IEMS Vol. 8, No. 1, pp. 14-21, March 2009.

Hybrid Priority-based Genetic Algorithm for Multi-stage Reverse Logistics Network

Jeong-Eun Lee†
Graduate School of Information, Production and Systems Waseda University
Kitakyushu, 808-0135 Japan, E-mail: leeje@toki.waseda.jp

Mitsuo Gen
Graduate School of Information, Production and Systems Waseda University
Kitakyushu, 808-0135 Japan, E-amil: gen@waseda.jp

Kyong-Gu Rhee
College of Business Administration Dongeui University
Busan, 614-714 Korea, E-mail: rhee@deu.ac.kr

Received January 22 2008/Accepted September 14 2008 (Selected from APIEMS 2007)

Abstract. We formulate a mathematical model of remanufacturing system as multi-stage reverse Logistics Network Problem (mrLNP) with minimizing the total costs for reverse logistics shipping cost and inventory holding cost at disassembly centers and processing centers over finite planning horizons. For solving this problem, in the 1st and the 2nd stages, we propose a Genetic Algorithm (GA) with priority-based encoding method combined with a new crossover operator called as Weight Mapping Crossover (WMX). A heuristic approach is applied in the 3rd stage where parts are transported from some processing centers to one manufacturer. Computer simulations show the effectiveness and efficiency of our approach. In numerical experiments, the results of the proposed method are better than pnGA (Prüfer number-based GA).

Keywords: Multi-stage Reverse Logistics Network Problem (mrLNP), Genetic Algorithm (GA), Priority-based Encoding Method, Weight Mapping Crossover (WMX).

1. INTRODUCTION

Many companies try to recover the residual value of their used products through remanufacturing process for environmental concerns and saving costs. Product remanufacturing such as transforming used items into marketable products through refurbishment, repair and upgrading can also yield substantial cost benefits (2007). Therefore, until now, most of strategies of companies were focused on processes from making products to marketing. But, according as changing of the life cycle of products including a process from collection of used products to remanufacturing, companies have to edit the strategies with adjustment themselves to this condition.

Reverse logistics is defined by REVLOG as "the propose of planning, implementing and controlling flows of raw materials, in process inventory, and finished goods, from the point of use back to point recovery or point of proper disposal" (2004). In a broader sense, reverse logistics refers to the distribution activities involved in product returns, source reduction, conservation, recycling, substitution, reuse, disposal, refurbishment, repair and remanufacturing (1992).

Researches related to the reverse logistics are conducting a lot of researches on various fields and subjects. In reuse logistics models, Kroon *et al.* (1995), they reported a case study concerning the design of a logistics system for reusable transportation packages. The authors proposed a mIP (mixed integer programming), closely related to a classical un-capacitated warehouse location model. Spengler *et al.* (1997) proposed a mIP model which based on the modified multi-level warehouse location problem. The model was solved by a modified Benders decomposition.

In recycling models, Pati *et al.* (2006), they developed an approach based on a mixed integer goal programming model to solve the problem. The model studies the inter-relationship between multiple objectives of a recycled paper distribution network. This study proposed reverse logistics network of remanufacturing

† : Corresponding Author

pro-cess. In remanufacturing models, Kim *et al.* (2006), discussed a notion of remanufacturing systems in reverse logistics environment. They proposed a general framework in view of supply planning and developed a ma-thematical model to optimize the supply planning function.

Lee *et al.* (2008) proposed the multistage reverse Logistics Network Problem (mrLNP) to minimize the total cost which involves reverse logistics shipping cost and fixed cost of opening the disassembly centers and processing centers.

Recently, GAs have received considerable attention regarding their potential as a novel approach to optimization problems and is often used to solve many real world problems, including the effective approaches on the SPR problem, capacity and flow assignment, and the dynamic routing problem. It is noted that all of these problem can be formulated as some sort of a combinatorial optimization problem.

In this paper, we consider a complex reverse logistics problem (rLP) including time period and inventory developing Lee *et al.* (2008). Furthermore, in disassembly process, the products are disassembled to several different parts. In the practical reverse logistics problems, the different parts should be assigned different processing processes based on the processing competence of the processing center. For this reason, the challenges of our study are both in the mathematical formulation and effective approach construction. In additional, multistage reverse logistics network problem (mrLNP) with minimum total reverse logistic shipping cost and inventory holding cost at disassembly center and processing center over finite planning horizons has been considered and new genetic algorithm approach is proposed. And considering the priority-based encoding method, we propose a new crossover operator called Weight Mapping Crossover (WMX), insertion mutation operator is adopted.

This paper is organized as follows: in Section 1, the problem is described in detail and the previous researches are reviewed; in Section 2, the mathematical model of the reverse logistics network is introduced; in Section 3, the priGA approach and heuristic approach are explained in order to solve this problem; in Section 4, numerical experiments are presented to demonstrate the efficiency of the proposed method; finally, in Section 5, concluding remarks are outlined.

2. MATHEMATICAL FORMULATION

In this Section, we consider constituents, variables, and assumptions for formulating a multistage logistics network model. In the remanufacturing process, after product recovery and disassembly part, the disassembled parts are sent to processing centers. We then consider if the state of these parts is the same as new products.

The mathematical models in this analysis have the following assumptions:

A1: We consider logistics network for treating single product.

A2: We consider the inventory factor at disassembly center and processing center over finite planning horizons.

A3: The demand of parts by manufacturer is known in advanced.

A4: The maximum capacities about four echelons are known: returning centers, disassembly centers, processing centers and manufacturer.

A5: If the number of provided parts from processing process is not enough for requirement of manufacturer, then manufacturer must

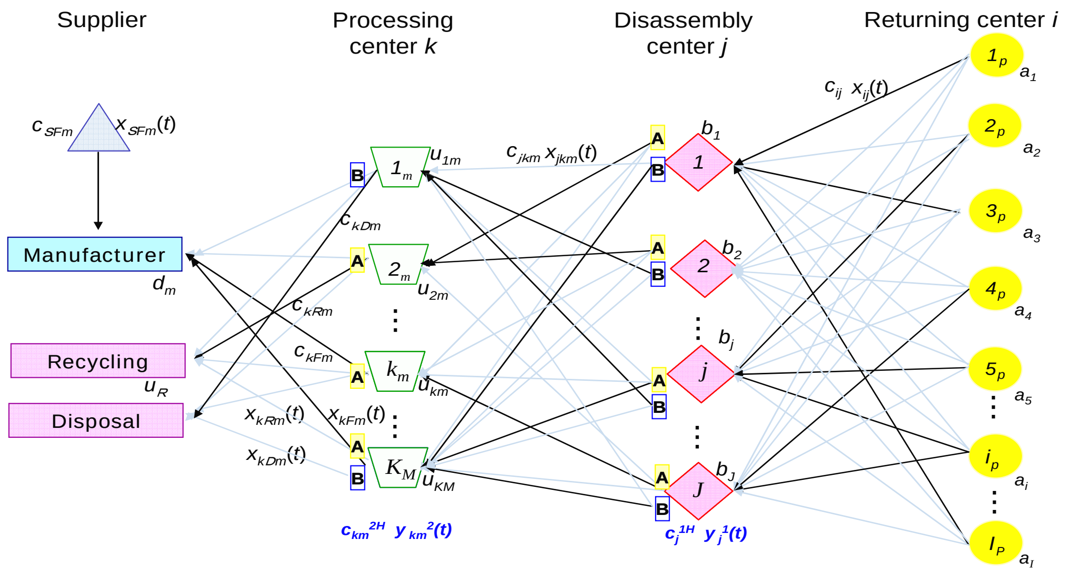

Figure 1. Multistage reverse logistics network model.

buy parts from supplier.

A6: If the number of provided parts from processing process exceeds in the requirement of ma-nufacturer, then exceeded capacities are distributed in order of recycling and disposal.

The notations are defined as follows:

Indices

i: index of returning center ($i = 1, 2, \cdots, I$)
j: index of disassembly center ($j = 1, 2, \cdots, J$)
k: index of processing center ($k = 1, 2, \cdots, K$)
m: index of part ($m = 1, 2, \cdots, M$)
t: index of time period ($t = 1, 2, \cdots, T$)

Parameters

I: number of returning centers
J: number of disassembly centers
K: number of processing centers
M: number of parts
T: planning horizons
a_i: capacity of returning center i
b_j: capacity of disassembly center j
u_{km}: capacity of processing center k for part m
u_R: capacity of recycling
u_D: capacity of disposal
d_m: demand of parts m in manufacturer F
n_m: the number of parts m from disassembling one unit of product
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 for part m
c_{kFm}: unit cost of transportation from processing center k to manufacturer F for part m
c_{kRm}: unit cost of transportation from processing center k to recycling R for part m
c_{kDm}: unit cost of transportation from processing center k to disposal D for part m
c_{SFm}: unit cost of transportation from supplier S to manufacturer F for part m
c_j^{1H}: unit holding cost of inventory per period at disassembly center j
c_{km}^{2H}: unit holding cost of inventory per period at processing center k for part m

Decision Variables

$x_{ij}(t)$: amount shipped from returning center i to disassembly center j in period t
$x_{jkm}(t)$: amount shipped from disassembly center j to processing center k in period t for part m
$x_{kFm}(t)$: amount shipped from processing center k to manufacturer F in period t for part m
$x_{kRm}(t)$: amount shipped from processing center k to recycling R in period t for part m
$x_{kDm}(t)$: amount shipped from processing center k to disposal D in period t for part m
$x_{SFm}(t)$: amount shipped from supplier S to manufacturer F in period t for part m
$y_j^1(t)$: inventory amount at disassembly center j in period t
$y_{km}^2(t)$: inventory amount at processing center k in period t for part m

The mathematical model of the problem is:

$$\begin{aligned}\min\ z = \sum_{t=0}^{T}\Bigg[& \sum_{i=1}^{I}\sum_{j=1}^{J} c_{ij}\, x_{ij}(t) + \sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{m=1}^{M} c_{jkm} x_{jkm}(t) \\ & + \sum_{k=1}^{K}\sum_{m=1}^{M} c_{kFm} x_{kFm}(t) - \sum_{k=1}^{K}\sum_{m=1}^{M} c_{kRm} x_{kRm}(t) \\ & + \sum_{k=1}^{K}\sum_{m=1}^{M} c_{kDm} x_{kDm}(t) + \sum_{m=1}^{M} c_{SFm} x_{SFm}(t) \\ & + \sum_{j=1}^{J} c_j^{1H} y_j^1(t) + \sum_{k=1}^{K}\sum_{m=1}^{M} c_{km}^{2H} y_{km}^2(t)\Bigg] \end{aligned} \tag{1}$$

$$\text{s.t. } x_{SFm}(t) + \sum_{k=1}^{K} x_{kFm}(t) \geq d_m, \quad \forall m, t \tag{2}$$

$$\sum_{j=1}^{J} x_{ij}(t) + y_j^1(t-1) \leq b_j, \quad \forall j, t \tag{3}$$

$$\sum_{k=1}^{K} x_{jkm}(t) + y_{km}^2(t-1) \leq u_{km}, \quad \forall k, m, t \tag{4}$$

$$x_{kRm}(t) \leq u_R, \forall k, m, t \tag{5}$$

$$x_{ij}(t), x_{jkm}(t), x_{kFm}(t), x_{kRm}(t), x_{kDm}(t), x_{SFm}(t) \geq 0 \tag{6}$$

The objective function (1) is to minimize total reverse logistics, *i.e.* shipping cost and inventory holding cost. Equation (2) shows the demand of parts. Equations (3), (4) and (5) are shows constraints about capacity of the disassembly center, processing center and recycling each other. Equation (6) shows the condition that all of variables are non-negative number.

3. HYBRID GENETIC ALGORITHM

3.1 Representation and Initialization

We here adopt the priority-based encoding method developed Gen *et al.* (2006). Although this encoding had been successfully applied on shortest path problem and project scheduling problem, the difference of our approach comes from the facts of special decoding and encoding procedures for transportation trees. The priority-based encoding method is an indirect approach. In this method, a gene in chromosome contains two kinds of information: the locus, the position of the gene within the structure of a chromosome, and the allele, the value the gene takes. The position of a gene is used to represent a node (source or depot), and the value is used to represent the priority of the node for constructing a tree among candidates.

For a transportation problem, a chromosome consists of priorities of sources and depots to obtain transportation tree and its length is equal to total number of sources m and depots n, *i.e.* m + n. The transportation tree corresponding with a given chromosome is generated by sequential arc appending between sources and depots. At each step, only one arc is added to tree selecting a source (depot) with the highest priority and connecting it to a depot (source) considering minimum cost.

For mrLNP, we use two priority-based encodings to represent the transportation trees on stages. This means that each chromosome in the population 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.

```
procedure 1.1 : 1st stage encoding
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 $j$
       $x_{ij}(t)$: the amount of shipment from $i$ to $j$
output: $v_1(i+j)$: chromosome, $\forall i \in I, \forall j \in J$
step 1 : priority $p \leftarrow (|I|+|J|)$, $v_1(i+j) \leftarrow 0$, $\forall i+j \in |I|+|J|$;
step 2 : $(i^*, j^*) \leftarrow$ argmin$\{c_{il} \mid x_{ij}(t) \neq 0$ & $(a_i = x_{ij}(t)\|, b_j = x_{ij}(t)\}$;
step 3 : $a_{i^*} = a_{i^*} - x_{i^*j^*}(t)$, $b_{j^*} = b_{j^*} - x_{i^*j^*}(t)$;
step 4 : if $a_{i^*} = 0$ then $v_1(i+j) \leftarrow p$, $p \leftarrow p-1$;
step 5 : if $(a_{i^*} = 0, \forall i^* \in I)$ & $(b_{j^*} = 0, \forall j^* \in J)$ then goto step 6;
         else return step 1;
step 6 : for $l = 1$ to $p$ do
         $v_1[i+j] \leftarrow l$, t=random$[1, (|I|+|J|)]$& $v_1[i+j] = 0$;
step 7 : output encoding $v_1[i+j]$ , $\forall$ t $\in |I|+|J|$
```

Figure 2. Encoding procedure for 1st stage of the chromo-
some.

```
procedure 1.2 : 1st stage decoding
input: $I$ : number of returning centers
       $J$ : number of disassembly centers
       $a_i$: capacity of returning center $i$, $\forall i \in I$ ,
       $b_j(t)=b_j - y_j^1(t-1)$ : capacity of disassembly center $j$ in period $t$, $\forall j \in J$ ,
       $c_{ij}$: shipping cost of one unit product from $i$ to $j$
       $v_1(i+j)$: chromosome, $\forall i \in I, \forall j \in J$ ,
output: $x_{ij}(t)$: the amount of shipment from $i$ to $j$
step 0: $x_{ij}(t) \leftarrow 0, \forall i \in I, \forall j \in J$
step 1: $l \leftarrow$ argmax$\{v_1(t), t \in I+J\}$; select a node
step 2: if $l \in I$, then $i \leftarrow l$; select a returning center
            $j^* \leftarrow$ argmin$\{c_{il} \mid v_1(j) \neq 0, j \in J\}$; select a $j$ with lowest cost
        else $j \leftarrow l$: select a disassembly process
            $i^* \leftarrow$ argmin$\{c_{il} \mid v_1(i) \neq 0, i \in I\}$; select a $i$ with lowest cost
step 3: $x_{i^*j^*}(t) \leftarrow$ min$\{a_{i^*}, b_{j^*} - y_j^1(t-1)\}$; assign available amount of units
        update the availabilities on $i$ $(a_{i^*})$ and $j$ $(b_{j^*} - y_j^1(t-1))$
            $a_{i^*} = a_{i^*} - x_{i^*j^*}(t)$, and $b_{j^*} = (b_{j^*} - y_j^1(t-1)) - x_{i^*j^*}(t)$
step 4: if $a_{i^*} = 0$, then $v_1(i^*) \leftarrow 0$
        if $(b_{j^*} - y_j^1(t-1)) = 0$, then $v_1(I+j^*) \leftarrow 0$
step 5: if $v_1(I+j) = 0$ , $\forall j \in J$, output $x_{ij}(t)$
        else return step 1
```

Figure 3. Decoding procedure for 1st stage of the chromosome.

The decoding procedure of 2nd stage priority-based decoding and its trace table are given in Fig. 4. 2nd stage encoding method is the same with in procedure 1.1 of 1st stage encoding.

Fig. 5 gives an illustration of the 3rd stage of transportation between processing centers and a manufacturer. The 3rd stage has three kinds of cases. In Case 1, if the quantity of parts provided from the processing centers is not enough for requirement of the manufacturer, parts for insufficient demand should be bought from a supplier. In Case 2, if the quantity of provided parts from processing centers is the same requirement of manufacturer, the parts privided from the processing centers are distributed to the manufacturer. In Case 3, if the quantity of parts provided from processing centers exceeds the requirement of manufacturer, the exceeded parts should be recycled. When if the parts still remain, they should be discarded.

```
procedure 2.2 : 2nd stage decoding
input: $J$ : number of disassembly centers
       $K$ : number of processing centers
       $b_j$: capacity of disassembly center $j$, $\forall j \in J$ ,
       $e_j(t)$ : shipping amount of disassembly center $j$, $\forall j \in J$ , $e_j(t) = \sum_{j=1}^{J} x_{ij}(t) + y_{ij}^1(t-1)$
       $u_{k^*m}(t) = u_{k^*m} - y_{km}^2(t-1)$: shipping amount of processing center $k$ in period $t$ for part $m$, $\forall k \in K$ ,
       $c_{jkm}$: unit cost of transportation from $j$ to $k$ for part $m$
       $v_2(j+k)$: chromosome, $\forall j \in J, \forall k \in K$ ,
output: $x_{jkm}(t)$: the amount of shipment from $j$ to $k$ for part $m$
step 0: $x_{jkm}(t)$: $\leftarrow 0, \forall j \in J, \forall k \in K$
step 1: $l \leftarrow$ argmax$\{v_2(t), t \in J+K\}$; select a node
step 2: if $l \in J$, then $j \leftarrow l$; select a disassembly center a $k$ with lowest cost
        else $k \leftarrow l$: select a processing center
            $k^* \leftarrow$ argmin$\{c_{jkm} \mid v_2(k) \neq 0, k \in K\}$; select a $j$ with lowest cost
step 3: $x_{j^*k^*m}(t) \leftarrow$ min$\{e_j(t), u_{k^*m} + y_{km}^2(t-1)\}$; assign available amount of units
        update the availabilities on $j$ $(e_j(t))$ and $k(u_{k^*m} + y_{km}^2(t-1))$
            $b_{j^*} = (e_j(t)) - x_{j^*k^*m}(t)$ and $u_{k^*m} = (u_{k^*m} + y_{km}^2(t-1)) - x_{j^*k^*m}(t)$; update the availability
step 4: if $b_{j^*} = 0$, then $v_2(j^*) = 0$
        if $(e_j(t)) = 0$, then $v_2(J+k^*) = 0$
step 5: if $v_2(J+k) = 0$ , $\forall k \in K$, output $x_{jkm}(t)$,
        else return step 1
```

Figure 4. Decoding procedure for 2nd stage of the chromosome.

Fig. 6 gives overall produce of proposed method.

3.2 Genetic Operators

Genetic operators mimic the process of heredity of gens to dreate new orrspring at each generation. Using the different genetic operators has very large influence on GA performance(1993). Therefore it is important to exmined different genetic operators.

3.2.1 Crossover Operator

Crossover is the main genetic operator. It operates on two parents (chromosomes) at a time and generates offspring by combining both chromosomes' features. The crossover is done to explore new solution space and crossover operator corresponds to exchanging parts of strings between selected parents.

Several crossover operators have been proposed for permutation representation, such as partially mapping crossover (PMX), order crossover (OX), position-based crossover (PX), cycle crossover (CX), Heuristic crossover, and so on. In this study, weight mapping crossover (WMX) operator is used.

WMX can be viewed as an extension of one-cut point crossover for permutation representation. As one-point crossover, two chromosomes (parents) would be

```
procedure 3: 3rd stage of transportation between processing  center and Manufacturer
input: $K$ : number of processing centers,
        $e_{km}(t)$: shipping amount of processing center $k$ for part $m$, $e_{km}(t) = \sum_{k=1}^{K} x_{jkm}(t) + y_{km}^{2}(t-1)$
        $u_{k*m}(t)$: shipping amount of processing center $k$ in period $t$ , for part $m$, $\forall k \in K$ ,
        $u_R$: capacity of recycling, $u_D$: capacity of disposal, $d_m$ : demand of parts $m$ in $M$
        $c_{kFm}$ : unit cost of transportation from processing center $k$ to manufacturer $F$ for part $m$, $\forall m$
        $c_{kRm}$ : unit cost of transportation from processing center $k$ to recycling $R$ for part $m$, $\forall m$
        $c_{kDm}$: unit cost of transportation from processing center $k$ to disposal $D$ for part $m$, $\forall m$
        $c_{SFm}$: unit cost of transportation from supplier $S$ to manufacturer $F$ for part $m$, $\forall m$
        $x_{jkm}$: the amount of shipment from $j$ to $k$ for part $m$ , $\forall m$
output: $x_{kFm}(t)$: amount shipped from $k$ to $M$ for part $m$,   $x_{kRm}(t)$: amount shipped from $k$ to $R$ for part $m$
        $x_{kDm}(t)$: amount shipped from $k$ to $D$ for part $m$,   $x_{SFm}(t)$: amount shipped from $S$ to $M$ for part $m$
step 0:  Calculate total shipment from parts $m$.
        $e_{km}(t) = \sum_{k=1}^{K} x_{jkm}(t) + y_{km}^{2}(t-1)$
step 1: for each part $m$, considering the follows cases:
    case 1: $d_m > e_{km}(x)$ then goto step 2, case 2: $d_m = e_{km}(x)$ then goto step 3, case 3: $d_m < e_{km}(x)$ then goto step 4.
step 2 : if $d_m > e_{km}(x)$, distribute to the manufacturer provided parts $e_{km}(x)$ from $k$,
        then $d_m - e_{km}(x) = x_{SFm}(t)$ , buy $x_{SFm}(t)$ from supplier.
step 3: if $d_m = e_{km}(x)$ distribute to the manufacturer provided parts $e_{km}(x)$ from $k$.
step 4: if $d_m < e_{km}(x)$ distribute to the manufacturer with low cost among provided parts $e_{km}(x)$ from processing center.
        then if remaining parts $e_{km}(x) - d_m \leq u_R$, distribute to the recycling parts $e_{km}(x) - d_m$,
        then if $e_{km}(x) - d_m > u_R$, distribute to the recycling parts $u_R$ and $(e_{km}(x) - d_m) - u_R - u_D$ distribute to the disposal.
step 5: output $x_{kFm}(t)$, $x_{kRm}(t)$, $x_{kDm}(t)$, $x_{SFm}(t)$
```

Figure 5. Decoding procedure for 3rd stage.

```
procedure 4: Overall procedure
step 0: $\bar{z} \leftarrow 0,\ z_t \leftarrow 0,\ y_j^1(t) \leftarrow 0,\ y_{km}^2(t) \leftarrow 0, \forall t$ ;
step 1: for all $t$
        find [$x_{ij}(t)$] by procedure 1
        find [$x_{jkm}(t)$] by procedure 2
        find [$x_{kFm}(t)$] [$x_{kRm}(t)$] [$x_{kDm}(t)$] [$x_{SFm}(t)$] by procedure 3
step 2:calculate and output the total cost for all $t$:
:
```

$$z = \sum_{t=0}^{T}\Bigg[\sum_{i=1}^{I}\sum_{j=1}^{J} c_{ij}x_{ij}(t) + \sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{m=1}^{M} c_{jkm}x_{jkm}(t) + \sum_{k=1}^{K}\sum_{m=1}^{M} c_{kFm}x_{kFm}(t) - \sum_{k=1}^{K}\sum_{m=1}^{M} c_{kRm}x_{kRm}(t) + \sum_{k=1}^{K}\sum_{m=1}^{M} c_{kDm}x_{kDm}(t) + \sum_{m=1}^{M} c_{SFm}x_{SFm}(t) + \sum_{j=1}^{J} c_j^{1H} y_j^1(t) + \sum_{k=1}^{K}\sum_{m=1}^{M} c_{km}^{2H} y_{km}^2(t)\Bigg]$$

output $\bar{z} = \frac{1}{T}\sum_{t=1}^{T} z_t$

Figure 6. Overall procedure.

chose a random cut-point and generate the offspring by using segment of own parent to the left of the cut-point, then remapping the right segment that base on the weight of other parent of right segment.

In this study, we proposed a new crossover operator, weight mapping crossover (WMX).

3.2.2 Mutation Operator

Mutation is a background operator which produces spontaneous random changes in various chromosomes. Similar to crossover, mutation is done to prevent the premature convergence and explores new solution space. However, unlike crossover, mutation is usually done by modifying gene within a chromosome. We also investigate the effects of two different mutation operators on the performance of GA. Insert mutation is used for this purpose. Several mutation operators have been proposed for permutation representation, such as inversion, insertion, displacement, and reciprocal exchange mutation.

In this study, insertion mutation has been adopted. Insertion mutation selects a gene at random and inserts it in a random position.

3.2.3 Evaluation and selection

Evaluation aims to associate each individual with a fitness value so that it can reflect the goodness of fit for an individual. This evaluation process intended to compare one individual with other individuals in the population. The choice of fitness function is also very critical because it has to accurately measure the desirability of the features described by the chromosome. The function should be computationally efficient since it is used many times to evaluate each and every solution. In the proposed algorithm, the fitness function has been taken as inverse of objective function.

The selection operator is intended to improve the average quality of the population by giving the high-quality chromosomes, *i.e.*, a better chance to get copied into the nest generation. The selection can be thought as the exploitation for the GA to guide the evolutionary process when we regard the genetic operation as the exploration for the search in solution space. We employ roulette wheel selection with elitist strategy as a selection mechanism.

In roulette wheel selection mechanism, the individuals on each generation are selected for survival into the next generation according to a probability value proportional to the ratio of individual fitness over total population fitness; this means that average quality of the population by giving the high-quality chromosomes a better chance to get copied into the next generation. Also the elitist method is employed to preserve the best chromosome for the next generation.

4. NUMERICAL EXPERIMENT

We used using pnGA (Prüfer number-based GA) proposed by Syarif and Gen (2003), to study the effectiveness of the developed GA with new encoding method (priGA).

As it is seen in Table 2, the number of returning centers changes between 4 and 30 on these problems, number of disassembly centers and number of processing centers change between 3 and 15, and 3 and 20, respectively. The data in test problems such as transportation costs, demand of parts, capacitates of returning centers, disassembly processes, processing processes, recycles and manufacturer were also randomly generated to provide realistic scenarios.

The parameters for the proposed GA approach are set as follows:

- Population size: *popSize* = 10 Mutation probability: p_M = 0.7
- Maximum generation: *maxGen* = 100 Crossover probability: p_C = 0.7

Table 3 gives computational results for the priGA and pnGA, on four test problems. Stage 1 represents cost of product transportation from returning center to disassembly center. Stage 2 represents cost of disassembled part transportation from disassembly center to processing center and holding cost of inventory in disassembly center. And stage 3 represents cost of processed part transportation from processing center to manufactuter, recyclig cost, disposal cost and holding cost of inventory in processing center.

In priGA, one-cut point crossover and insertion mutation operators were used as genetic operators and its rates were taken as 0.5. Each test problem is run by 10 times using GA approaches. As the results, priGA exhibits better performance than pnGA according to solution quality. This analysis indicates that the priGA is superior to the pnGA.

5. CONCLUSIONS

Table 3. Computational results with pnGA and priGA.

Problems No.	No. of constraints	No. of shipping variables	Stage	pnGA		priGA	
				average	ACT	average	ACT
1	**30**	**34**	stage 1	6990	0.13	6990	0.13
			stage 2	8470	0.12	8470	0.13
			stage 3	10020	-	10020	-
			total	**25480**	-	**25480**	-
2	**48**	**71**	stage 1	10790	0.18	10470	0.20
			stage 2	9940	0.19	9720	0.19
			stage 3	15100	-	15100	-
			total	**35830**	-	**35290**	-
3	**105**	**217**	stage 1	16930	0.43	15730	0.45
			stage 2	17110	0.40	16610	0.44
			stage 3	22010	-	22010	-
			total	**56050**	-	**54350**	-
4	**168**	**556**	stage 1	39265	0.84	37475	0.90
			stage 1	42260	0.77	40435	0.77
			stage 1	47535	-	47535	-
			total	**129060**	-	**125445**	-

ACT: Average computation time as second.

In this paper, we address reverse Logistics Network Problem for treating a remanufacturing problem which is one of the most important problems in the environment situation for the recovery of used products and materials.

We formulated a mathematical model for the rLNP by using priority-based genetic algorithm approach (priGA) and a heuristic approach. We also combined a new crossover operator, weight mapping crossover (WMX), with insertion mutation in hybrid priGA.

Numerical experiments demonstrated the efficiency and effectiveness of the hybrid GA approach for solving the mrLNP problem. Although memory requirement for new representation is greater than pnGA, *i.e.*, Prüfer number-based GA, only 2 digits for each stage in transportation problem, this representation shows very important two advantages in the real world applications. One of them is that its implementation is very easy. Another one is that after genetic operators, always feasible solutions are obtained. Based on this study, it was seen that the hybrid priGA with WMX demonstrated the best performance according to solution quality.

Limitations of this study are assumptions as follows. First of all, the scale of using just one product is

problem. And, in the case of numerical experiment, because amount of collected products and demand of parts are decided randomly, there are some doubtful points about result in the case of real data are applied.

In the future, it is possible to investigate the performance of the mrLNP on the large scale problems also including real-data.

ACKMOWLEDGMENTS

This work is partly supported by the Ministry of Education, Science and Culture, the Japanese Government: Grant-in-Aid for Scientific Research (No. 17510138, No.19700071, No. 20500143).

REFERENCES

Chung, S. L., Wee, H. M., and Yang, P. C. (2007), Optimal policy for a closed-loop supply chain inventory system with remanufacturing, *Mathematical and Computer Modeling*, in press.

REVLOG (2004), http://www.fbk.eur.nl/OZ/REVLOG/PROJECTS/TEMMINOLOGY/def-reverselogistics.htmt.

Stock, J. K. (1992), *Reverse logistics*, White Paper, Council of Logistics Management, Oak Brook, IL

Ko, H. J. and Evans, G. W. (2007), A genetic algorithm-

Table 2. The size of test problems.

Problem No.	Time Period (t)	Retuming Centers (I)	Disassembly Centers (J)	Processing Centers (K)	No. of Constraints	No. of shipping variables
1	3	4	3	3	30	34
2	3	6	5	5	48	71
3	3	15	10	12	105	217
4	3	30	15	20	168	556

based heuristic for the dynamic integrated forward/reverse logistics network for 3PLs, *Computers and Operations Research*, **34**(2), 346-366.

Lieckens, K. and Vandaele, N. (2007), Reverse logistics network design with stochastic lead times, *Computers and Operations Research*, **34**(2), 395-416.

Biehl, M., Prater, E., and Realff, M. J. (2007), Assessing performance and uncertainty in developing carpet reverse logistics systems, *Computers and Operations Research*, **34**(2), 443-463.

Kroon, L. and Vrijens, G. (1995), Returnable containers: an example of reverse logistics, *International journal of Physical Distribution and Logistics Management*, **25**(2), 56-68.

Spengler, T., Puchert, H., Penkuhn, T., and Rentx, O. (1997), Environmental integrated production and recycling management, *European Journal of Operational Research*, **97**(2), 308-326.

Pati, R. K., Vrat, P., and Kumar, P. (2008), A goal programming model for paper recycling system, *The International Journal of Management Science*, Omega, **36**(2), 405-417.

Kim, K. B., Song, I. S., and Jeong, B. J. (2006). Supply planning model for remanufacturing system in reverse logistics environment, *Computers and Industrial Engineering*, **51**(2), 279-287.

Lee, J. E., Gen, M., and Rhee, K. G. (2008), A multi-stage reverse logistics network problem by using hybrid priority-based genetic algorithm, *IEEJ Transactions on Electronics, Information and Systems*, **128**(3), 460-465.

Shimamoto, N., Hiramatsu, A., and Yamasaki, K. (1993), A Dynamic Routing Control Based on a Genetic Algorithm, *Proc. IEEE. Int. Conf. Neural Networks*, 1123-1128.

Syarilf, A. and Gen, M. (2003). Double Spanning Tree-based Genetic algorithm For Two Stage Transportation Problem, *International Journal of Knowledge-Based Intelligent Engineering System*, **7**(4), 388-389.

Gen, M. and Cheng, R. W. (2000), *Genetic Algorithm and Engineering Optimization*, Wiley, New York.

Gen, M., Altiparmak, F., and Lin, L.(2006), A genetic algorithm for two-stage transportation problem using priority-based encoding, *OR Spectrum*, **28**(3), 337-354.