ARCHIVES OF TRANSPORT *ISSN (print):* *0866-9546*
Volume 47, Issue 3, 2018 ***e-ISSN (online):*** *2300-8830*
DOI: *10.5604/01.3001.0012.6503*

REVERSE LOGISTICS NETWORK PROBLEM USING SIMULATED ANNEALING WITH AND WITHOUT PRIORITY-ALGORITHM

Mounir BENAISSA[1], Ilhem SLAMA[2], Mohamed Mahjoub DHIAF[3]

[1] *University of Sfax, OASIS Laboratory, ISGI, Sfax, Tunisia*
[2] *University of Sfax, MODILS Research Unit, FSEG Sfax, Tunisia*
[3] *Emirates College of Technology, Abu Dhabi, UAE*

Contact:
1) benaissamounir@ieee.org, 2) ilhemslm.fsegs@gmail.com, 3) dhiafmohamed@yahoo.fr

Abstract:

In recent years, Reverse Logistics (RL) has become a field of importance for all organizations due to growing environmental concerns, legislation, corporate social responsibility and sustainable competitiveness. In Reverse logistics, the used or returned products are collected after their acquisition and inspected for sorting into the different categories. The next step is to disposition them for repair, remanufacturing, recycling, reuse or final disposal. Manufacturers may adopt reverse logistics by choice or by force, but they have to decide whether performing the activities themselves or outsourcing to a third party (Martin et al., 2010). Lourenço et al., (2003) described three main areas of improvement within the RL process. Firstly, companies can reduce the level of returns through the analysis of their causes. Secondly, they can work on the improvement of the return's process and, thirdly, they can create value from the returns. This paper considers the multistage reverse Logistics Network Problem (mrLNP) proposed by Lee et al., (2008). With minimizing the total of costs to reverse logistics shipping cost. We will demonstrate the mrLNP model will be formulated as a three-stage logistics network model. Since such network design problems belong to the class of NP-hard problems we propose a Simulated Annealing (SA) and simulated annealing with priority (priSA) with special neighborhood search mechanisms to find the near optimal solution consisting of two stages. Computer simulations show the several numerical examples by using, SA, priSA and priGA(Genetic algorithm with priority-based encoding method) and effectiveness of the proposed method.

Key words:

Simulated Annealing (SA), multistage reverse Logistics Network Problem (mrLNP), Priority-based encoding

To cite this article:

Benaissa, M., Slama, I., Dhiaf, M. M., 2018. Reverse Logistics Network Problem using simulated annealing with and without Priority-algorithm. Archives of Transport, 47(3), 7-17. DOI: https://doi.org/10.5604/01.3001.0012.6503

1. Introduction

Recently recovery of used products and product recovery are becoming increasingly important for economic, environmental or legislative. A reverse logistics system includes a series of activities, which form a continuous process to treat back-products until they are well recovered or disposed. After customers make their products already used, these products will be sent to the disassembly; in a 2nd stage the products are then completely disassembled. Each component must be examined and inspected to determine its status. Three choices available are possible for subsets that are in a functional state, they will be sent to the manufacturer in order to rebuild them, however subsets that can be reused are transported to a recycling center and finally the pieces are in an unusable state are being discarded.

Recently the reverse logistics holds a great attention because of the legislative growing concern for the environment and economic opportunities for the minimization costs or income from returned products. In recent decades, various mathematical models for the design of the RL network have been proposed in the literature (Ilgin and Grupta,2010). (Lewczuk, 2015) proposed to using genetic algorithm to support organization of internal transport processes in logistics facilities. Barros et al., (1998) proposed a mixed programming model integer (MILP) multi-level to solve a warehouse location problem using heuristic procedures. The model determined the optimal capacity, and deposit locations. Kirkke et al.,(1999) presented an MILP model based on a multi-level incapacitated warehouse location model. They described a case study, dealing with a reverse logistics network for the returns, processing, and recovery of discarded copiers. The model was used to determine the locations and capacities of the recovery facilities. Karkula (2014) presented an analysis of selected aspects of modelling the internal transport systems using methods of stochastic discrete event simulation. Jayaraman et al., (1999), proposed an MILP model to determine the optimal number and locations of distribution/remanufacturing facilities for electronic equipment. Jayaraman et al.,(2003) developed a mixed integer programming model and solution procedure for a reverse distribution problem focused on the strategic level. The model determines whether each remanufacturing facility is open considering the product return flow. (Min et al.,2005) proposed a Lagrangian relaxation heuristics to design the multi-commodity, multi-echelon reverse logistics network. (Kim et al.,2006) proposed a general framework for remanufacturing environment and a mathematical model to maximize the total cost saving. The model determines the quantity of products/parts processed in the remanufacturing facilities/subcontractors and the amount of parts purchased from the external suppliers while maximizing the total remanufacturing cost saving. (Min et al.,2006) proposed a nonlinear mixed integer programming model and a genetic algorithm that can solve the reverse logistics problem involving product returns. Their study proposes a mathematical model and GA which aim to provide a minimum-cost solution for the reverse logistics network design problem involving product returns. Ko et al., (2007) presented a mixed integer nonlinear programming model for the design of a dynamic integrated distribution network to account for the integrated aspect of optimizing the forward and return network simultaneously. They also proposed a genetic algorithm-based heuristic for solving this problem. Lee et al., (2008)., Proposed a multi-stage, multi-product, MILP model for minimizing the total of costs to reverse logistics shipping cost and fixed opening cost of facilities. They also proposed a hybrid genetic algorithm for solving this problem. (Żochowska and Soczówka, 2018) analyses the assessment of selected structures of a transportation network based on graph measures.

In this article, we took as support the reverse logistics network model multistage proposed by Lee et al., (2008)., our contribution is to change the resolution approach to finding better. We will choose the default approach simulated annealing (SA) based on research method of dynamic neighborhood that consists of two decoding part, we will make the model resolution by simulated annealing algorithm in the first place without the algorithm (SA) priority and secondly with the priority algorithm (priSA) which was proposed with Gen et al. (2006). Then we compare our approach with priority-based genetic algorithm (PRIGA), which is the approach most used logistic models.

This paper is organized as follows: in Section 2, the mathematical model (mrLNP) proposed by Lee et al., (2008)..is introduced, in Section 3,an efficient simulated annealing (SA) algorithm is applied to solve the model with and without adapting the encoding method based on the priority and a dynamic

neighborhood search strategy that enhances the performance of the applied algorithm are the main contributions of this paper; Section 4, numerical experiments are presented to demonstrate the effectiveness of the proposed approach; Finally, in Section 5, we close our work a little insight on the procedures followed during the entire research and open new perspectives.

2. Problem definition and mathematical modeling

The reverse logistics network discussed in this paper is a multi-stage logistics network, includes the areas of return, disassembly, processing and manufacturing with limited capabilities. Fig. 1 illustrates the general outline of a multistage reverse logistics network model proposed by Lee et al., (2008)..In the first stage and after that customers return their products which are already used, these products are transported in the disassembly center for disassembled. Three cases are possible for the shipping of disassembled parts, the first case is the 2nd stage of this model, and the parts of which the treatment process is necessary for reuse are transported to the processing center. Second, the disassembled parts that can be reused without any treatment process are transported to a recycling center. Finally, the parts that are not functional are transported in disposal
In the 3rd stage, the parts treated are transported to the manufacturer
The basic assumptions in this model m-rLNP (multi stage Reverse Logistics Network problem) are:
A1: If the quantity of provided parts from processing center is not enough for requirement of manufacturer, then manufacturer must buy parts from supplier.
A2: If the quantity of provided parts from processing center exceeds the requirement of manufacturer, then exceeded capacities distribute in order of recycling and disposal.
A3: Not considering the inventories.
A4: The parts demands of manufacturer are known in advance.
A5: Recycle rate and disposal rate are 1% and 0.5% of capacity of part each other on the disassembly process. Only, round off the numbers to one decimal place of result.

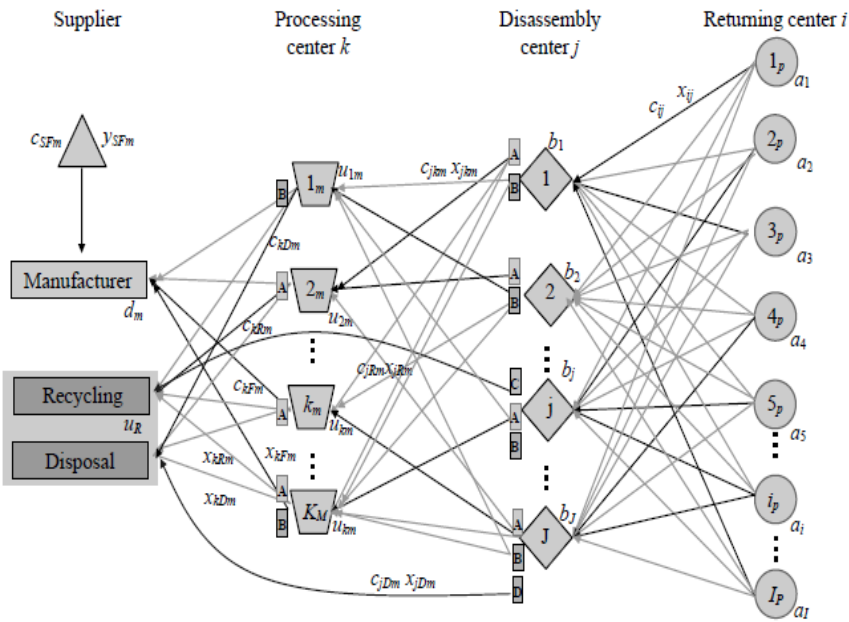

Fig. 1. Multistage reverse logistics network model m-rLNP (Lee et al., 2008).

The indices, parameters and decision variables used in the mathematical formulation are:

Indices:
i: index of returning center ($i = 1, 2, \ldots, I$)
j: index of disassembly center ($j = 1, 2, \ldots, J$)
k: index of processing center ($k = 1, 2, \ldots, K$)
m: index of part ($m = 1, 2, \ldots, M$)

Parameters:
I: number of returning centers
J: number of disassembly centers
K: number of processing centers
M: number of parts
a_i: capacity of returning center i
b_j: capacity of disassembly center j
u_{km}: capacity of processing center k for parts m
$u_{R:}$ capacity of recycling R
$d_{m:}$ demand of parts m in manufacturer F
$n_{m:}$ the number of parts m from disassembling one unit of product
r_R: Recycling rate
r_D: Disposal rate
c_{ij}: unit cost of transportation from returning center i to disassembly center j
c_{jkm}: unit cost of transportation from disassembly center j to processing center k
c_{iRm}: unit cost of transportation from disassembly center j to recycling R
c_{iDm}: unit cost of transportation from disassembly center j to disposal D
c_{kFm}: unit cost of transportation from processing center k to manufacturer F
c_{kRm}: unit cost of transportation from processing center k to recycling R
c_{kDm}: unit cost of transportation from processing center k to disposal D
c_{SFm}: unit cost of transportation from supplier S to manufacturer F

Decision Variables:
x_{ij}: amount shipped from returning center i to disassembly center j
x_{jkm}: amount shipped from disassembly center j to processing center k
x_{jRm}: amount shipped from disassembly center j to recycling R
x_{jDm}: amount shipped from disassembly center j to disposal D
x_{kFm}: amount shipped from processing center k to manufacturer F
x_{kRm}: amount shipped from processing center k to recycling R
x_{kDm}: amount shipped from processing center k to disposal D
y_{SFm}: amount shipped from supplier S to manufacturer F

The mathematical model of the problem is:

$$\textbf{Minimize } Z = \sum_{i=1}^{I}\sum_{j=1}^{J} C_{ij}\, x_{ij} + \sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{m=1}^{M} c_{jkm}\, x_{jkm} - \sum_{j=1}^{J}\sum_{m=1}^{M} C_{jRm}\, X_{jRm} + \sum_{j=1}^{J}\sum_{m=1}^{M} C_{jDm}\, X_{jDm} + \sum_{k=1}^{K}\sum_{m=1}^{M} C_{kFm}\, X_{kFm} - \sum_{K=1}^{J}\sum_{m=1}^{M} C_{KRm}\, X_{KRm} + \sum_{k=1}^{k}\sum_{m=1}^{M} C_{kDm}\, X_{kDm} + \sum_{m=1}^{M} C_{SFm}\, Y_{SFm} \quad (1)$$

S/C

$$\sum_{j=1}^{J} X_{ij} \le a_i, \forall\, i \quad (2)$$

$$\sum_{k=1}^{K} X_{jkm} + X_{jRm+}\, X_{jDm} \le b_{jm}, \forall\, j,m \quad (3)$$

$$\sum_{k=1}^{K} (X_{kFm} + X_{kRm} + X_{kDm}) \le U_{km}, \forall\, m \quad (4)$$

$$Y_{SFm} + \sum_{k=1}^{K} X_{kFm} \ge d_{m}, \forall m \quad (5)$$

$$X_{ij,}\, X_{jkm}\, X_{jdm}\, X_{jrm}\, X_{kfm}\, X_{krm}\, X_{kdm}\, Y_{sfm} \ge 0\ \forall m, j,k,i \quad (6)$$

Fig.2 The mathematical model (Lee et al., 2008).

The objective function (1) minimizes the cost of transportation between the centers (from the shipment of the products used to manufacture the disassembly and reassembly in reusable together). Constraint (2) ensures that the quantity of products considerate back center dismantling facility does not exceed the total capacity of center back. Constraint (3) ensures that the amount subsets ships dismantling center to the processing center or recycling or disposal, does not exceed the total capacity of dismantling center. Constraint (4) ensures that the amount subsets manufacturer sent to the processing center or recycling or disposal, does not exceed the total capacity of manufacturer center. Equation (5) provides that the manufacturer of demand must be met by the processing center and by the supplier, and finally the stress (6) applies the non-negativity restriction on the decision variables used in this model.
We know that the design problem of reverse logistics network is a problem NP-hard combinatorial optimization. Given the complexity of this problem, the resolution of this model 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,

the simulated annealing algorithm with and without priority is applied, we will then develop a comparative literature emanated.

3. Proposed The simulated annealing algorithm

The algorithm of Simulated Annealing (SA) was proposed (Kirkpatrick et al.,1983) Its principle is based on the process of annealing of metals used by metallurgists .the metheuristic simulated annealing is inspired by the Metropolis algorithm (Metropolis et al.,1953) the principle (for a minimization problem) can be summarized as follows: The search starts from an initial feasible solution. Each solution has a specific value of the costs. A small change in one or a combination of some variables can generate a similar solution with a different value of cost. In simulated annealing, the next solution is randomly generated. If the value of the cost of the candidate solution is lower than the current solution, the transition to the candidate solution is made. However, if the candidate does not improve the current solution, there is still a transition opportunity as a probability function based on the "Boltzmann".

The simulated annealing algorithm we have chosen to make the resolution consists of two component decoding; we will make the resolution in the first step without the priority algorithm and in 2nd step with priority algorithm.

3.1. Definition of the neighborhood searches

In the simulated annealing algorithm, three neighborhood searching methods are used in each selected segment; the value in some cells will be exchanged with each other:

- 2-opt mutation In the 2-opt perturbation scheme, two cells in the vector are exchanged with each other(Fig.3).
- Inverse mutation in inverse mutation, a length of the vector is selected and the values in the segment are reversed (Fig.4).
- mutation by insertion : we will choose two random positions and we will shift all the cells between these two positions, then we need to insert the position of the first cell to the last position(Fig.5)
- The logic of the use of different types of search neighborhood in the simulated annealing algorithm, described above, is illustrated in Fig 3-5.

2	7	1	5	3	8	6	9	4

Before mutation

2	3	1	5	7	8	6	7	4

After mutation

Fig. 3. 2-Opt mutation schema

2	7	1	5	3	8	6	9	4

Before mutation

2	7	5	3	8	1	6	9	4

After mutation

Fig. 4. Insertion mutation scheme.

2	7	1	5	3	8	6	9	4

Before mutation

9	7	1	6	8	3	5	2	4

After mutation

Fig. 5. Inverse mutation scheme.

According to the above explanations, the SA algorithm and the other parameters of the SA algorithm are shown in Fig 6, 7:

Algorithm 1 Simulated annealing

Step 1: *Initial temperature=100, Frozen state=0.001 , Cooling rate=0.9,*
Step 2: Generate *random* solutions of size (p)
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 *P*
Ti=Temperature
Create a new solution using *k* neighborhood search algorithm *(Xnew)*
Decode *Xnew* using multi-stage decoding algorithm
if $f(X^*) - f(Xnew) \geq 0$ **then** $f(X^*) = f(Xnew)$, $X^*=Xnew$
Else if *Exp* (f (X*) – f (Xnew)/ Ti)>*Random [0 ,1]* **then** $f(X^*) = f(Xnew)$, $X^*=Xnew$
Temperature=Temperature*0.9
Step 5: Return the best solution

Fig. 6. The pseudo code of the SA algorithm

Algorithm 2 Neighborhood search algorithm
X_1 =Mutation-2opt (X_0)
X*=Simulated annealing (X 1)
X_2 =Mutation by insertion (X_1)
X*=Simulated annealing (X 2)
X_3 =Mutation inverse (X_2)
X*=Simulated annealing (X 3)

Fig. 7. The pseudo code of the Neighborhood search algorithm

3.2. Decoding of solutions without priority based method

In this section, first we introduce the simulated annealing algorithm (SA) for solving 1st stage and 2nd stage sub-problems of mrLNP. In this process, solutions are encoded as a vector of size | N | and the position of each cell represents sources (Fig.8) . To decode a solution, the algorithm starts from the first center of the source. In each iteration, the first center of the source is selected and connected to the first depot whose capacity is not saturated. 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 sources end their shipments and demand of depot are met

As an example, we consider the problem that has 5 return centers and 2 disassembly centers (Fig.8).

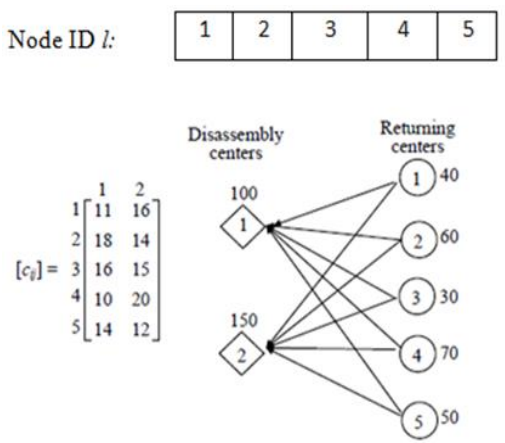

Fig. 8. An illustration of the basic solution of the transport shaft and transport costs for 1st stage on mrLNP

Table 1. Trace table of 1st stage decoding procedure

Iteration	*V(i+j)*	*a*	*b*	*I*	*J*	X_{ij}
0	[12345]	(40,60,30,70,50)	(100,150)	1	1	40
1	[02345]	(0, 60,30,70,50)	(60,150)	2	1	60
2	[00345]	(0,0, 30,70,50)	(0,150)	3	2	30
3	[00045]	(0,0,0,70,50)	(0,120)	4	2	70
4	[00005]	(0,0,0,0,50)	(0,50)	5	2	50
5	[00000]	(0,0,0,0,0)	(0,0)			

As we see in Table 1 to the first procedure of decoding step, between returning center (1) and disassembly center (1). After determining the amount of shipping that is X11 = min {40, 100} = 40, capacity of returning center and disassembly center are updated a1 = 40- 40 = 0, b1 = 100- 40 = 60. Then the value of returning center (1) is set to 0, and the next iteration in the returning center (2) is selected. This process is repeated until all returning centers finish their shipping and need to be satisfied disassembly center. We will follow the same decoding process for the second stage.

The 2nd stage is transportation sub problem between 3 disassembly centers and 3 processing center, first, we calculate recycle rate and disposal rate. Then subtract calculated result from capacity of part. Lastly, we consider processable processing center according to type of part (Fig.10). The decoding algorithm of 2st stage priority-based decoding and its trace table are given in Fig.11 and Table 2.

Procedure1.2: 1st stage decoding
Input: *I:* number of returning centers
J: number of Disassembly centers
a_i: capacity of Returning center i, $\forall i \in I$
b_j: capacity of Disassembly center j, $\forall j \in J$
c_{ij}: shipping cost of one unit product from i to j
$v1$ (i): encoded solution $\forall i \in I$, $\forall j \in J$
Output: x_{ij}: the amount of shipment from i to j
Step 0: $x_{ij} \leftarrow 0$, $\forall i \in I$, $\forall j \in J$
Step 1: select a node (Disassembly and Processing center)
Step 2: $xi^*j^* \leftarrow$ **min** {a_i^*, b_j^*}; assign available amount of units
Update the availabilities on i (a_i^*) and j (b_j^*)
$ai^* = ai^* - xi^*j^*$, and $bj^* = bj^* - xi^*j^*$
Step 3: **if** $ai^* = 0$, **then** $v1(i^*) \leftarrow 0$
If $bj^* = 0$, **then** $v1(I + j^*) \leftarrow 0$
Step 4: **if** $v1(I + j) = 0$, $\forall j \in J$, **output** x_{ij}
Else return step 1

Fig.9. Decoding procedure for 1st stage

Table 2. Trace table of 2nd stage decoding procedure

Iteration	V(i+j)	B	U	J	K	X_{kj}
0	[123456]	(78,157,59,118,108, 216)	(200,200,350)	1	1	78
1	[023456]	(0, 157, 59, 118,108, 216)	(122,200,350)	2	2	157
2	[003456]	(0, 0, 59, 118,108, 216)	(122,43,350)	3	1	59
3	[000456]	(0, 0, 0, 118,108, 216)	(63, 43,350)	4	2	43
4	[000456]	(0, 0, 0, 75,108, 216)	(63, 0,350)	4	3	75
5	[000056]	(0, 0, 0, 0,108, 216)	(63, 0,275)	5	1	63
5	[000056]	(0, 0, 0, 0,45, 216)	(0, 0,275)	5	3	45
6	[000006]	(0, 0, 0, 0,0, 216)	(0, 0,230)	6	3	216
7	[000000]	(0,0,0,0,0,0)	(0,0,14)			

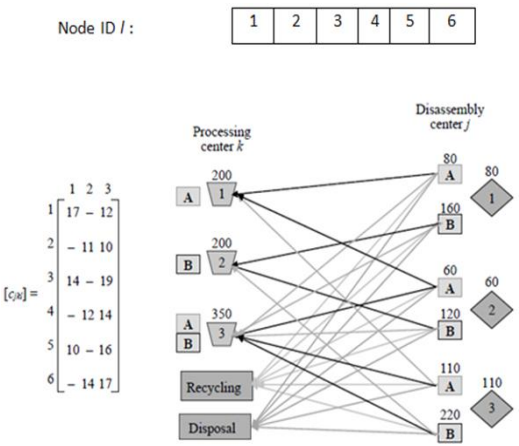

Fig. 10. An illustration of the basic solution of the transport shaft and transport costs for 2nd stage on mrLNP

3.3. Decoding of solutions using priority based method

In this section, first we introduce the priority-based simulated annealing algorithm (priSA) for solving 1st stage and 2nd stage sub-problems of mrLNP.

Gen et al., (2006) presented a priority-based encoding algorithm as an alternative that does not need any excessive repair mechanism. 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. 12). 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., (2006), we refer the readers to Fig. 13.

Procedure 2.2: 2nd stage decoding
Input: J: number of Disassembly centers, K: number of processing centers,
r_R: Recycling rate, r_D: Disposal rate,
n_m: the number of parts m from disassembling one unit of product
$b_{j:}$ capacity of Disassembly center j, $\forall j \in J$
u_{km}: capacity of Processing center k, $\forall k \in K$
u_R: capacity of Recycling
c_{jkm}: unit cost of transportation from j to k
c_{jRm}: unt cost of transportation from j to Recycling R
c_{jDm}: unit cost of transportation from j to Disposal D
$v2$ (k): encoded solution , $\forall j \in J$,
Output: $x_{jkm:}$ the amount of shipment from j to k
Step 0: Calculate recycle rate and disposal rate.

$$b_{jm} = \sum_{i=1}^{I} (1 - rR - rD)\, n_m\, x_{ij} \quad \forall j,\ m$$

Step 1: $x_{jkm} \leftarrow 0$, $\forall j \in J$, $\forall k \in K$
Step 2: select a node (Disassembly and processing center)
Step 3: $X_{j^*k^*m} \leftarrow$ **min** $\{b_{j^*m,}\ u_{k^*m}\}$; assign available amount of units update the availabilities on $j(b_{j^*m})$ and $k(uk^*l)$
$b_{j^*m} = b_{j^*m} - X_{j^*k^*}m$ and $U_{k^*m} = U_{k^*m} - X_{j^*k^*m}$; update the availability
Step 4: **if** $bj_{*m} = 0$, **then** $v2(j^*) = 0$
If $u_{k^*m} = 0$, **then** $v2(J + k^*) = 0$
Step 5: **if** $v2(J + k) = 0$, $\forall k \in K$, **output** $x_{jkm,}$
Else return step 2

Fig.11. Decoding procedure for 2nd stage

For mrLNP, we use two priority-based encodings to represent the transportation trees on stages. This means that each segment consists of two parts. While the first part (i.e., the first priority-based encoding) represents transportation tree between return centers and disassembly centers, the second part (i.e., the second priority-based encoding) represents transportation tree between disassembly centers and processing centers

The 1ST stage is transportation sub problem between 5 return centers and 2 disassembly centers (Fig.12). The decoding procedure of 1ST stage priority-based decoding and its trace table are given in Fig. 13 and Table 3.

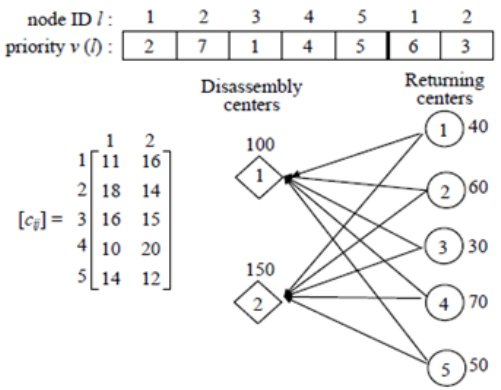

Fig. 12. An illustration of the basic solution of the transport shaft and transport costs for 1st stage on mrLNP

Table 3. Trace table of 1st stage decoding procedure

Iteration	*V(i+j)*	*a*	*b*	*I*	*J*	X_{ij}
0	[2 7 1 4 5⁄6 3]	(40, 60, 30, 70, 50)	(100, 150)	2	2	60
1	[2 0 1 4 5⁄6 3]	(40, 0, 30, 70, 50)	(100, 90)	4	1	70
2	[2 0 1 0 5⁄6 3]	(40, 0, 30, 0, 50)	(30, 90)	1	1	30
3	[2 0 1 0 5⁄0 3]	(10, 0, 30, 0, 50)	(0, 90)	5	2	50
4	[2 0 1 0 0⁄0 3]	(10, 0, 30, 0, 0)	(0, 40)	3	2	30
5	[2 0 0 0 0⁄0 3]	(10, 0, 0, 0, 0)	(0, 10)	1	2	10
	[0 0 0 0 0⁄0 0]	(0, 0, 0, 0, 0)	(0, 0)			

As it is seen in trace table, at the first step of decoding procedure, between returning center 1 and disassembly center 4 is added to transportation tree since returning center 1 has highest priority in the chromosome and the lowest cost is between returning centers 2 and disassembly centers 2. After determining the amount of shipment that is x22 = min {60, 150} = 60, capacity of returning center and disassembly center are updated as a2 = 60 – 60 = 0, b2 = 150 – 60 = 90, respectively. Since a2 = 0, the priority of disassembly center 2 is set to 0, and disassembly center 4 with next highest priority is selected. After adding between disassembly center 4 and returning center 1, the amount of shipment between them is determined and their capacity are updated as it is explained above, and this process repeats until capacities of all disassembly centers are met.

The 2nd stage decoding method is the same with in procedure 1.1 of 1st stage encoding. The 2nd stage is transportation sub problem between 3 disassembly centers and 3 processing center .In the 2nd stage, first, we calculate recycle rate and disposal rate. Then subtract calculated result from capacity of part. Lastly, we consider processable processing center according to type of part. The decoding procedure of 2nd stage priority-based decoding and its trace table are given in Fig. 15 and Table 4

Procedure1.2: 1st stage decoding
Input: *I:* number of returning centers
J: number of Disassembly centers
a_i: capacity of Returning center i, $\forall\, i \in I$
b_j: capacity of Disassembly center j, $\forall\, j \in J$
c_{ij}: shipping cost of one unit product from i to j
$v1\,(i + j)$: encoded solution $\forall\, i \in I, \forall\, j \in J$
Output: x_{ij}: the amount of shipment from i to j
Step 0: $x_{ij} \leftarrow 0$, $\forall\, i \in I, \forall\, j \in J$
Step 1: $l \leftarrow$ **argmax** $\{v1(t), t \in I + J\}$; select a node
Step 2: **if** $l \in I$, **then** $i \leftarrow l$; select a Returning center $j^* \leftarrow$ **argmin**$\{c_{ij} | v1(j) \neq 0, j \in J\}$; select a j with lowest cost
Else $j \leftarrow l$: select a Disassembly center $i^* \leftarrow$ **argmin**$\{c_{ij} | v1(i) \neq 0, i \in I\}$; select a i with lowest cost.
Step 3: $xi^*j^* \leftarrow$ **min** $\{a_i^*, b_j^*\}$; assign available amount of units
Update the availabilities on i (a_i^*) and j (b_j^*)
$ai^* = ai^* - xi^*j^*$, and $bj^* = bj^* - xi^*j^*$
Step 4: **if** $ai^* = 0$, **then** $v1(i^*) \leftarrow 0$
If $bj^* = 0$, **then** $v1(I + j^*) \leftarrow 0$
Step 5: **if** $v1(I + j) = 0$, $\forall\, j \in J$, **output** x_{ij}
Else return step 1

Fig.13. Decoding procedure for 1st stage

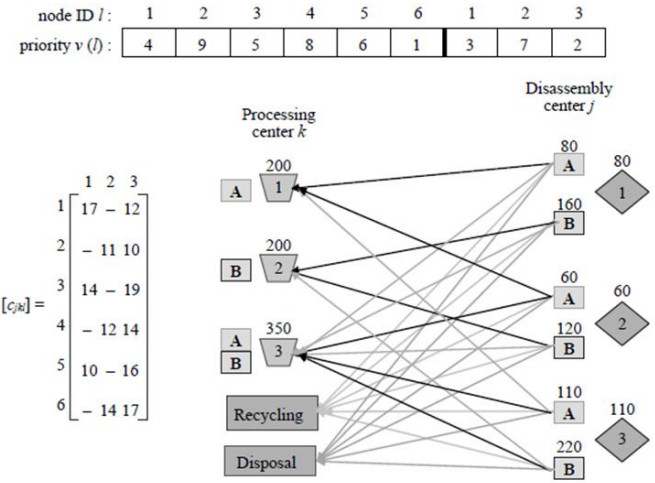

Fig. 14. An illustration of the basic solution of the transport shaft and transport costs for 2nd stage on mrLNP

Procedure 2.2: 2nd stage decoding
Input: J: number of Disassembly centers, K: number of processing centers
R: Recycling, D: Disposal, r_R: Recycling rate,
r_D: Disposal rate,
n_m: the number of parts m from disassembling one unit of product
b_j: capacity of Disassembly center j, $\forall\, j \in J$
u_{km}: capacity of Processing center k, $\forall\, k \in K$
u_R: capacity of Recycling
c_{jkm}: unit cost of transportation from j to k
c_{jRm}: unit cost of transportation from j to Recycling R
c_{jDm}: unit cost of transportation from j to Disposal D
$v2\ (j + k)$: encoded solution, $\forall$ j $\in J$, $\forall\, k \in K$
Output: x_{jkm}: the amount of shipment from j to k
Step 0: Calculate recycle rate and disposal rate.

$$b_{jm} = \sum_{i=1}^{l} (1 - rR - rD)\, n_m\, x_{ij} \;\forall\, j\ ,\ m$$

Step 1: $x_{jkm} \leftarrow 0$, $\forall\, j \in J$, $\forall\, k \in K$
Step 2: $l \leftarrow$ **argmax**$\{v2(t), t \in J + K\}$; select a node
Step 3: **if** $l \in J$, **then** $j \leftarrow l$; select a Disassembly center
$j^* \leftarrow$ **argmin**$\{c_{jkm} | v2(j) \neq 0, j \in J\}$; select a k with lowest cost
Else $k \leftarrow l$: select a Processing center
$k^* \leftarrow$ **argmin**$\{c_{jkm} | v2(\mathrm{k}) \neq 0, k \in K\}$; select a j with lowest cost
Step 4: $X_{j^*k^*m} \leftarrow$ **min** $\{b_{j^*m}, u_{k^*m}\}$; assign available amount of units update the availabilities on $j(b_{j^*m})$ and $k(uk^*l)$
$b_{j^*m} = b_{j^*m} - X_{j^*k^*}m$ and $U_{k^*m} = U_{k^*m} - X_{j^*k^*m}$; update the availability
Step 5: **if** $bj_{^*m} = 0$, **then** $v2(j^*) = 0$
If $u_{k^*m} = 0$, **then** $v2(J + k^*) = 0$
Step 6: **if** $v2(J + k) = 0$, $\forall\, k \in K$, **output** $x_{jkm,}$
Else return step 2

Fig. 15. Decoding procedure for 2nd stage

Table 4. Trace table of 2nd stage decoding procedure

Iteration	V(i+j)	B	U	J	K	X_{kj}
0	[756124|983]	*(78,157,59,118,108,216)*	*(200,200,350)*	5	1	108
1	[756104|983]	*(78, 157, 59, 118, 0,216)*	*(92,200,350)*	3	1	59
2	[750104|983]	*(78, 157, 0, 118, 0,216)*	*(33,200,350)*	1	1	33
3	[750104|083]	*(45, 157, 0, 118, 0,216)*	*(0,200,350)*	2	2	157
4	[700104|083]	*(45, 0, 0, 118, 0,216)*	*(0,43,350)*	4	2	43
5	[700104|003]	*(45, 0, 0, 75, 0,216)*	*(0, 0,350)*	1	3	45
6	[000104|003]	*(0,0,0,0,75,0,206)*	*(0, 0,305)*	6	3	216
7	[000100|003]	*(0,0,0,0,75,0,0)*	*(0, 0,89)*	4	3	75
8	[000000|003]	*(0,0,0,0,0,0,0)*	*(0, 0,14)*			

4. Computational results

The numerical experiment was used to demonstrate the effectiveness and efficiency of the approach of Simulated Annealing (SA) and Simulated Annealing Hybrid with priority algorithm (priSA) to solve the problem (m-rLNP). In this part, we will show the performance comparisons between our approach (SA) and (SA) based on the priority (priSA) and (priGA) proposed approach Lee JE, Gen M, Rhee KG [1]. the results of calculation are given in table 5, 6 and 7 .The algorithms are executed with c ++ on Intel (R) Pentium (R) M @ 2.40GHz CPU2020, 4.00GA RAM. Stage 1 represents cost of product transportation from returning center to disassembly center. Stage 2 represents cost of disassembled part 'a' and 'b' transported from disassembly center to processing center.

Table 5. Calculation results with the neighborhood search for stage 1 with SA, priSA

	neighborhood search	**SA**	**priSA**
Stage 1	Mutation 2-opt	3290	3130
	Mutation by insertion	3130	3080
	Mutation inverse	3130	3080

Table 6. Calculation results with the neighborhood search for stage 2 with SA, priSA

	neighborhood search	**SA**	**priSA**
Stage 2	Mutation 2-opt	9301	9419
	Mutation by insertion	9301	9301
	Mutation inverse	9301	9231

Table 7. Computational results with SA and priSA

Problem N:	Stage :	PriGA	***SA***	***PriSA***
1	Stage 1	3140	***Cost***	***Cost***
			3130	3080
	Stage 1	9427	9301	9231

To evaluate the results, we used two performance measures: speed of research and quality of provided solutions. The final results of the calculations are given in the Table above.

First we examined the effectiveness of the priority algorithm, as shown in Table 8, all the costs obtained with the simulated annealing hybrid algorithm with priority algorithm (priSA) are better at all costs obtained with the simulated annealing algorithm (SA), although the calculation time (priSA) or worse (SA), because the decoding process used (priSA) is more complex. This means that we can always get the best solution using the simulated annealing hybrid algorithm with priority algorithm, in other words (priSA) has better stability (SA).

Similarly, after the comparison of results found with the genetic algorithm based on priority (priGA), we said that the algorithm of Simulated Annealing with and without hybridization of the priority algorithm is more efficient in terms of costs obtained. This indicates the success of the proposed methods for the problem studied (m-rLNP).

In general, we can conclude that the Simulated annealing algorithm with and without priority algorithm is better to find better solutions than the hybrid genetic algorithm with priority algorithm in a reasonable calculation time.

5. Conclusions

We presented a logistics network design model multi-stages reverse (m-rLNP) proposed by Lee JE, Gen M, Rhee KG [1] to minimizes total transportation costs. The reverse logistics network includes studied areas of return products, disassembly areas and treatment centers with limited capabilities. Moreover the proposed model is able to find the amount of transport between those areas.

To solve this problem, a Simulated Annealing approach without and with the priority algorithm (SA)

and (priSA) was proposed with dynamic neighborhood search mechanisms are used to find the best solutions.

As continuity of this study we plan to expand soon by the following perspectives in order of chronology:

– The implementation of a heuristic for hybridization of the simulation and optimization.
– In the future, it is possible to study the performance of m-rLNP on large-scale problems also including real data.
– In this work, we consider a single product network with deterministic returned products; however, in many real cases, we have a multi-network product with the uncertainty in the statements. Therefore, given these assumptions may be a subject for future research.

References

[1] BARROS, A.I., DEKKER, R., SCHOLTEN, V., 1998. A two-level network for recycling sand: a case study. *European journal of operational research*, 110, 199–214.

[2] GEN, M., ALTIPARMAK, F., LIN, L. 2006 A genetic algorithm for two stage transportation problem using priority-based encoding. *OR Spectrum* 28, 337–354

[3] JAYARAMAN, V., GUIDE, V.D.R.J, SRIVASTAVA, R., 1999. A closed loop logistics model for remanufacturing. *Journal of the Operational Research Society*, 50, 497–508.

[4] JAYARAMAN, V., PATTERSON, R.A., ROLLAND, E., 2003. The design of reverse distribution networks: models and solution procedures. *European journal of operational research*, 150, 128–149.

[5] KIM, K., SONG, I., KIM, J., JEONG, B., 2006. Supply planning model for remanufacturing system in reverse logistics environment. *Computers & Industrial Engineering*, 51, 279–87.

[6] KIRKPATRICK, S., GELATT, C. D., VECCHI, M. P., 1983. Optimization by simulated annealing. science, 220(4598), 671-680.

[7] KIRKKE, H.R., HARTEN, A.V., SCHUUR, P.C., 1999. Business case Oce: reverse logistic network redesign for copiers. *OR Spectrum*, 21, 381–409.

[8] KO, H.J., EVANS, G.W., 2007. A genetic algorithm-based heuristic for the dynamic integrated forward/reverse logistics network for 3PLs. *Computer Operational Research,* 34, 346–66.

[9] LEWCZUK, K., 2015. The concept of genetic programming in organizing internal transport processes. *Archives of Transport*, 34(2), 61-74.

[10] LEE, J.E, GEN. M., RHEE, K.G., 2008. Network model and optimization of reverse logistics by hybrid genetic algorithm. *Computers and Industrial Engineering*, 56, 951–64.

[11] SOTO, J.P., RAMALHINHO-LOURENÇO, H., 2002 A Recoverable Production Planning Model (July 2002). UPF Economics and Business Working Paper No. 636. Available at SSRN: https://ssrn.com/abstract=394302 or htt://dx.doi.org/10.2139/ssrn.394302

[12] KARKULA, M., 2014. Selected aspects of simulation modelling of internal transport processes performed at logistics facilities. *Archives of Transport*, 30(2), 43-56

[13] MARTIN, P., GUIDE JR., V.D.R., CRAIGHEAD, C.W., 2010. Supply chain sourcing in remanufacturing operations: an empirical investigation to remake versus buy. *Decision Sciences*, 41(2), 301–321.

[14] ILGIN, M. A., GUPTA, S. M. 2010. Environmentally conscious manufacturing and product recovery (ECMPRO): A review of the state of the art. *Journal of environmental management*, 91(3), 563-591.

[15] METROPOLIS, N., ROSENBLUTH, A. W., ROSENBLUTH, M. N., TELLER, A. H., TELLER, E. 1953. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6), 1087-1092.

[16] MIN, H., KO, H.J., PARK, B.I., 2005. A Lagrangian relaxation heuristic for solving the multi-echelon, multi-commodity, closed-loop supply chain network design problem. *International Journal Logistics System Management*, 1, 382–404.

[17] MIN, H., KO, H.J., KO, C.S., 2006. A genetic algorithm approach to developing the multiechelon reverse logistics network for product returns. *Omega*, 34, 56–69.

[18] ŻOCHOWSKA, R., SOCZÓWKA, P., 2018. Analysis of selected transportation network structures based on graph measures. Scientific *Journal of Silesian University of Technology. Series Transport*, 98, 223-233.