Closed-Loop Supply Chain Design for Production System using Part and Module

Xing Chen
Department of Business Administration
Chosun University, Gwangju, South Korea
Tel: (+82)10-2607-4670, Email:chenxing1982@naver.com

Anudari Chuluunsukh
Department of Business Administration
Chosun University, Gwangju, South Korea
Tel: (+82)10-2959-1141, Email: anudr11@naver.com

YoungSu Yun†
Department of Business Administration
Chosun University, Gwangju, South Korea
Tel: (+82)62-230-6243, Email: ysyun@chosun.ac.kr (Coressponding author)

Mitsuo Gen
Department of Research, Fuzzy Logic Systems Institute, Japan
Research Institute of Science & Technology, Tokyo University of Science, Japan
Tel: (+81) 948-24-2771, Email: gen@flsi.or.jp

Abstract. Under the competitive economy environment, reducing the waste of resources and protecting the environment can provide companies with an additional revenue or cost reduction. So many companies concentrate their efforts on the design of closed-loop supply chain (CLSC) model. In previous literatures, many CLSC models were studied, but few of researchers mentioned modular production system and discussed the situation of disassembling of modules and parts together. In this paper, we proposed a new type of CLSC model. The proposed CLSC model considers the production system using part and module in forward logistics and the reselling activities in reverse logistics. For forward logistics, four kinds of part and two kinds of module are used for module manufacturer and product manufacturer, respectively. For reverse logistics, the returned products from customers are classified into two categories which are recoverable and unrecoverable products at recovery center. The recoverable products are resold at used market. The unrecoverable products are disassembled and then classified into recoverable modules, recoverable parts and unrecoverable parts. The recoverable modules and recoverable parts are sent to product manufacturer and module manufacturer. The unrecoverable parts are sent to waste disposal center. The proposed CLSC model is formulated using a nonlinear mixed integer programming and implemented using a hybrid genetic algorithm (HGA) approach. In numerical experiments, various scales of the proposed CLSC problem are presented to compare the performances of the hybrid genetic algorithm with other competing approaches. Experimental results prove that the hybrid genetic algorithm approach outperforms the others.

Keywords: closed-loop supply chain (CLSC), forward and reverse logistics, modular production system, hybrid genetic algorithm (HGA)

1. INTRODUCTION

For meeting internationalization of the market and the diversity of customer demand, flexibility in manufacturing activities becomes more and more important for manufacturers. A production system using part and module has been conceived to be very effective in producing product and meeting customer demand. Also, satisfying various environmental legislations is another consideration to manufacturers during their manufacturing activities. Therefore, in order to reduce the waste of resource and protect environment, many manufacturers have been developing various closed - loop supply chain (CLSC) models (Georgiadis *et al*. 2010; Wang *et al*., 2011; Zhang *et al*. 2011; Chung *et al*. 2011; Hamed *et al*. 2013; Demirel *et al*. 2014; Chen *et al*. 2015).

For the design of the CLSC model, Hamed *et al*. (2013) suggested the CLSC model with three handling processes of returned products, that is, the returned products are sent to secondary market, remanufacturer, and waste disposal center. Chen *et al*. (2015) proposed another type of CLSC model. In the model, they classified returned products into good-quality and poor-quality parts by considering their quality properties.

For production system using module, Wu *et al*. (2011) showed that module production system can increase the product life cycle. Fujita *et al*. (2013) proved that the module production system is able to improve the flexibility and productivity.

For environmental regulation in the CLSC model. Georgiadis *et al*. (2010) performed a case study on the role of recycle products with the waste electrical and electronic equipment (WEEE) in EU.

Many studies related to the CLSC model including the above mentioned studies have been performed using genetic algorithm (GA) approach (Zhang *et al*. 2011; Demirel *et al*. 2014; Chen *et al*. 2015). Zhang *et al*. (2011) studied a CLSC model with capacitated production plan and it was implemented using GA approach. Demirel *et al*. (2014) analyzed the situation of returned product in a CLSC model using GA.

However, although many researchers have studied the CLSC model using GA, GA has some weakness in it search process such as premature convergence or the absence of local search. Therefore, various hybrid GA (HGA) approaches using GA and some local search approaches have been developed (Gen and Cheng 1997, 2000; Lee *et al*. 2002; Yun *et al.* 2013). Recently, Kanagaraj *et al*. (2013) suggested a HGA approach using GA and Cuckoo search (CS). They used Levy flight scheme in CS to improve the search ability in GA.

Based on the above mentioned conventional studies, we propose a new type of the CLSC model using a HGA approach. The proposed CLSC model uses various part and module types for producing product in forward logistics (FL) and reselling activities in reverse logistics (RL). For the HGA approach, we develop an improved HGA (iHGA) approach using the conventional HGA approach by Kanagaraj *et al.* (2013).

In section 2, the proposed CLSC model is represented. The model is designed by a mathematical formulation in section 3. Section 4 shows the procedure of the iHGA approach to implement the mathematical formulation. The iHGA approach including some conventional approaches are applied to the three scales of the CLSC model presented in Section 5. Their performances are compared using various measures. Finally, some conclusions are summarized and the directions for future study are presented in Section 6.

2. STRUCTURE OF CLSC MODEL

Figure 1 shows the structure of the proposed CLSC model.

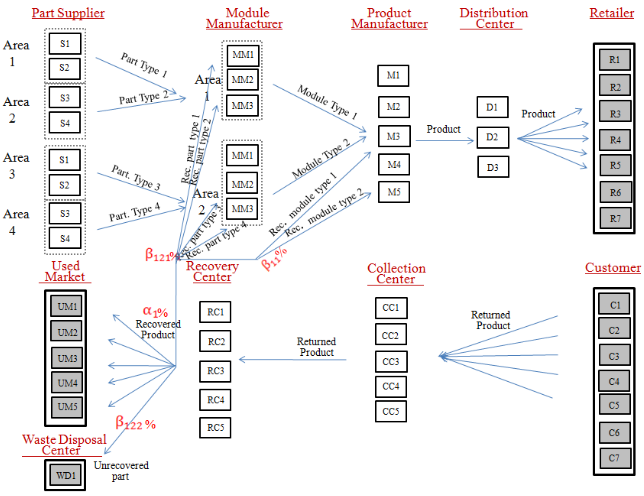

Figure 1: Structure of CLSC model

In Figure 1, the proposed CLSC model has five stages in FL and four stages in RL. For the five stages in FL, part suppliers in four areas, module manufacturers in two areas, product manufacturers, distribution centers and retailers are considered. For the four stages in RL, customers, collection centers, recovery centers, used markets and waste disposal centers are taken into consideration.

For the FL, four types of part are manufactured in part supplier and they are sent to module manufacturer so that two types of module are produced in it. The modules are sent to product manufacturer and then they are assembled for the production of products. Finally, the product are sent to retailer via distribution center.

For the RL, all returned products from customer are sent to recovery center through collection center. At recovery center, all returned products are classified into recoverable products with $\alpha_1\%$ and unrecoverable products with $\beta_1\%$ after testing and checking. The recovered products are resold at used market after some proper recovering process using the recoverable product at recovery center. However, the unrecoverable products are disassembled into recoverable modules with $\beta_{11}\%$ and unrecoverable modules with $\beta_{12}\%$. The recovered modules are sent to product manufacturer in FL after some proper handling process. The unrecoverable modules are disassembled into recoverable parts with $\beta_{121}\%$ and unrecoverable parts with $\beta_{122}\%$. The recovered parts are sent to module manufacturer after some proper handling process. The unrecoverable parts are sent to waste disposal center. Figure 2 shows the handling processes at recovery center.

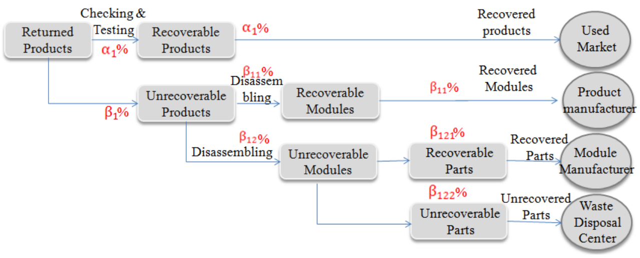

Figure 2: Handling processes at recovery center

3. MATHEMATICAL FORMULATION

In this section, a mathematical formulation is designed for the proposed CLSC model. First, some assumptions are considered as follows.

- Only single item is produced.
- The number of facility at each stage are already known.
- The numbers of retailer, customer, used market and waste disposal center are fixed and all of them are always opened.
- Only one facility at each stage should be opened, except for the stages of retailer, customer, used market and waste disposal center.
- Fixed costs of the facility which will be opened at each stage are different each other and already known.
- Unit handling costs of facilities at each stage are identical and already known.
- Unit transportation cost between each stage are different each other and already known.
- The handling capacity of facilities at a stage is the same or greater than that of facilities at the previous stage.
- The return rate of the returned products from customer is 100%.
- The qualities of recovered modules and parts at recovery center are identical with new ones.

Based on the assumptions mentioned above, a mathematical formulation for the proposed CLSC model is designed. First, index set, parameters, decision variables are defined as follow:

Index Set

a: index of area of part supplier; $a \in A$
b: index of area of module manufacturer; $b \in B$
h : index of part supplier; $h \in H$
i : index of module manufacturer; $i \in I$
j : index of product manufacturer; $j \in J$
k : index of distribution center; $k \in K$
l : index of retailer/customer; $l \in L$
m : index of collection center; $m \in M$
n : index of recovery center; $n \in N$
o : index of used market; $o \in O$
p : index of waste disposal center; $p \in P$

Parameters

$FIXSP_{ha}$: fixed cost of part supplier *h* at area *a*
$FIXMM_{ib}$: fixed cost of module manufacturer *i* at area *b*
$FIXPM_j$: fixed cost of product manufacturer *j*
$FIXDC_k$: fixed cost of distribution center *k*
$FIXCC_m$: fixed cost of collection center *m*
$FIXRM_n$: fixed cost of recovery center *n*
$UHSP_{ha}$: unit handling cost of part supplier *h* at area *a*
$UHMM_{ib}$: unit handling cost of module manufacturer *i* at area *b*
$UHPM_j$: unit handling cost of product manufacturer *j*
$UHDC_k$: unit handling cost of distribution center *k*
$UHCC_m$: unit handling cost of collection center *m*
$UHRM_n$: unit handling cost of recovery center *n*
$UTSMM_{hai}$: unit transportation cost from part supplier *h* at area *a* to module manufacturer *i*
$UTMMPM_{ibj}$: unit transportation cost from module manufacturer *i* at area *b* to product manufacturer center *j*
$UTPMD_{jk}$: unit transportation cost from product manufacturer *j* to distribution center *k*
$UTDR_{kl}$: unit transportation cost from distribution center *k* to retailer(customer) *l*
$UTRCC_{lm}$: unit transportation cost from retailer *l* to collection center *m*
$UTCCRC_{mn}$: unit transportation cost from collection center *m* to recovery center *n*
$UTRCMM_{nib}$: unit transportation cost from recovery center *n* to module manufacturer *i* at area *b*
$UTRCPM_{nj}$: unit transportation cost from recovery center *n* to product manufacturer *j*
$UTRCWD_{np}$: unit transportation cost from recovery center *n* to waste disposal center *p*
$UTRCSM_{no}$: unit transportation cost from recovery center *n* to used market *o*

Decision Variables

ks_{ha} : handling capacity of part supplier *h* at area *a*
kmm_{ib} : handling capacity of module manufacturer *i* at area *b*
kpm_j : handling capacity of product manufacturer *j*

kdc_k : handling capacity of distribution center *k*
kr_l : handling capacity of retailer *l*
kcc_m: handling capacity of collection center *m*
krc_n : handling capacity of recovery center *n*
ksm_o : handling capacity of used market *o*
kwd_p : handling capacity of waste disposal center *p*

$$ys_{ha} = \begin{cases} 1, \text{when part supplier } h \text{ at area } a \text{ is opened} \\ 0, \quad \text{otherwise} \end{cases}$$

$$ymm_{ib} = \begin{cases} 1, \text{when module supplier } i \text{ at area } b \text{ is opened} \\ 0, \text{otherwise} \end{cases}$$

$$ypm_j = \begin{cases} 1, \text{when product manufacturer } j \text{ is opened} \\ 0, \text{otherwise} \end{cases}$$

$$ydc_k = \begin{cases} 1, \text{when distribution center } j \text{ is opened} \\ 0, \text{otherwise} \end{cases}$$

$$ycc_m = \begin{cases} 1, \text{when collection center } m \text{ is opened} \\ 0, \text{otherwise} \end{cases}$$

$$yrc_n = \begin{cases} 1, \text{when recovery center } n \text{ is opened} \\ 0, \text{otherwise} \end{cases}$$

Objective Functions

Minimize *Total Cost (TC)* =

$$\begin{aligned}
&\sum_a \sum_h FIXSP_{ha} \cdot ys_{ha} + \sum_b \sum_i FIXMM_{ib} \cdot ymm_{ib} + \\
&\sum_j FIXPM_j \cdot ypm_j + \sum_k DC_k \cdot ydc_k + \\
&\sum_m FIXCC_m \cdot ycc_m + \sum_n FIXRC_n \cdot yrc_n + \\
&\sum_a \sum_h UHSP_{ha} + \sum_b \sum_i UHMM_{ib} \cdot kmm_{ib} \cdot ymm_{ib} + \\
&\sum_j UHPM_j \cdot kpm_j \cdot ypm_j + \sum_k UHDC_k \cdot kdc_k \cdot ydc_k + \\
&\sum_m UHCC_m \cdot kcc_m \cdot ycc_m + \sum_n UHRC_n \cdot krc_n \cdot yrc_n + \\
&\sum_a \sum_h \sum_i UTSMM_{iha} \cdot ks_{ha} \cdot ys_{ha} \cdot kmm_{ib} + \\
&\sum_b \sum_i \sum_j UTMMPM_{jib} \cdot kmm_{ib} \cdot ymm_{ib} \cdot ypm_j + \\
&\sum_j \sum_k UTPMD_{kj} \cdot kmm_j \cdot ymm_j \cdot ydc_k + \\
&\sum_k \sum_l UTDR_{lk} \cdot kdc_k \cdot ydc_k + \\
&\sum_l \sum_m UTRCC_{ml} \cdot kr_l \cdot ycc_m + \\
&\sum_m \sum_n UTCCRC_{nm} \cdot kcc_m \cdot ycc_m \cdot yrc_n + \\
&\sum_n \sum_i \sum_b UTRCMM_{bin} \cdot kmm_{ib} \cdot ymm_{ib} \cdot yrc_n + \\
&\sum_n \sum_j UTRCPM_{jn} \cdot krc_n \cdot ypm_j + \\
&\sum_n \sum_o UTRCSM_{on} \cdot ksm_o \cdot yrc_n + \\
&\sum_n \sum_o UTRCWD_{pn} \cdot kwd_p \cdot yrc_n
\end{aligned} \tag{1}$$

Subject to

$$\sum_{\mathbf{h}} \boldsymbol{ys_{ha}} = 1, \forall a \in A \tag{2}$$
$$\sum_i ymm_{ib} = 1, \forall b \in B \tag{3}$$
$$\sum_j ypm_j = 1 \tag{4}$$
$$\sum_k ydc_k = 1 \tag{5}$$
$$\sum_m ycc_m = 1 \tag{6}$$
$$\sum_n yrc_n = 1 \tag{7}$$
$$\sum_h (ks_{ha} ys_{ha}) - \sum_i (kmm_i \cdot ymm_{ib}) = 0, \ \forall a \in A \tag{8}$$
$$\sum_i (kmm_{ib} \cdot ymm_{ib}) - \sum_j (kpm_j \cdot ypm_j) = 0, \ \forall b \in B \tag{9}$$
$$\sum_i (kpm_j \cdot ypm_j) - \sum_k (kdc_k \cdot ydc_k) = 0 \tag{10}$$
$$\sum_k (kdc_k \cdot ydc_k) - \sum_l kr_l = 0 \tag{11}$$
$$\sum_n (krc_n - \sum_m (kcc_m \cdot ycc_m) = 0 \tag{12}$$
$$\sum_p ksm_o - \alpha_1 \% \sum_n (krc_n \cdot yrc_n) \geq 0 \tag{13}$$
$$\sum_j (kpm_j \cdot ypm_j) - \beta_{11} \% \sum_n (krc_n \cdot yrc_n) \geq 0 \tag{14}$$
$$\sum_i (kmm_{ib} \cdot ymm_{ib}) - \beta_{121} \% \sum_n (krc_n \cdot yrc_n) \geq 0 \tag{15}$$
$$\sum_a (kwd_p - \beta_{122} \% \sum_m (krc_n \cdot yrc_n) \geq 0 \tag{16}$$
$$\sum_n (krc_n \cdot ydc_k) - \sum_o ksm_o = 0 \tag{17}$$
$$ys_{ha} = \{0,1\}, \forall h \in H, \ a \in A \tag{18}$$
$$ymm_{ib} = \{0,1\}, \forall i \in I, \ b \in B \tag{19}$$
$$ypm_j = \{0,1\}, \forall j \in J \tag{20}$$
$$ydc_k = \{0,1\}, \forall k \in K \tag{21}$$
$$ycc_m = \{0,1\}, \ \forall m \in M \tag{22}$$
$$yrc_n = \{0,1\}, \forall n \in N \tag{23}$$
$$ks_{ha}, kmm_{ib}, kpm_j, kdc_k, kcc_m, krc_n, ksm_o, kwd_p \geq 0, \\ \forall h \in H, \ a \in A, \ \forall i \in I, \ b \in B, \ j \in J, \ \forall k \in K, \forall l \in L, \ \forall m \in M, \ \forall n \in N, \\ \forall o \in O, \ \forall p \in P \tag{24}$$

The above mathematical formulation is a nonlinear mixed integer programming. The equation (1) shows the objective function which total cost should be minimized. The total cost is consist of the sum of fixed costs, handling costs and transformation costs resulting from each stage. From equation (2) to (7) represent that only one facility at each stage can be opened. Equation (8) means that the each sum of parts at all part suppliers of areas 1, 2, 3, and 4 is the same as that of the modules at all module manufacturer of areas 1 and 2. Equation (9) to (12) have the same meaning with equation (8). Equation (13) to (17) restrict that the sum of the handling capability at each stage is the same or greater than that at the previous stage. From equation (18) to (23) shows that all facilities at each stage have 1 (opening) or 0 (closing). Equation (24) restricts that all variables have non negativity value.

4. PROPOSED METHODOLOGY

As shown in Section 1, GA has been proved to be very effective in locating global optimal solution to many CLSC models (Chen *et al.* 2015). Unfortunately, however, GA has some weakness such as premature convergence and the absence of local search. Therefore, various HGAs have been developed.

Recently, Kanagaraj *et al.* (2013) proposed a new type of HGA using GA and CS. They used Levy flight concept in CS in order to produce respective offspring in GA. The main logic of the HGA is to apply Levy flight scheme to a new individual from the population after GA search process. If the fitness value of the new individual is superior to that of the individual randomly selected from the population, then the new individual is included into the offspring for next generation. However, this process is done by only one time at every generation. Therefore, a weakness in this process may be happened, that is, if the fitness value of the new individual does not be superior to that of the individual randomly selected from the population, then the new individual is not included into the offspring for next generation.

For overcome this weakness of the search process of the HGA by Kanagaraj *et al*. (2013), the iHGA is proposed here. In the iHGA, Levy flight scheme is adapted to all individuals from the population after GA search process, which can increase the occurrence possibility of respective individuals of GA. The detailed implementation procedure is shown in Figure 3.

```
procedure iHGA approach
input problem data, parameters
begin
  Randomly generate initial parent population;
  while (not termination condition) do
     Create offspring population by two-point crossover
     operator and one-point mutation operator;
     for i = 1 to population size
        Select a new individual (new_i) from offspring
        population;
        Apply Levy flight to new_i ;
        Randomly select a new individual (old_i) from
        offspring population;
        if f(new_i) >= f(old_i) then
          Insert new_i into Offspring population;
        end
     end
     Sorting all individuals of offspring;
     A fraction (p_a) of worst individuals in offspring are
     abandoned;
     Regenerate new individuals randomly to replace the
     lost individuals;
  end
  output a best solution;
end;
```

Figure 3: Implementation procedure of iHGA

5. NUMERICAL EXPERIMENT

In numerical experiment, three scales of the proposed CLSC model are presented as shown in Table 1. Each scale has various numbers of facilities. As mentioned in assumption of Section 3, all facilities at retailer/customer (R), used market (UM) and waste disposal center (WD) are always opened. However, only one facility at part supplier (PS), module manufacturer (MM), product manufacturer (PM), distribution center (DC), collection center (CC), recovery center (RC) should be opened.

For each scale, the iHGA, the HGA by Kanagaraj *et al*. (2013), the GA by Gen and Cheng (1997) and Lingo by Lindo Systems (2015) are adapted. All approaches were run on IBM compatible PC3.4HGZ processor (Inter Core i7-3770 CPU), 8GB RAM and Window 7. The iHGA, HGA and GA except for Lingo were programmed using MATLAB 2015a.

In parameter setting, population size is 20, crossover rate is 0.5, mutation rate is 0.3, and total number of generation is 10,000. All approaches were run 30 trials for eliminate a randomness in their search processes. For various comparisons among all approaches, five measures of performance as shown in Table 2 are used.

Table 3 shows computation result for scale 1, 2 and 3. For scale 1, in terms of the best solution, the Lingo shows the best result and the GA has the worst one. The iHGA slightly outperforms the HGA. Similar results are also shown in terms of the average solution. In terms of the average iteration, the performance of the GA is the worst and that of the HGA is the best. The average search time to the pre-defined total number of generations shows the iHGA is the slowest and the HGA is the quickest.

For scale 2, in terms of the best solution, average solution and average iteration, the performances of the iHGA are superior to those of the others except for the Lingo. However, the search speed of the iHGA is almost three times slower than those of the GA and HGA. For scale 3, in terms of the best solution, the Lingo shows the best result, but in the comparison of the others, the performance of the iHGA is slightly better than those of the GA and HGA. A similar result is also shown in terms of the average solution, that is, the iHGA outperforms the GA and HGA. However, in terms of the average time, the search speed of the iHGA is significantly slower than those of the others.

Table 2: Measure of performance

Measure	Description
Best solution	The best value among the objective functions under satisfying all constraints.
Average solution	Averaged value of the objective functions under satisfying all constraints.
Average iteration	Averaged number of iteration after all trials.
Best setting	Opened number PS, MM, PM, DC, CC and RC when the best solution is obtained.
Average time	Average running time (in Sec.) after all trials.

Table 1: Three types of CLSC model

Scale	No. of PS	No. of MM	No. of PM	No. of DC	No. of R	No. of CC	No. of RC	No. of UM	No. of WD
	Area 1,2,3,4	Area 1,2							
1	8	8	12	8	20	12	8	20	4
2	10	10	14	10	25	14	10	25	5
3	20	20	28	20	50	20	20	50	10

Table 4 show the location decision result of GA, HGA, and iHGA for scale 3.

product manufacturer, 2,000 modules (=1,000 module 1 + 1,000 module scale 2) at module manufacturer of each area are required. Also, for producing 2,000 module at module manufacturer at each area, 4,000 parts (=1,000 part type 1 +1,000 part type 2 + 1,000 part type 3+ 1,000 part type 4) at part suppler of each area are required.

Table 3: Computation results of scale 1, 2 and 3

	Scale 1				Scale 2				Scale 3			
	GA	HGA	iHGA	Lingo	GA	HGA	iHGA	Lingo	GA	HGA	iHGA	Lingo
Best Solution	138,460	138,253	**138,177**	136,305	138,869	138,952	**138,724**	136,112	142,711	142,579	**142,450**	133,443
Average Iteration	5,547	3,941	**4,789**	-	5,832	5,267	**4,469**	-	4,986	4,754	**4,946**	-
Average Solution	139,081	139,039	**139,006**	-	139,839	139,866	**139,763**	-	143,337	143,368	**143,244**	-
Average Time	12.10	10.14	**32.74**	-	22.93	23.88	**58.30**	-	17.37	19.67	**58.27**	-

Table 4: Location decision result of GA, HGA and iHGA for scale 3

		PS No.				MM No.		PM No.	DC No.	CC No.	RC No.
	Best Solution	Area 1	Area 2	Area 3	Area 4	Area 1	Area 2				
GA	142,711	4	13	18	18	5	5	19	10	17	16
HGA	142,579	2	19	16	10	10	6	25	14	4	14
iHGA	142,450	18	3	9	14	20	3	11	13	3	9

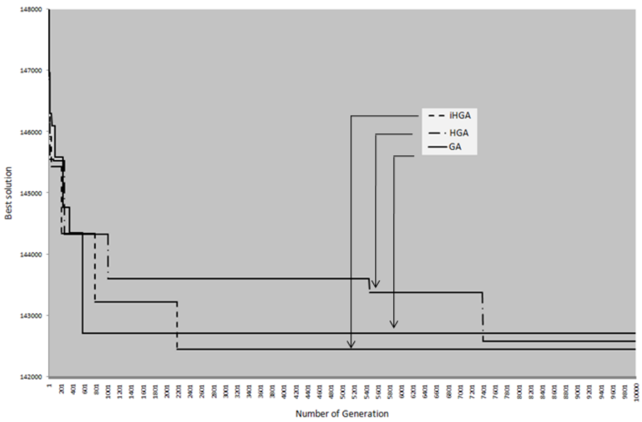

Figure 4: Convergence behaviors of GA, HGA and iHGA

Figure 4 shows that the convergence behaviors of GA, HGA, and iHGA for scale 3. The GA, HGA and iHGA are rapidly converging until about 103, 265 and 223 generations, respectively. After that, they all have slow convergence processes. This convergence behaviors implies that the search scheme used in the iHGA is more efficient that those of the GA and HGA.

Figure 5 shows the location decision result using the CLSL model for scale 3. We suppose that there are 1,000 products to be produced. For producing 1,000 product at

The 1,000 product at product manufacturer are then sent to all retailers through distribution center. The 1,000 returned products are sent to recovery center from customer and collection center. Of 1,000 returned products, 700 (=1,000 * 70%) recovered products are sent and resold at used market. 300(=1,000*30%) unrecovered products are disassembled into 300 (150=300*15% for module type 1, 150=300*15% for module type 2) recoverable modules and 300 (150=1000*15% for module type 1, 150=1000*15% for module type 2) unrecoverable modules. The 300 unrecoverable modules are also disassembled into 400(100 = 1000*10% for part type 1, 100 = 1000*10% for part type 2, 100 = 1000*10% for part type 3, 100 = 1000*10% for part type 4) recoverable parts, and 200 (50 = 1000*5% for part type 1, 50 = 1000*5% for part type 2, 50 = 1000*5% for part type 3, 50 = 1000*5% for part type 4) unrecoverable parts.

Based on the analyses using scale 1, 2, and 3, we can summarize the following some conclusions.

- In terms of the best and average solutions, although the Lingo used in a benchmark shows the best results, the performances of the iHGA is superior to those of the GA and HGA, which implies that the search scheme used in the iHGA is more efficient in locating best solution than those in the GA and HGA, since the iHGA uses a improve levy flight scheme rather than the HGA does and the GA does not use any hybrid scheme.

- In terms of the average time, the search speeds of the HGA is significantly quicker than that of the iHGA. This means that HGA uses Levy flight only one time at every generation, but the iHGA uses Levy flight as many as population size at every generation. Therefore, the former is able to search all possible search spaces more quickly than the latter.

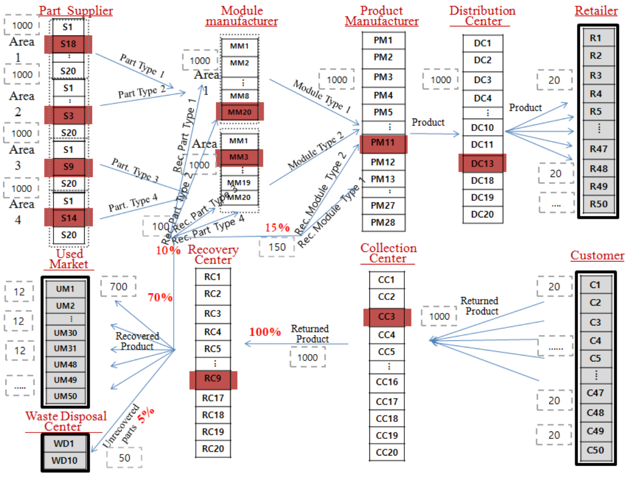

Figure 5: Location decision result using CLSC network for scale 3

6. CONCLUSIONS

In this paper, the CLSC model including forward and reverse logistics has been studied. Many researchers have suggested various types of the CLSL model. However, few of them focused on the production activity using part and module and the resell activities at used market. Therefore, this paper have proposed a new type of the CLSC model both with the production activity using part and module and with the resell activities at used market.

The proposed CLSC model have considered part suppliers and module manufacturer at various areas, product manufacturer, distribution center and retailer in forward logistics and customer, collection center, recovery center, used market and waste disposal center in reverse logistics. For representing the proposed CLSC model, a mathematical formulation is designed and it is implemented in iHGA approach. The iHGA approach is a revised version of the HGA Kanagaraj *et al*. (2013).

In numerical experiments, three scales of the CLSC model have been presented for comparing the performances of the iHGA, HGA and GA approaches under various measures of performance. The experiment results have shown the following two points. Firstly, the performances of the iHGA are more efficient in locating best solution and average solution than those of the GA and HGA. Secondly, in terms of the search speed, the iHGA does not be efficient than the GA and HGA. Therefore, an effort to reduce the search speed will be required for the iHGA and larger-sized CLSC model will be presented to compare the performance of the iHGA with those of the others. Also, various hybrid approaches using GA and other local search such as particle swarm optimization, ant colony optimization and so on will be proposed.

ACKNOWLEDGEMENT

This work is partly supported JSPS: Grant-in-Aid for Scie ntific Research (C; No. 15K00357).

REFERENCES

Chen, T.T., Chen, F.T.S. and Chung, S.H. (2015) An integrated closed-loop supply chain model with location allocation problem and product recycling decisions. *International Journal of Production Research*, **53(10)**, 3120-3140.

Chung, W. H., Okudan, G. and Wysk. R.A. (2011) Modular design to optimize product life cycle metrics in a closed-loop supply chain. *Proceedings of the 2011 Industrial Engineering Research Conference*.

Demirel, N. Ozceylan, E., Paksoy, T. and Gokcen, H. (2014) A genetic algorithm approach for optimizing a closed-loop supply chain network with crisp and fuzzy objectives. *International Journal of Production Research*, **52(12)**, 3637-3664.

Fujita. K., Amaya, H. and Akai, R. (2013) Mathematical model for simultaneous design of module commonalization and supply chain configuration toward global product family. *Journal of Intelligent Manufacturing*, **24**, 991-1004.

Gen. M. and Cheng, R. (1997) *Genetic Algorithm and Engineering Design*, New York: John Wiley and Sons.

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

Georgiadis, P. and Athanasiou, E. (2010) The impact of two-product joint lifecycles on capacity planning of remanufacturing networks. *European Journal of Operational Research*, **202(2)**, 420-433.

Hamed, S., Mirmehdi, S.E. and Mohsen, A.S. (2014) Designing and Planning a multi-echelon multi-period multi-product closed-loop supply chain utilizing genetic algorithm. *International Journal of Advanced Manufacturing Technology*, **68**, 917-931.

Kanagaraj, G., Phonnambalam, S.G. and Jawahar, N. (2013) A hybrid cuckoo search and genetic algorithm for reliability-redundancy allocation problems. *Computers & Industrial Engineering*, **66**, 1115-1124.

Lee, Y., Yun, Y.S. and Gen, M. (2002) Reliability optimization design for complex systems by hybrid GA with fuzzy logic control and local search, *IEICE Transactions on Fundamentals*, E85-A(4), 880-891.

Lingo, (2015) Lindo Systems, (www.lindo.com.)

Wang, H.F. and Hsu, H.W. (2010) A closed-loop logistics model with a spanning tree based genetic algorithm. C*omputers and Operations Research*, **37(2)**, 376-389.

Yun, Y.S., Chung, H.S. and Moon, C.U. (2013) Hybrid genetic algorithm approach for precedence-constrained sequencing problem. *Computers & Industrial Engineering*, 65: 137-147.

Zhang, J., Liu, X. and Tu, Y.L. (2011) A capacitated production planning problem for closed-loop supply chain with remanufacturing. *The international journal of advanced manufacturing technology*, **54(5-8)**, 757-766.