring the clusters obtained from the method to determine the number of collection points (Procedure 1).

- *SRL-G (Simple Rule Location and GAP) algorithm*

This algorithm is motivated from the observation that the collection points are generally located around the center of a region. Based on this observation, the SRL-G algorithm selects the collection points within the 80% bi-sector of the region. More specifically, the collection points within the 80% bi-sector are sorted in the non-increasing order of the refuse amount, and the final collection points are selected according to this order by the number given by the method to determine the number of collection points (Procedure 1). See Figure 3 for an example of the bi-sector for a 100×100 grid.

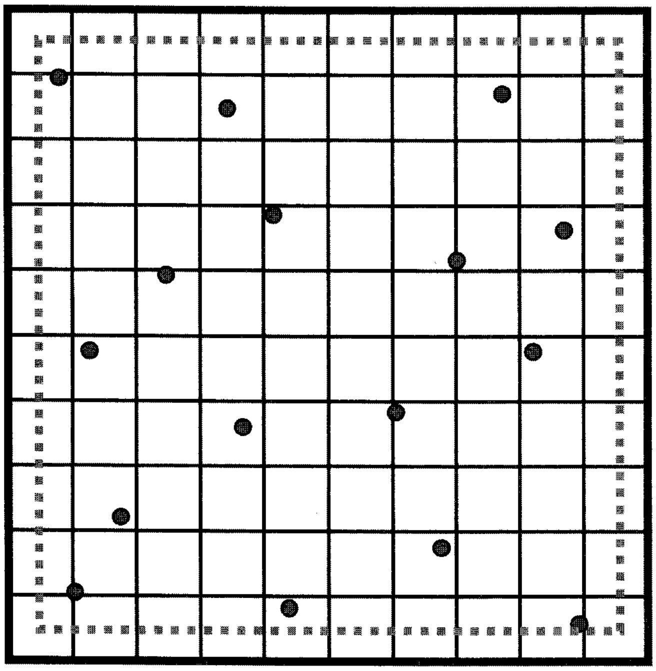

Figure 3. An example of the bi-sector

• *RML-G (Region Moving Location and GAP) algorithm*

This algorithm is a generalized version of the SRL-G algorithm in that the collection points are selected for each bi-sector from the border to the center. More specifically, this algorithm moves the bi-sector by the amount of 5%, and selects the best solution among those obtained for each bi-sector. See Figure 4 for an example of moving the bi-sector for a 100×100 grid.

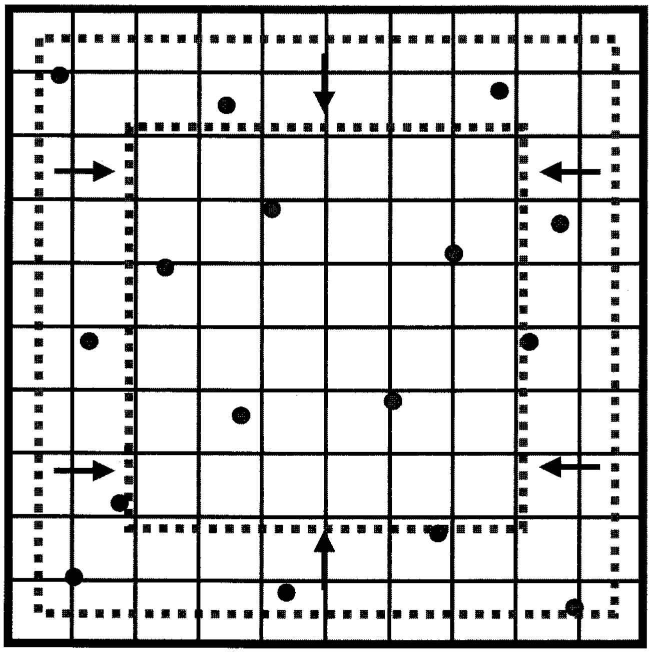

Figure 4. An example of moving the bi−sector

5. Computational Experiments

To show the performances of the heuristics suggested in this paper, computational experiments were done on a number of test problems, and the results are reported in this section. Since there are no benchmarking instances on the problem, we generated the test problems randomly.

The effectiveness of each heuristic is shown with three performance measures: (a) percentage gaps from the optimal solution values for small-size test problems; (b) relative performance ratios for the large-size test problems; and (c) CPU seconds. The optimal solutions for small-size test problems were obtained by solving the formulation [P] using CPLEX 9.0, and the relative performance ratio of heuristic *a* for a problem is defined as

$$100 \cdot (C_a - C_{best}) / C_{best}$$

where C_a is the objective function value obtained using heuristic *a* for the problem and C_{best} is the best objective function value among those obtained from the six heuristic algorithms. All the algorithms were coded in C, and tests were done on a workstation with a Xeon Processor operating at 3.2GHz speed.

For the test, 300 problems were generated randomly, i.e., 10 problems for each of 30 combinations of ten levels for the number of potential sites (10, 20, 30, 40, 50, 100, 200, 300, 400, and 500) and three levels for the capacity tightness of the collection point (loose, medium, and tight). The problem data were generated using the method of Daskin [5]. That is, the potential sites with their *x* and *y* coordinates were generated from *DU*(0, 100) on a square-shaped grid with a size of 100×100 unit length, where *DU*(*l*, *u*) denotes the discrete uniform distribution with range [*l*, *u*]. Also, the distance between two points was obtained by calculating its Euclidean distance and then rounding it to the nearest integer, and the unit transportation cost was set to 10. Refuse amount and fixed cost at each potential site were generated from *DU*(10, 40) and *DU*(120000, 150000), respectively. Finally, the capacity of each collection point was set to 40, 80, and 120 for tight, medium, and loose capacity cases, respectively.

Results for small-size test problems are summarized in Table 1 which shows the average gaps from the optimal solution values and the standard deviations. It can be seen from the table that RL-G, PFL-G, and DRL-G outperform the others for the problems with loose, medium, and tight capacities. Among the three better algorithms, the RL-G algorithm gave the best result, i.e., within 2% from the optimal solution values. Also, we can see that the separate type algorithms perform better than the simultaneous type CLA algorithm. This is because the CLA algorithm considers smaller number of alternatives for collection points than the separate type algorithms. In particular, all the heuristics gave the optimal solutions for the problem with 10 potential sites. Finally, the CPU seconds are not reported here since all the heuristics gave the solutions within 0.5 second. Note that the CPU seconds of CPLEX 9.0 were about 3 hours for the problems with 30 potential sites.

Table 1. Results for small-size test problems

Tightness of capacity	Number of potential sites	CLA[1]	RL-G[2]	DRL-G[3]	PFL-G[4]	SRL-G[5]	RML-G[6]
Loose	10	0.00 (0.00)*	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)
	20	2.93 (2.91)	1.15 (1.64)	3.02 (2.61)	0.75 (0.51)	3.02 (2.61)	2.97 (2.69)
	30	2.48 (1.54)	0.82 (0.52)	2.17 (1.15)	1.40 (0.62)	2.17 (1.15)	2.33 (1.23)
Medium	10	1.48 (1.94)	0.45 (1.38)	0.28 (0.36)	0.78 (1.37)	3.00 (2.45)	2.24 (1.99)
	20	3.17 (3.30)	1.20 (1.20)	0.95 (1.15)	1.98 (1.78)	7.09 (6.89)	5.13 (4.49)
tight	10	1.54 (0.75)	0.03 (0.12)	0.21 (0.31)	0.83 (0.92)	2.09 (1.19)	1.82 (0.59)

Note: [1] Consolidation Location and Allocation Algorithm
[2] Random Location and GAP Algorithm
[3] Double Random Location and GAP Algorithm
[4] Preprocessing Fixed Location and GAP Algorithm
[5] Simple Rule Location and GAP Algorithm
[6] Region Moving Location and GAP Algorithm
* average gap (standard deviation in parenthesis) from the optimal solution values out of 10 problems
(This value was obtained only for problems for which optimal solutions are obtained)

Results for large-size test problems (with 40, 50, 100, 200, 300, 400 and 500 potential sites) are summarized in Tables 2 and 3, which show the average relative performance ratios and the average CPU seconds of the heuristics. As in the results for small-size test problems, RL-G, PFL-G, and DRL-G outperform the others in overall average. Results on paired *t*-tests are also given in Table 4 from which we can see that performances of RL-G, PFL-G, and DRL-G are statistically different from those of the others. However, as can be seen in Table 3, PFL-G required much shorter computation time than RL-G and DRL-G. In fact, PFL-G solved all the test problems within 1.5 seconds. In summary, we can recommend the RL-G algorithm for small-size problems (up to 20 potential sites) and the PFL-G algorithm up to large-size problems in terms of solution quality and computation time.

Table 2. Results for large-size test problems

(a) Case of loose capacity

Number of potential sites	CLA[1]	RL-G[2]	DRL-G[3]	PFL-G[4]	SRL-G[5]	RML-G[6]
40	0.10 (0.10)*	0.29 (0.11)	0.41 (0.38)	0.06 (0.14)	0.41 (0.38)	0.25 (0.18)
50	0.04 (0.05)	0.32 (0.12)	0.25 (0.27)	0.18 (0.34)	0.25 (0.27)	0.14 (0.12)
100	0.06 (0.05)	0.46 (0.06)	0.10 (0.11)	0.00 (0.01)	0.10 (0.11)	0.08 (0.07)
200	0.19 (0.34)	0.38 (0.15)	0.24 (0.32)	0.11 (0.34)	0.24 (0.32)	0.21 (0.33)
300	0.06 (0.04)	0.38 (0.07)	0.08 (0.06)	0.00 (0.00)	0.08 (0.06)	0.07 (0.05)
400	0.07 (0.02)	0.35 (0.04)	0.07 (0.03)	0.00 (0.00)	0.07 (0.03)	0.07 (0.02)
500	0.05 (0.03)	0.34 (0.04)	0.08 (0.03)	0.00 (0.00	0.08 (0.03)	0.07 (0.02)
Average	0.08 (0.09)	0.36 (0.09)	0.18 (0.17)	0.05 (0.12)	0.18 (0.17)	0.13 (0.11)

Note: See the footnotes of Table 2.

*Average relative performance ratio (standard deviation in parenthesis) out of 10 problems.

(b) Case of medium capacity

Number of potential sites	CLA[1]	RL-G[2]	DRL-G[3]	PFL-G[4]	SRL-G[5]	RML-G[6]
40	2.58 (3.03)	0.73 (0.30)	0.00 (0.00)	2.85 (2.53)	3.48 (1.55)	3.03 (1.64)
50	2.88 (4.01)	2.05 (3.88)	0.02 (0.06)	2.34 (3.90)	3.00 (4.08)	2.94 (3.83)
100	1.26 (2.14)	0.99 (0.30)	0.19 (0.23)	0.92 (1.04)	1.65 (0.85)	1.46 (1.19)
200	0.98 (1.55)	0.87 (0.25)	0.15 (0.19)	0.82 (1.55)	1.35 (0.75)	1.16 (0.89)
300	0.86 (1.39)	0.72 (0.14)	0.18 (0.15)	0.84 (1.36)	0.82 (0.60)	0.84 (0.77)
400	0.87 (1.22)	0.77 (0.20)	0.21 (0.16)	0.74 (1.14)	1.04 (0.53)	0.95 (0.61)
500	0.82 (1.17)	0.74 (0.18)	0.25 (0.21)	0.73 (1.15)	0.72 (0.39)	0.77 (0.51)
Average	1.46 (2.07)	0.98 (0.75)	0.14 (0.14)	1.32 (1.81)	1.72 (1.25)	1.59 (1.35)

(c) Case of tight capacity

Number of potential sites	CLA[1]	RL-G[2]	DRL-G[3]	PFL-G[4]	SRL-G[5]	RML-G[6]
40	5.0 (11.14)	0.77 (0.35)	0.21 (0.25)	4.71 (10.86)	6.50 (2.98)	5.75 (6.01)
50	1.22 (1.05)	0.81 (0.51)	0.19 (0.33)	0.67 (0.67)	3.97 (2.34)	2.60 (1.38)
100	0.62 (0.64)	1.44 (0.90)	0.44 (0.51)	0.40 (0.57)	4.22 (1.58)	2.42 (0.78)
200	0.42 (0.33)	1.47 (0.31)	0.65 (0.38)	0.03 (0.08)	3.97 (1.42)	2.19 (0.77)
300	0.14 (0.17)	1.52 (0.31)	0.86 (0.23)	0.07 (0.09)	3.28 (1.24)	1.71 (0.63)
400	0.22 (0.24)	1.35 (0.18)	0.80 (0.23)	0.09 (0.13)	3.19 (1.38)	1.70 (0.73)
500	0.25 (0.21)	1.45 (0.19)	1.04 (0.17)	0.02 (0.05)	2.42 (1.47)	1.34 (0.68)
Average	1.12 (1.97)	1.26(0.39)	0.60 (0.30)	0.86 (1.78)	3.94 (1.77)	2.53 (1.57)

Table 3. CPU seconds of the heuristics

(a) Case of loose capacity

Number of potential sites	CLA[1]	RL-G[2]	DRL-G[3]	PFL-G[4]	SRL-G[5]	RML-G[6]
40	0.01[*]	0.7	0.01	0.00	0.01	0.02
50	0.01	1.3	0.00	0.00	0.02	0.02
100	0.04	8.3	0.02	0.01	0.02	0.1
200	0.2	58.1	0.1	0.04	0.1	0.3
300	0.8	223.5	0.3	0.13	0.3	1.3
400	1.8	591.8	0.6	0.28	0.6	3.0
500	3.7	823.3	1.2	0.48	1.1	6.1

Note: See the footnotes of Table 2.

[*] Average CPU second out of 10 problems.

(b) Case of medium capacity

Number of potential sites	CLA[1]	RL-G[2]	DRL-G[3]	PFL-G[4]	SRL-G[5]	RML-G[6]
40	0.00	0.7	0.7	0.00	0.01	0.01
50	0.01	1.2	1.2	0.01	0.01	0.02
100	0.03	8.2	9.4	0.01	0.02	0.1
200	0.2	68.1	68.6	0.1	0.1	0.4
300	0.7	265.9	187.7	0.2	0.3	1.3
400	1.9	598.6	534.3	0.5	0.6	3.1
500	3.8	833.0	833.9	0.9	1.1	6.3

(c) Case of tight capacity

Number of potential sites	CLA[1]	RL-G[2]	DRL-G[3]	PFL-G[4]	SRL-G[5]	RML-G[6]
40	0.01	0.7	0.6	0.0	0.00	0.02
50	0.01	1.2	1.2	0.0	0.01	0.02
100	0.03	8.1	9.3	0.02	0.02	0.1
200	0.2	67.8	68.2	0.1	0.1	0.3
300	0.7	266.2	185.6	0.2	0.3	1.2
400	1.6	544.0	532.6	0.5	0.6	2.8
500	3.3	837.5	828.7	1.1	1.1	5.6

Table 4. Results for paired t-tests

(a) Case of small-size problems

Algorithms	RL-G (0.61)	PFL-G (0.96)	DRL-G (1.11)	CLA (1.94)	RML-G (2.42)
RL-G (0.61)					
PFL-G (0.96)	=				
DRL-G (1.11)	=	=			
CLA (1.94)	**	**	**		
RML-G (2.42)	**	**	**	**	
SRL-G (2.90)	**	**	**	=	=

Note: ** Indicates statistically different at significance level of 99%.
= Indicates not statistically different at significance level of 99%.

(b) Case of large-size problems

Algorithms	DRL-G (0.31)	PFL-G (0.74)	RL-G (0.87)	CLA (0.89)	RML-G (1.42)
DRL-G (0.31)					
PFL-G (0.74)	=				
RL-G (0.87)	**	=			
CLA (0.89)	**	=	=		
RML-G (1.42)	**	**	**	**	
SRL-G (1.94)	**	**	**	**	**

Note: See the foot notes of (a).

6. Conclusion

This paper considered a basic network design problem for the collection activity in reverse logistics. The problem is to determine the location of collection points and the allocation of refuses at demand points to collection points while satisfying the capacity restriction at each collection point. The objective is to minimize the sum of the fixed costs to open collection points and the transportation costs between demand points and collection points. An integer programming model was suggested to represent the problem mathematically, and two types of heuristic algorithms, one with simultaneous and the others with separate location and allocation, were suggested. Computational experiments were done on test problems up to 500 potential sites, and the results were reported. From the test results, we recommend the RL-G algorithm for small-size problems (up to 20 potential sites) and the PFL-G algorithm up to large-size problems in terms of solution quality and computation time. In particular, both algorithms gave near optimal solutions for small-size test problems (about 2% average gaps from the optimal solution values).

This paper can be extended in several directions. First, it is needed to develop more effective methods to locate collection points. In this case, the search heuristics, such as simulated annealing, genetic algorithm, and tabu search, can be applied this problem. Second, it is required to extend the model by incorporating other features of refuse collection systems, since we consider the most basic form of the problem. Finally, case studies are needed on designing real refuse collection systems, from which one can specify the differences from the existing facility location models.

References

[1] 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.

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

[3] Campbell, J. F. A., "survey of network hub location," *Studies in Locational Analysis* 6 (1994), 31-47.

[4] 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.

[5] Daskin, M. S., *GENRAND2: A random network generator,* Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL, USA 1993.

[6] Dowlatshahi, S., "Developing a theory of reverse logistics," *Interfaces* 30 (2000), 143-155.

[7] Ebery, J., M. Krishnamoorthy, A. Ernst, and B. Boland, "The capacitated multiple allocation hub location problem: Formulations and algorithms," *European Journal of Operational Research* 120 (2000), 614-631.

[8] Ferguson, N. and J. Browne, "Issues in end-of-life product recovery and reverse logistics," *Production Planning and Control* 12 (2001), 534-547.

[9] 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.

[10] 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.

[11] 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.

[12] Guide, Jr. V. D. R., V. Jayaraman, R. Srivastava, and W. C. Benton, "Supply-chain management for recoverable manufacturing systems," *Interfaces* 30 (2000), 125-142.

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

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

[15] Kariv, O. and S. L. Hakimi, "An algorithmic approach to network location problem: part 2 the p-medians," *SIAM Journal on Applied Mathematics* 37 (1969), 539-560.

[16] Kim, J. D., H. S. Choi, and D. H. Lee, "A case study on the stochastic vehicle routing in a refuse collection system," *Journal of the Korean Society of Supply Chain Management* 7 (2007), 57-65.

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

[18] 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.

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

[20] 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.

[21] Krikke, H. R., A. van Harten, and P. C. Schuur, "Business case océ: reverse logistics network re-design for copiers," *OR Spectrum* 21 (1999), 381-409.

[22] 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.

[23] 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.

[24] 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.

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

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

[27] 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 (2006), 56-69.

[28] Mladenović, N., J. Brimberg, P. Hansen, and J. A. Moreno-Pérez, "The p-median problem: a survey of metaheuristic approaches," *European Journal of Operational Research* 179 (2007), 927-939.

[29] Realff, M. J., J. C. Ammons, and D. Newton, "Carpet recycling: determining the reverse logistics system design," *The Journal of Polymer-Plastics Technology and Engineering* 38 (1999), 547-567.

[30] Revelle, C. S. and H. A. Eiselt, "Location analysis: a synthesis and survey," *European Journal of Operational Research* 165 (2005), 1-19.

[31] 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.

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