Int J Adv Manuf Technol (2010) 47:269–281
DOI 10.1007/s00170-009-2194-5

ORIGINAL ARTICLE

Reverse logistics network design using simulated annealing

Mir Saman Pishvaee · Kamran Kianfar · Behrooz Karimi

Received: 3 August 2008 / Accepted: 29 June 2009 / Published online: 17 July 2009

Abstract Reverse logistics is becoming more important in overall industry area because of the environmental and business factors. Planning and implementing a suitable reverse logistics network could bring more profit, customer satisfaction, and a nice social picture for companies. But, most of logistics networks are not equipped to handle the return products in reverse channels. This paper proposes a mixed integer linear programming model to minimize the transportation and fixed opening costs in a multistage reverse logistics network. Since such network design problems belong to the class of NP-hard problems, we apply a simulated annealing (SA) algorithm with special neighborhood search mechanisms to find the near optimal solution. We also compare the associated numerical results through exact solutions in a set of problems to present the high-quality performance of the applied SA algorithm.

Keywords Reverse logistics · Logistics network design · Supply chain network · Simulated annealing · Priority-based encoding

1 Introduction

Logistics network design is one of the most important strategic decisions in supply chain management. In general, network design decisions include determining the numbers, locations, and capacities of facilities and the quantity of the flow between them.

Since opening and closing a facility is very expensive and time-consuming, making changes in facility location decisions is impossible in a short time. Investment on strategic level decisions such as logistics network design has a higher return in comparison with the tactical and operational levels. Since strategic decisions are determined before other tactical and operational decisions, the configuration of logistics network will become a constraint for tactical- and operational-level decisions [19].

In the last decade, many companies such as Dell, General Motors, Kodak, and Xerox focused on remanufacturing and recovery activities and have achieved significant successes in this area [23].

Meade et al. [18] divide driving forces leading to increased interest and investment in reverse logistics into two groups: environmental and business factors. The first one includes the environmental impact of the used products, environmental legislation, and growing environmental consciousness of the customers. Business factors are related to economic benefits of using return products and liberal return policies for gaining customer satisfaction.

In fact, companies could gain the economic benefits of using remanufactured products in production process directly, and they could also gain more profit indirectly through liberal return policies and repair services that ends in customer satisfaction.

In some cases, recovery activities on return products create an opportunity for consumers to buy and use a product that meets the original product standards at a lower price than a new one [9].

Reverse logistics network design includes determining numbers, locations, and capacities of collection, recovery, and disposal centers, buffer inventories in each site, and the quantity of flow between them. Reverse logistics networks have special characteristics differentiating them from forward logistics networks.

One of these characteristics is the important role of collection/inspection centers. Since return products have different qualities, they have different potentials for recovery activities, too. After testing in collection/inspec-

M. S. Pishvaee · K. Kianfar · B. Karimi (✉)
Department of Industrial Engineering,
Amirkabir University of Technology,
Tehran, Iran
e-mail: B.Karimi@aut.ac.ir

tion centers, return products are divided into recoverable and scrapped products to prevent excessive transportation and to ship the return products directly to proper facilities [6]. Thereupon, the number and location of these facilities have a fundamental impact on transportation costs.

In many cases, logistics networks are designed for forward logistics activities without considering the reverse flow of return products, and most of them are not equipped to handle the return products in reverse channels [9]. This may be derived from the lack of knowledge about benefits of reverse logistics.

In this paper, authors consider a multistage reverse logistics network with collection/inspection, recovery, and disposal facilities. A mixed integer linear programming (MILP) model is developed to minimize the total costs, and an efficient simulated annealing (SA) algorithm is also applied to solve the model. An effective mathematical model that could support recovery and disposal processes, adapting priority-based encoding method to the applied SA algorithm, and finally, a dynamic neighborhood search strategy that enhances the performance of the applied algorithm are the main contributions of this paper. The remainder of this paper is organized as follows. In Section 2, we provide a review on related literature. The problem definition and mathematical model of reverse logistics network is presented in Section 3, and the applied simulated annealing algorithm is described in Section 4. Computational results are reported in Section 5, and finally, a summary of the paper and some possible future works are given in Section 6.

2 Literature review

Most of the literature about logistics network design considers various facility location models based on the MILP. These models include a range of models from simple uncapacitated facility location models to more complex models such as capacitated multistage or multi-commodity models. Also, powerful algorithms have been developed based on the combinatorial optimization for solving these models. Following them, we will review the literature of some models proposed for reverse logistics network design.

Jayaraman et al. [9] developed a MILP model for reverse logistics network design under a pull system based on customers' demands for recovered products. The objective of the proposed model is to minimize the total costs.

Krikke et al. [12] designed a MILP model for a two-stage reverse logistics network of a copier manufacturer. In this model, the processing costs of returned products and inventory costs are considered in the objective function.

Based on the reverse logistics literature, the uncertainty of the quantity and quality of the returned products is an important factor in the design of the reverse logistics network. Accordingly, Listes and Dekker [16] propose a stochastic mixed integer programming model in a sand recycling network to maximize the total profit. They developed their model for different situations regarding several scenarios.

Aras et al. [1] develop a nonlinear model for determining the locations of collection centers in a simple reverse logistics network. The important point regarding their article is the capability of the presented model in determining the optimal buying price of used products with the objective of maximizing the total profit. They developed a heuristic approach based on tabu search to solve the model.

Üster et al. [23] designed a semi-integrated network in which the direct logistics network exists and only collection and recovery centers must be located. The model optimizes the direct and reverse flows simultaneously. An exact method is developed based on the Benders decomposition technique.

Lu and Bostel [17] considered a two-level location problem with three types of facilities to be located in a special reverse logistics system called a remanufacturing network. They propose a 0–1 mixed integer-programming model that considers the forward and reverse flows and their interactions at the same time. They also developed an algorithm based on the Lagrangian heuristics to solve the proposed model.

Wojanowski et al. [24] studied the interplay between the industrial firms and government concerning the collection of used products from households. They presented a continuous modeling framework for designing a dropoff facility network and determining the sales price to maximize the firm's profit under a specific deposit–refund.

For designing a closed-loop logistics network for third-party logistics providers, Du and Evans [5] proposed an advanced bi-objective MILP model. The objectives of the model include the minimization of the tardiness and the total costs. In order to solve the model, a hybrid-scatter-search method is developed.

Many research directions still require intensive research in the area of reverse logistics network design. The literature of uncertainty in reverse logistics is still scarce. Moreover, since network design problems belong to the class of NP-hard problems, developing efficient solution methods is still a critical need in this area.

3 Problem definition and mathematical modeling

The reverse logistics network discussed in this paper is a multistage logistics network including customers, collection/inspection, recovery, and disposal centers with limited capacities. As shown in Fig. 1, returned products are collected from customer zones into collection/inspection centers and after quality inspection; they are divided into recoverable products and scrapped products. The recoverable products are carried to the recovery centers and scrapped products are sent to the disposal centers. Disposals may include any form of recovery that is outsourced to a third party, e.g., recycling. As mentioned before, through this strategy, returned products' excessive transportation is prevented and could be shipped directly to the proper facilities. The studied reverse logistics network, illustrated in Fig. 1, has a convergent structure from customers to recovery centers with a push system.

To specify the study scope, three assumptions and simplifications in the proposed model formulation are postulated as follows:

– All of the returned products from customers must be collected;
– Customer locations are fixed and predefined;
– Numbers, locations, and capacities of recovery and disposal centers are known in advance.

The objective of this reverse logistics network design is to choose the location and to determine the number of collection/inspection centers which represents the degree of centralization of the network and to determine the quantity of flow between the network facilities.

The following notations are used in the formulation of the proposed model.

Sets

I Set of the candidate points for collection/inspection centers, $\forall i \in I$.
J Fixed set of points for recovery centers, $\forall j \in J$.
K Fixed set of points for disposal centers, $\forall k \in K$.
L Fixed set of points for customer centers, $\forall l \in L$.

Parameters

d Average fraction of disposed products (percent).
r_l Amount of returned products from customer center l.
f_i Fixed cost to set up collection/inspection center i.
cf_{li} Transportation cost for a unit of returned product from customer center l to collection/inspection center i.
cs_{ij} Transportation cost for a unit of recoverable product from collection/inspection center i to recovery center j.

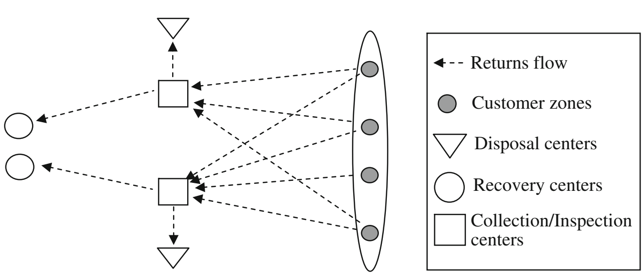


Fig. 1 Structure of a reverse logistics network

ct_{ik} Transportation cost for a unit of scrapped product from collection/inspection center i to disposal center k.
caf_i Capacity of the collection/inspection center i.
cas_j Capacity of the recovery center j.
cat_k Capacity of the disposal center k.

Variables

X_{li} Amount of returned products transferred from customer center l to collection/inspection center i.
Z_{ij} Amount of recoverable products transferred from collection/inspection center i to recovery center j.
W_{ik} Amount of scrapped products transferred from collection/inspection center i to disposal center k.

$$Y_i = \begin{cases} 1 & \text{if a collection/inspection center is open at location i,} \\ 0 & \text{otherwise} \end{cases}$$

In terms of the above notations, the reverse logistics network design problem can be formulated as follows:

$$\mathrm{Min}\left[\sum_{i\in I} f_i Y_i + \sum_{l\in L}\sum_{i\in I}\mathrm{cf}_{li}X_{li} + \sum_{i\in I}\sum_{j\in J}\mathrm{cs}_{ij}Z_{ij} + \sum_{i\in I}\sum_{k\in K}\mathrm{ct}_{ik}W_{ik}\right] \tag{1}$$

$$\sum_{i\in I} X_{li} = r_l \qquad \forall l \in L \tag{2}$$

$$\sum_{j\in J} Z_{ij} = (1-d)\sum_{l\in L} X_{li} \qquad \forall i \in I \tag{3}$$

$$\sum_{k\in K} W_{ik} = d\sum_{l\in L} X_{li} \qquad \forall i \in I \tag{4}$$

$$\sum_{l\in L} X_{li} \le Y_i \mathrm{caf}_i \qquad \forall i \in I \tag{5}$$

$$\sum_{i\in I} Z_{ij} \le \mathrm{cas}_j \qquad \forall j \in J \tag{6}$$

$$\sum_{i\in I} W_{ik} \le \mathrm{cat}_k \qquad \forall k \in K \tag{7}$$

$$Y_i \in \{0,1\} \qquad \forall i \in I \tag{8}$$

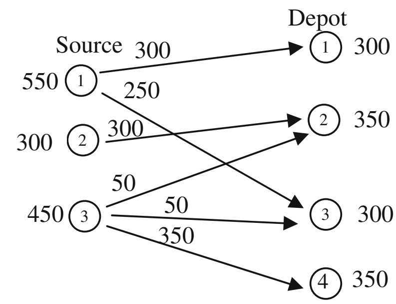
$$X_{li}, Z_{ij}, W_{ik} \ge 0 \qquad \forall i \in I, j \in J, k \in K, l \in L \tag{9}$$

In this model, the objective is to minimize the total costs including the fixed opening costs of collection/inspection centers and transportation costs between the facilities.

Fig. 2 Sample of transportation tree and related encoding (adopted from [8])

	Sources			Depots			
node	**1**	**2**	**3**	**1**	**2**	**3**	**4**
$v(k+j)$ **priority**	**2**	**5**	**3**	**7**	**4**	**1**	**6**

	1	2	3	4
1	11	19	17	18
2	16	14	18	15
3	15	16	19	13



Constraint (2) ensures that all of the returned products are collected from each customer. Constraints (3) and (4) assure the flow balance at collection/inspection centers. Constraint (5) prohibits the units of returned products from being transferred to collection/inspection centers unless the centers are built up and also ensures that they do not exceed the collection/inspection centers' capacities. Constraints (6) and (7) make sure that the returned products shipped to the recovery and disposal facilities do not exceed their capacities. Finally, Constraints (8) and (9) enforce the binary and non-negativity restrictions on corresponding decision variables.

The resulting model is a MILP with $(LI+IJ+IK)$ continuous variables and (I) binary variables. The number of constraints is $(L+J+K+3I)$, excluding constraints (8) and (9).

The NP-hardness of the logistics network design problem is proved in many researches (e.g., [11]). Precisely, the model considered in this paper consists of two problems, capacitated facility location problem and flow optimization; therefore, the model is reducible to capacitated facility location problem. Since the capacitated facility location problem is NP-complete [4], the reverse logistics network design problem discussed in this paper is NP-Hard.

Solving this problem in a large size by exact algorithms is very time-consuming; therefore, many heuristics and meta-heuristics have been developed to get near optimal solutions for these kinds of problems. Here, a SA algorithm based on priority-based encoding is applied, which is described in next section.

4 The simulated annealing algorithm

SA [3,13] is among the most popular iterative methods that have been applied widely to solve many combinatorial optimization problems [14,15]. This method is a random local search technique based on the principles of physics. The search starts from an initial feasible solution. Each solution has a specific cost value. A small change in one or a combination of some variables can generate a neighboring solution with a different cost value. In simulated annealing, the neighboring solution is generated randomly. If the cost value of the candidate solution is lower than that of the current solution, a move to the candidate solution is made. However, if the candidate does not improve the current solution, there is still a chance of transition according to a special probability function based on the "Boltzmann" distribution.

Detailed discussions of simulated annealing can be found in Aarts and Lenstra [2] or in Pirlot [22].

To the best of our knowledge, except Jayaraman and Ross [10], there is no use of SA in the logistics network

Table 1 Trace table of encoding procedure

Iteration	v(k+j)	a	b	K	J	g_{kj}
0	(2 5 3 | 7 4 1 6)	(550, 300, 450)	(300, 350, 300, 350)	1	1	300
1	(2 5 3 | 0 4 1 6)	(250, 300, 450)	(0, 350, 300, 350)	4	3	350
2	(2 5 3 | 0 4 1 0)	(250, 300, 100)	(0, 350, 300, 0)	2	2	300
3	(2 0 3 | 0 4 1 0)	(250, 0, 100)	(0, 50, 300, 0)	2	3	50
4	(2 0 3 | 0 0 1 0)	(250, 0, 50)	(0, 0, 300, 0)	3	3	50
5	(2 0 0 | 0 0 1 0)	(250, 0, 0)	(0, 0, 250, 0)	3	1	250
6	(0 0 0 | 0 0 0 0)	(0, 0, 0)	(0, 0, 0, 0)			

Adopted from [8]

	First segment						Second segment				Third segment			
node	1	2	3	4	1	2	1	2	1	2	1	2	1	2
v(l) **priority**	2	4	1	5	6	3	3	1	2	4	2	1	4	3

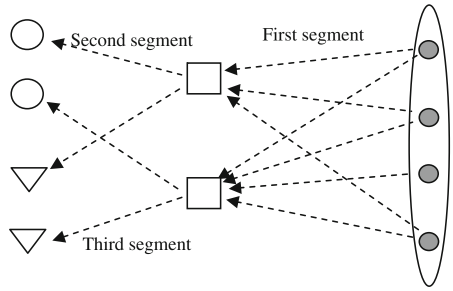


Fig. 3 An illustration of the reverse logistics problem and its encoded solution

design problem. However, Jayaraman and Ross [10] studied direct logistics network design, and there is no research that uses SA for solving reverse logistics network design problem. Authors' motivation to use SA instead of other methods is its simplicity and its high speed in achieving the near optimal solutions.

Laha and Chakraborty [14] presented three probabilistic hybrid heuristics for solving flow shop scheduling problem. Their proposed heuristics combine elements from both constructive heuristic search and simulated annealing as a stochastic improvement technique. One year later, Laha and Chakraborty [15] considered the same problem and developed another hybrid heuristic based on simulated annealing. In comparison with the above two papers, our algorithm combines simulated annealing with priority-based encoding method and special neighborhood search mechanisms for the problem of reverse logistic network design. In this section, different parts of the applied algorithm are introduced, and the relation between them is explained.

4.1 Encoding and decoding of solutions using priority-based method

Different methods have been developed to encode trees; one of them is matrix-encoding method, which is developed by Michalewicz et al. [20]. In this method, the solution is presented through a $|K|.|J|$ matrix in which $|K|$ implies the number of sources, and $|J|$ shows the number of depots.

Step1: $Y_i = 0, X_{ij} = 0, Z_{li} = 0, W_{ik} = 0 \quad \forall i \in I, l \in L, j \in J, k \in K$

Step2: calculate $X_{li} \quad \forall i \in I, l \in L$ using Gen and Cheng's decoding algorithm

Step3: For $i = 1$ to $i = I$

if $\sum_{l \in L} X_{li} \geq 0$ then $Y_i = 1$

Step4: calculate $Z_{ij} \quad \forall i \in I, j \in J$ using Gen and Cheng's decoding algorithm

Step5: calculate $W_{ik} \quad \forall i \in I, k \in K$ using Gen and Cheng's decoding algorithm

Fig. 4 Multistage decoding algorithm for reverse logistics network design problem encoded solution

	First segment						Second segment				Third segment			
node	1	2	3	4	1	2	1	2	1	2	1	2	1	2
v(l) **priority**	2	4	1	5	6	3	3	1	2	4	2	1	4	3
	1						0				1			
	1	2	3	4	1	2	1	2	1	2	1	2	1	2
	2	5	1	4	6	3	3	1	2	4	4	1	2	3

Fig. 5 An illustration of 2-opt neighborhood search

Despite its simple representation of solution, the implementation of this method requires developing special operators and using huge memory space.

Gen and Cheng [7] developed spanning tree method that uses a special algorithm based on the PrÜfer number. In this method, solutions are represented through a $|K|+|J|-2$ length array. This method may result in infeasible solutions; to prevent this, repair mechanisms have been developed. However, Gen and Cheng [7] presented a priority-based encoding algorithm as an alternative that does not need any excessive repair mechanism. They used this method for chromosome representation in a genetic algorithm.

In this method, solutions are encoded as arrays of size $|K|+|J|$, and the position of each cell represents the sources and depots and the value in cells represent the priorities (Fig. 2). To decode a solution, after priority assignment, the algorithm starts from highest priority. In each iteration, the node (depot or source) with the highest priority is selected and then connected to a depot or source with the minimum transportation cost. After that, the minimum of demand and capacity of the selected depot and source is determined as the amount of shipment between the selected nodes. This process is repeated until all demands of depots are satisfied. For more information about the decoding algorithm proposed by Gen et al. [8], we refer the readers to Appendix 1.

For example, in the network shown in Fig. 2, in the first step, depot 1 with the maximum priority (equal to 7) is selected. Then, according to the cost matrix, source 1 is

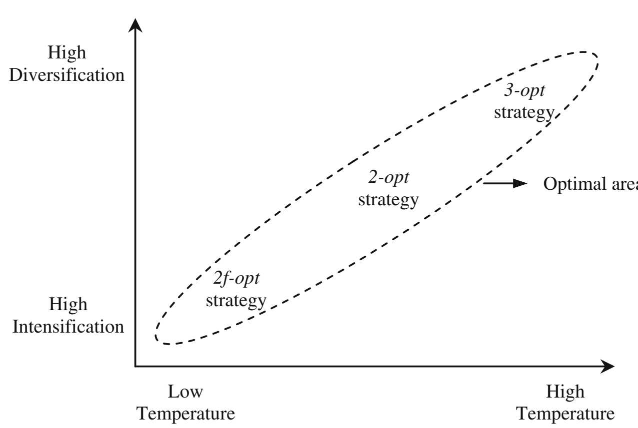


Fig. 6 The logic of using different neighborhood search methods

According to the above explanations, the SA algorithm is outlined as follows:

Step 1: *Initial temperature=100, Frozen state=0, Cooling rate=cr, k=7*

Step 2: Generate $P= (I+J+K+L)$ random solutions

Step 3: Decode random solutions using multi-stage decoding algorithm

$f^*(X) = \min (f(X_p)),\ X^*=X$; Select the best solution as the initial solution

Step 4: *Temperature=Initial temperature*

While *Temperature>Frozen state* **do**

From *i=1* to *1+ P/(2*Temperature)*

Ti=Temperature

if *Temperature>70*

Create a new solution using *3-opt* neighborhood search algorithm (X_{new})

Decode X_{new} using multi-stage decoding algorithm

if $f^*(X) - f(X_{new}) \geq 0$ **then** $f^*(X) = f(X_{new}),\ X^*=X_{new}$

else if $\exp(f^*(X) - f(X_{new})/kTi) > Random\ [0\ 1]$ **then** $f^*(X) = f(X_{new}),\ X^*=X_{new}$

else if $5 < Temperature \leq 70$

Create a new solution using *2-opt* neighborhood search algorithm (X_{new})

Decode X_{new} using multi-stage decoding algorithm

if $f^*(X) - f(X_{new}) \geq 0$ **then** $f^*(X) = f(X_{new}),\ X^*=X_{new}$

else if $\exp(f^*(X) - f(X_{new})/kTi) > Random\ [0\ 1]$ **then** $f^*(X) = f(X_{new}),\ X^*=X_{new}$

else

Create a new solution using *2f-opt* neighborhood search algorithm (X_{new})

Decode X_{new} using multi-stage decoding algorithm

if $f^*(X) - f(X_{new}) \geq 0$ **then** $f^*(X) = f(X_{new}),\ X^*=X_{new}$

else if $\exp(f^*(X) - f(X_{new})/kTi) > Random\ [0\ 1]$ **then** $f^*(X) = f(X_{new}),\ X^*=X_{new}$

Temperature=Temprature-cr

Step 5: Return the final solution

Fig. 7 The pseudo code of the SA algorithm

selected because it has the lowest related cost to depot 1. After that, the amount of transportations between them is calculated as min{550,300}, and then the demand and the capacity related to the selected nodes are updated. The trace table of the above sample is given in Table 1.

To apply the above method to the discussed problem, we should divide the solution vector to three segments, in which first segment is devoted to the customers and collection/inspection centers, second one is devoted to the collection/inspection and recovery centers, and finally, the third segment is for collection/inspection and disposal centers (Fig. 3).

Decoding of the second and the third segments of the solution vector are impossible until the first segment is decoded because the number and location of collection/ inspection centers and the amount of returned products shipped to each one should be known for decoding the next stages. The decoding algorithm for the multistage reverse logistics network design solution is shown in Fig. 4.

Table 2 Size of some test problems in literature

References	No. of (initial\fixed) collection/inspection centers	No. of (initial\fixed) recovery centers	No. of (initial\fixed) disposal centers	No. of customer zones
Min et al. [21]	10	1	–	30
Jayaraman et al. [9]	5	10	–	10
Üster et al. [23]	35	10	–	120
Du and Evans [5]	80	20	–	–
Aras et al. [1]	4	–	–	200

Another important issue is that there is not a one-to-one relationship between each encoding and corresponding decoded solution in general; rather, different encodings may result in the same solution when decoded.

4.2 Definition of the neighborhood searches

In the SA algorithm, three neighborhood search methods are used, with process consisting of two stages. In the first stage, each segment of the solution vector is randomly selected with equal chance through a binary mask. Then, at the second stage in each selected segment, the value (priority) in some cells will be exchanged with each other (Fig. 5). But, we use three kinds of neighborhood searches by making changes in the stages of the method. In the first one (*3-opt*), three cells in each segment are exchanged with each other; in the second method (*2-opt*), two cells in each segment are exchanged with each other. Finally, in the third one (*2f-opt*), like the *2-opt* method, two cells are exchanged in the second stage, but in the first stage, only selecting one or two segments is possible and selecting three segments is prohibited. However, in the *3-opt* and *2-opt* methods, selecting one, two, or three segments in the first stage are possible.

In the applied SA algorithm at high temperatures, we use *3-opt* neighborhood searches to achieve more diversification and to take bigger steps towards better solutions. At low temperatures, we use *2f-opt* neighborhood search to achieve more intensification and to search more carefully around the obtained solution. Finally, in the moderate temperatures, *2-opt* neighborhood search is used. The logic of using different kinds of neighborhood searches in the SA algorithm, described above, is illustrated in Fig. 6.

4.3 SA algorithm

According to the above explanations, the SA algorithm is outlined as follows:

With the aid of test problems, the parameters of the SA algorithm are tuned by changing their values and comparing the results. As is indicated in the above algorithm, the number of inner loop iterations is increased dynamically with the rate of ($P/2$*temperature) to gain more intensification and better solutions in lower temperatures.

In addition, 3-opt neighborhood search is used in temperatures between 70 and 100 in order to increase the speed of approaching to desirable solutions. 2-opt neighborhood search is applied in temperatures between 5 and 70, and finally 2*f*-opt neighborhood search is used in temperatures less than 5. The other parameters of the SA algorithm are shown in Fig. 7.

5 Computational results

Various test problems, with different sizes, are presented in the field of reverse logistics network design. The sizes of

Table 3 Test problems' sizes

Problem no.	No. of initial collection\inspection centers	No. of recovery centers	No. of disposal centers	No. of customer zones
Small sizes				
1	5	2	2	5
2	10	2	2	5
3	10	2	4	8
4	14	3	5	10
Medium sizes				
5	20	3	5	15
6	25	3	5	20
7	30	4	6	25
8	35	4	6	30
9	55	4	6	45
Large sizes				
10	80	4	6	60
11	100	4	6	80
12	120	4	6	100
13	150	5	7	120
14	180	5	7	150
15	220	5	7	200

Table 4 Summary of test results

Test problems	Cooling rate	SA objective value			CPU time (seconds)			LINGO objective value	LINGO time(seconds)	Error (%)	Standard Deviation	SD/obj. ave.
		Max	Min	Ave	Max	Min	Ave					
1	0.03	97,049	96,395	96,793	6.31	5.75	6.13	96,235	1	0.58	225.95	0.002
	0.05	97,513	96,610	97,034	3.91	3.39	3.51			0.83	163.17	0.002
	0.07	98,082	96,877	97,372	2.68	2.51	2.58			1.18	428.39	0.004
	0.1	98,741	97,605	98,311	1.95	1.68	1.72			2.16	276.96	0.003
	0.15	99,316	98,067	98,718	1.31	1.02	1.01			2.58	209.81	0.002
2	0.03	114,059	113,383	113,760	7.27	6.95	7.12	113,198	1	0.5	221.87	0.002
	0.05	115,167	114,190	114,442	4.28	4.16	4.2			1.1	231.62	0.002
	0.07	117,128	115,204	116,193	3.72	3.65	3.68			2.65	692.93	0.006
	0.1	118,115	116,663	117,171	2.21	2.02	2.12			3.51	460.93	0.004
	0.15	118,876	115,889	117,349	1.48	1.31	1.39			3.67	1007.31	0.009
3	0.03	163,813	162,062	162,928	9.23	8.79	9.01	162,163	1	0.47	680.40	0.004
	0.05	164,346	162,773	163,572	5.63	5.11	5.38			0.87	557.67	0.003
	0.07	165,610	163,875	164,341	3.94	3.66	3.82			1.34	454.88	0.003
	0.1	168,582	166,912	167,504	2.76	2.59	2.62			3.29	282.10	0.002
	0.15	172,391	165,877	168,613	1.86	1.72	1.8			3.98	1377.13	0.008
4	0.03	189,914	188,553	189,012	11.43	10.39	10.83	188,729	1	0.15	242.72	0.001
	0.05	191,246	188,721	189,839	6.62	6.32	6.47			0.59	795.69	0.004
	0.07	196,455	189,016	192,827	4.39	4.28	4.32			2.17	1579.96	0.008
	0.1	196,252	192,337	194,311	3.41	3.18	3.27			2.96	995.49	0.005
	0.15	206,610	191,236	195,859	2.24	2.12	2.18			3.78	3010.35	0.015
5	0.03	208,241	203,638	206,002	13.77	13.38	13.42	204,616	2	0.68	1080.09	0.005
	0.05	210,466	206,051	207,113	8.37	8.01	8.14			1.22	1714.12	0.008
	0.07	217,455	205,898	210,125	5.57	5.29	5.44			2.69	3728.02	0.018
	0.1	215,932	209,115	212,129	4.22	4.01	4.08			3.67	2528.41	0.012
	0.15	224,407	209,324	213,121	2.85	2.59	2.68			4.16	3099.10	0.015
6	0.03	271,673	266,028	268,316	22.61	20.11	21.36	265,152	42	1.19	1239.11	0.005
	0.05	271,379	267,285	269,155	13.53	12.1	12.81			1.51	999.81	0.004
	0.07	281,210	269,133	272,823	9.22	8.15	8.58			2.89	4278.71	0.016
	0.1	284,767	270,059	273,632	6.77	6.11	6.37			3.2	4016.54	0.015
	0.15	290,755	271,115	276,497	4.58	4.02	4.26			4.28	3687.73	0.013
7	0.03	286,612	281,431	284,039	21.81	21.18	21.51	281,428	81	0.93	915.83	0.003
	0.05	291,128	282,003	287,022	13.33	12.49	12.88			1.99	1600.53	0.006
	0.07	301,863	289,637	292,646	8.79	8.51	8.63			3.99	2184.49	0.007
	0.1	302,735	291,844	295,282	6.66	6.4	6.52			4.92	1504.09	0.005

	0.15	301,246	285,886	296,527	4.48	4.11	4.29			5.37	2951.32	0.010
8	0.03	325,076	318,387	321,355	29.73	28.13	28.66	317,919	1412	1.08	1309.66	0.004
	0.05	329,452	319,243	324,178	17.51	17.02	17.24			1.97	1854.01	0.006
	0.07	339,628	321,563	328,225	11.63	11.29	11.47			3.24	3829.05	0.012
	0.1	341,992	324,458	335,418	8.75	8.41	8.56			5.50	3194.49	0.010
	0.15	351,919	329,026	337,175	5.83	5.59	5.72			6.06	5113.65	0.015
9	0.03	421,247	413,168	417,781	42.58	41.88	42.32	411,169	4806	1.61	1427.67	0.003
	0.05	432,245	423,551	425,647	25.89	25.11	25.48			3.52	1744.61	0.004
	0.07	438,527	424,354	429,452	17.31	16.88	17.03			4.45	2788.72	0.006
	0.1	441,899	417,562	430,324	12.93	12.59	12.7			4.66	4543.66	0.011
	0.15	448,934	431,235	438,956	8.65	8.36	8.44			6.76	3709.93	0.008
10	0.03	614,786	596,353	610,588	138.95	121.27	131.15	587,235	16,451	3.98	3109.66	0.005
	0.05	617,992	599,721	612,513	79.83	78.09	78.85			4.30	3827.73	0.006
	0.07	628,117	616,109	620,270	53.33	51.12	52.62			5.63	3208.86	0.005
	0.1	642,853	621,255	628,125	39.88	39.09	39.31			6.96	3763.05	0.006
	0.15	645,628	621,012	632,047	26.62	25.79	26.27			7.63	4045.75	0.006
11	0.03	666,328	639,141	659,542	787.75	774.03	781.45	633,764	31,988	4.07	4734.52	0.007
	0.05	677,570	651,018	663,163	534.91	520.29	528.41			4.64	4718.05	0.007
	0.07	682,691	658,117	671,347	358.9	345.97	352.04			5.93	5066.35	0.008
	0.1	693,426	647,229	677,512	269.96	257.48	265.61			6.90	9658.76	0.014
	0.15	705,947	667,004	681,379	179.77	171.93	176.13			7.51	8320.45	0.012
12	0.03	1,121,900	1,107,597	1,115,163	1417.23	1386.78	1402.42	1,061,802[a]	43,795	5.03	2958.26	0.003
	0.05	1,133,892	10,931,644	1,117,427	1231.16	1201.96	1219.17			5.24	9142.50	0.008
	0.07	1,192,397	1,113,690	1,128,862	1150.27	1124.91	1139.33			6.32	16983.47	0.015
	0.1	1,163,615	1,083,215	1,136,982	935.19	910.22	921.35			7.08	16590.95	0.015
	0.15	1,180,944	1,128,346	1,157,007	631.04	607.55	615.61			8.97	12646.12	0.011
13	0.03	1,618,224	1,554,373	1,582,957	2011.3	1968.03	1981.66	1,527,309[a]	43,987	3.64	10713.18	0.007
	0.05	1,687,021	1,567,468	1,612,390	1812.86	1782.11	1797.38			5.57	20994.82	0.013
	0.07	1,742,669	1,598,790	1,648,903	1624.07	1598.47	1608.27			7.96	24378.81	0.015
	0.1	1,738,022	1,597,502	1,654,751	1327.44	1303.55	1318.49			8.34	29417.07	0.018
	0.15	1,771,869	1,624,578	1,678,552	1174.09	1149.61	1169.85			9.91	27588.58	0.016
14	0.03	2,258,392	2,165,450	2,205,633	2603.33	2549.81	2575.58	2,128,466	44,502	3.63	17811.04	0.008
	0.05	2,283,994	2,186,754	2,238,950	2374.74	2335.69	2349.21			5.19	18837.25	0.008
	0.07	2,374,925	2,172,443	2,263,928	2138.04	2107.11	2128.57			6.36	35953.19	0.016
	0.1	2,349,278	2,216,639	2,271,513	1964.73	1938.27	1950.8			6.72	23605.42	0.010
	0.15	2,491,844	2,248,311	2,335,885	1784.27	1758.74	1772.29			9.74	42335.88	0.018
15	0.03	3,204,592	3,021,693	3,098,121	3254.12	3184.99	3217.44	2,993,585	44,816	3.49	28877.91	0.009
	0.05	3,189,237	3,039,227	3,127,760	2981.26	2927.26	2956.28			4.48	27813.47	0.009

Table 4 (continued)

Test problems	Cooling rate	SA objective value			CPU time (seconds)			LINGO objective value	LINGO time(seconds)	Error (%)	Standard Deviation	SD/obj. ave.
		Max	Min	Ave	Max	Min	Ave					
	0.07	3,268,901	3,122,074	3,176,227	2744.11	2702.61	2723.36			6.10	28307.78	0.009
	0.1	3,319,004	3,096,873	3,192,440	2579.12	2538.33	2549.72			6.64	38532.66	0.012
	0.15	3,383,860	3,146,772	3,281,196	2368.03	2336.8	2354.41			9.61	48805.43	0.015

[a] Lower bound achieved by Lingo after about 12 h

the test problems considered in some studies are listed in Table 2. Since the logistics network discussed in this paper is not exactly the same as those presented in Table 2, therefore, 15 test problems with different sizes are randomly generated to evaluate the performance of the SA algorithm. But, as is illustrated in Tables 2 and 3, the generated large-sized test problems are in the range of the largest existing test problems. The first four test problems are grouped as small-sized, the test problems 5–9 are defined as medium-sized, and the rest are large-sized problems. In Table 3, these test problems and their related number of initial points and fixed points of logistics centers (collection/inspection, recovery, disposal centers, and customer zones) are listed.

The SA is coded in MATLAB 7.0, and LINGO 8.0 software is used to compare the results. All the test problems are solved on Pentium core 2 duo 2.50–2.50 GHz computer with 2 GB RAM. LINGO solves the problems with branch-and-bound algorithm and guaranteed to find the global optimal solution.

To compare the optimal solutions obtained by LINGO with the results of the SA algorithm, a quality criterion, the gap of solution, is defined according to the following equation:

$$\%\ \text{gap} = \frac{(\text{SA}_{\text{answer}} - \text{LINGO}_{\text{answer}})}{\text{LINGO}_{\text{answer}}} \times 100$$

To evaluate the performance of the SA algorithm under different cooling rates, each of the test problems is considered with five different cooling rates (0.03, 0.05, 0.07, 0.1, and 0.15). For each combination of test problems and cooling rates, the SA is run for 10 times. In Table 4, maximum, minimum, and the average of CPU time, the optimal solution, and also the standard deviation of the solutions are presented.

As the computation times show (see Table 4), LINGO takes about 9 and 5 h to find the optimal solution for the test problems 10 and 11, and for problems 12–15, the optimal solution could not be found in 12 h. Thus, the results of the SA algorithm are compared with the lower bound obtained by LINGO in 12 h.

As the results show, the solution gaps vary from 0.58% to 5.03% for cooling rate 0.03, and for cooling rate 0.05, they are from 0.83% to 5.57%. Also, for other cooling rates, the maximum gap for the largest test problem is less than 10%. The comparison between cooling rates indicates that the two cooling rates 0.05 and 0.03 have an acceptable performance with respect to the solution gap criterion. The performance of the SA algorithm under different cooling rates is illustrated in Figs. 8 and 9, with respect to the solution gap and CPU time.

Since the CPU times for the SA algorithm are significantly lower than the LINGO computation times, using the SA algorithm with cooling rates 0.03 and 0.05 is quite

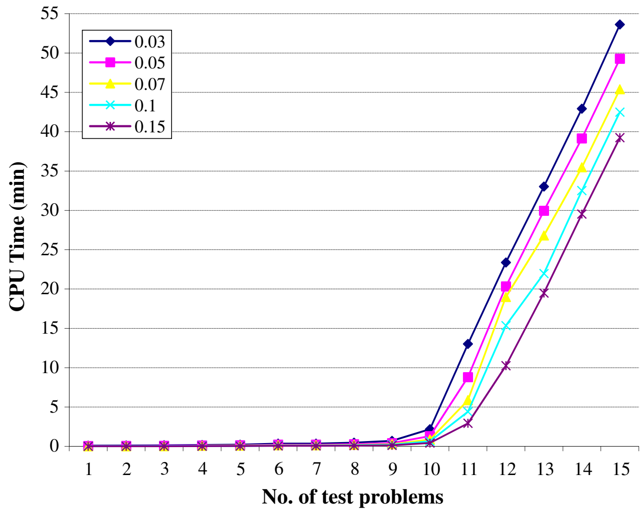


Fig. 8 CPU time of the SA algorithm under different cooling rates

Table 5 Comparison of different cooling rates with t test results

Test problems	Alternative hypothesis			
	$\mu_{0.05-0.03}>0$	$\mu_{0.07-0.05}>0$	$\mu_{0.1-0.07}>0$	$\mu_{0.15-0.1}>0$
1	1.41	3.07[a]	3.16[a]	1.60
2	3.32[a]	6.04[a]	3.14[a]	0.46
3	1.61	2.89[a]	7.83[a]	1.82
4	1.67	5.20[a]	2.36[a]	2.09[a]
5	1.73	2.17[a]	1.03	0.34
6	1.70	2.71[a]	0.29	0.58
7	2.93[a]	3.66[a]	3.31[a]	1.09
8	2.24[a]	2.92[a]	2.56[a]	0.72
9	4.00[a]	2.74[a]	0.35	4.20[a]
10	0.46	3.92[a]	2.75[a]	1.28
11	0.76	2.06[a]	2.05[a]	0.67
12	0.61	0.98	1.46	0.75
13	1.82	3.21[a]	0.33	1.44
14	3.14[a]	1.41	0.68	3.46[a]
15	2.21[a]	2.31[a]	0.62	2.62[a]

[a] The alternative hypothesis is significant at the 0.05 level of significance

acceptable in the case of medium- and large-sized problems. Also, the standard deviation (SD) increases with respect to the size of the test problems. To assess the performance of the SA algorithm with respect to the SD in a more desirable way, we use another measure by dividing the SD by the average of the objective function values. The ratio of the standard deviations per averages of the objective values is less than 0.02 in all of the test problems; thus, it seems that the SA algorithm generates an acceptable range of solutions for each test problem.

In order to evaluate the difference between cooling rates precisely, some statistical test of significance are conducted. To make a paired comparison, for each test problem, two cooling rates are selected. In each test of significance, the mean and standard deviation of the 10 differences are calculated. Each test of significance examines the null hypothesis that the population corresponding to the difference has mean zero, $\mu=0$, against the alternative $\mu>0$. It is assumed that the significance level is $\alpha=0.05$, and the differences are normal random variables. The random variable $t=\frac{\overline{X}-\mu_0}{\left(S/\sqrt{N}\right)}$ has a t distribution with $N-1$ degree of freedom, where N, $\overline{X}$, and S are, respectively, the number, mean, and standard deviation of differences. The null hypothesis rejected if the calculated value of t exceeds t_α, where t_α is such that the area under the t distribution with $N-1$ degree of freedom to its right is equal to $\alpha=0.05$.

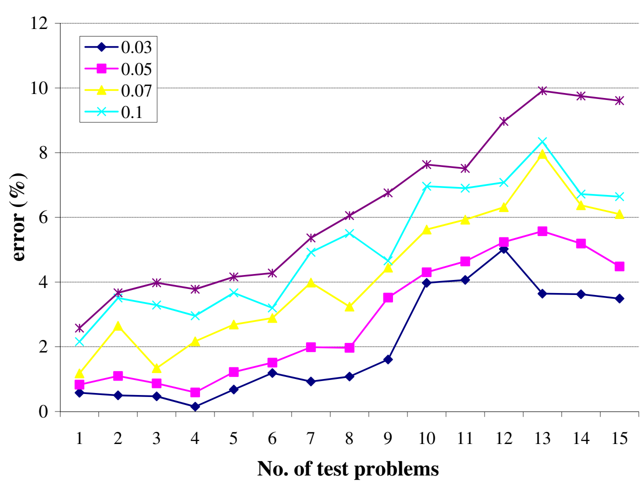


Fig. 9 Gap of the SA algorithm under different cooling rates

Table 5 shows the calculated random variables t and the results of statistical tests of significance. For example, the entry for the test problem 12 and cooling rates 0.03 and 0.05 corresponds to $N=10$, $\mu_0=0$, $\overline{X}=1314.03$, $S=6{,}865.09$, and the sample $t=(1314.03-0)/\left(6865.09/\sqrt{10}\right)=0.61$. Since $t<t_{0.05,9}=1.833$, we conclude for this test problem that the mean of objective functions for cooling rate 0.05 is not statistically greater than cooling rate 0.03.

As seen from Table 5, SA solutions with cooling rate 0.03 are statistically significantly different from their 0.05 cooling rate counterparts in only six of the 15 test problems, but the comparison between cooling rates 0.05 and 0.07 shows the statistical difference in 14 test problems. Thus, SA algorithms with cooling rates 0.03 and 0.05, with respect to their acceptable CPU times and gaps, are preferred to solve the problem.

6 Conclusions

In this paper, an MILP model is developed for multistage reverse logistics network design to minimize the total costs including fixed opening costs and transportation costs. The studied reverse logistics network includes customer zones, collection/inspection, recovery, and disposal centers, with limited capacities. The proposed model is able to find locations and the number of collection/inspection centers and also the quantity of transportation between facilities.

An SA algorithm with dynamic neighborhood search mechanisms is used to find the near optimal solution of the proposed model for large-sized problems. Also, some test problems in small, medium, and large sizes are solved, whose results are proven to be satisfactorily close to the exact solutions obtained by LINGO 8.0 optimization software.

The followings are proposed for future researches:

- Besides minimizing the costs, other objective functions such as "responsiveness" and "robustness" could be considered in designing the reverse logistics networks; therefore, the applied algorithm should be changed to achieve the ability of solving multiobjective models.
- In this paper, we consider a single product network with deterministic returned products; however, in many real cases, we have a multiproduct network with uncertainty in returns. Therefore, considering these assumptions can be a subject for future researches.
- The applied neighborhood search methods and also the priority-based encoding method can be used in other meta-heuristic algorithms such as tabu search and scatter search.

Appendix 1

Decoding algorithm proposed by Gen et al. [8] for priority-based encoding.

Inputs: $\boldsymbol{K}$: set of sources
$\boldsymbol{J}$: set of depots
$\boldsymbol{b_j}$: demand on depot j
$\boldsymbol{a_k}$: capacity of source a
$\boldsymbol{c_{kj}}$: transportation cost of one unit of product from source k to depot j
$\boldsymbol{v(j+j)}$: encoded solution
Output: $\boldsymbol{g_{kj}}$: amount of shipment between nodes
While $v(|k|+j) \neq 0\,,\ \forall j \in J$
Step 1: $g_{kj} = 0 \quad \forall j \in J\,,\ k \in K$
Step 2: select a node based on $l = \arg\max\{v(t), t \in |K| + |J|\} \forall j \in J, k \in K$
Step 3: if $l \in k$ then a source is selected $k^*=l$
$j^* = \arg\min\{c_{kj}|v(j) \neq 0, j \in J$ Select a depot with minimum cost
else $j^*=l$ a depot is selected
$k^* = \arg\min\{c_{kj}|v(j) \neq 0, k \in K$ Select a source with minimum cost
Step 4: $g_{k^*j^*}=\min(a_{k^*}, b_{j^*})$
Update demands and capacities
$a_{k^*}=a_{k^*}-g_{k^*j^*}$ and $b_{j^*}=b_{j^*}-g_{k^*j^*}$
Step 5: if $a_{k^*}=0$ *then* $(k^*)=0$
if $b_{j^*}=0$ *then* $v(j^*)=0$

References

1. Aras N, Aksen D, Tanugur AG (2008) Locating collection centers for incentive-dependent returns under a pickup policy with capacitated vehicles. Eur J Operat Res 191:1223–1240
2. Aarts E, Lenstra JK (1997) Local search in combinatorial optimization. Wiley, New York
3. Arts E, Korst J (1989) Simulated annealing and Boltzmann machines: a stochastic approach to combinatorial optimization and neural computing. Wiley, Chichester
4. Davis PS, Ray TL (1969) A branch-and-bound algorithm for the capacitated facilities location problem. Nav Res Logist 16:331–344
5. Du F, Evans GW (2008) A biobjective reverse logistics network analysis for post-sale service. Comput Oper Res 35:2617–2634
6. Fleischmann M, Beullens P, Bloemhof-ruwaard JM, Wassenhohve V (2001) The impact of product recovery on logistics network design. Prod Oper Manag 10:156–173
7. Gen M, Cheng R (2000) Genetic algorithms and engineering optimization. Wiley, New York
8. Gen M, Altiparmak F, Lin L (2006) A genetic algorithm for two-stage transportation problem using priority-based encoding. OR Spectrum 28:337–354
9. Jayaraman V, Guige VDR Jr, Srivastava R (1999) A closed-loop logistics model for remanufacturing. J Oper Res Soc 50:497–508
10. Jayaraman V, Ross A (2003) A simulated annealing methodology to distribution network design and management. Eur J Oper Res 144:629–645
11. Jayaraman V, Patterson RA, Rolland E (2003) The design of reverse distribution networks: models and solution procedures. Eur J Oper Res 150:128–149
12. Krikke HR, Van Harten A, Schuur PC (1999) Reverse logistic network redesign for copiers. OR Spektrum 21:381–409
13. Kirkpatrick S, Gelatt CD Jr, Vecchi MP (1983) Optimization by simulated annealing. Science 220:671–680
14. Laha D, Chakraborty UK (2007) An efficient stochastic hybrid heuristic for flowshop scheduling. Eng Appl Artif Intell 20:851–856
15. Laha D, Chakraborty UK (2008) An efficient hybrid heuristic for makespan minimization in permutation flow shop scheduling. Int J Adv Manuf Technol. doi:10.1007/s00170-008-1845-2
16. Listes O, Dekker R (2005) A stochastic approach to a case study for product recovery network design. Eur J Oper Res 160:268–287
17. Lu Z, Bostel N (2007) A facility location model for logistics systems including reverse flows: the case of remanufacturing activities. Comput Oper Res 34:299–323
18. Meade L, Sarkis J, Presley A (2007) The theory and practice of reverse logistics. Int J Logist Syst Manag 3:56–84
19. Meepetchdee Y, Shah N (2007) Logistical network design with robustness and complexity considerations. Int J Phys Distrib Logist Manag 37:201–222

20. Michalewicz Z, Vignaux GA, Hobbs M (1991) A non-standard genetic algorithm for the transportation problem. ORSA J Comput 3:307–316
21. Min H, Ko CS, Ko HJ (2006) The spatial and temporal consolidation of returned products in a closed-loop supply chain network. Comput Ind Eng 51:309–320
22. Pirlot M (1992) General local search heuristics in combinatorial optimization: a tutorial. Belg J Oper Res, Stat Comput Sci 32:7–68
23. Üster H, Easwaran G, Akçali E, Çetinkaya S (2007) Benders decomposition with alternative multiple cuts for a multi-product closed-loop supply chain network design model. Nav Res Logist 54:890–907
24. Wojanowski R, Verter V, Boyaci T (2007) Retail–collection network design under deposit–refund. Comput Oper Res 34:324–345