Integrated Supply Chain Design Problem with Various Handling Processes

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

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

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

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. In this paper, an integrated supply chain (ISC) design problem with various handling processes is proposed. Both forward logistics (FL) and reverse logistics (RL) are considered for designing the ISC problem. For both logistics, part suppliers, module manufacturers, product manufacturers, distribution centers, retailers, customers, collection centers, recovery centers, secondary markets, and disposal centers are taken into consideration. For considering various handling processes, collection center in reverse logistics checks and tests the returned products collected from customers and then classifies them as recoverable products, recoverable modules, recoverable parts, and unrecoverable parts. Each product, module, and part handled at collection centers are sent to part suppliers and module manufacturers in forward logistics and to recovery centers and waste disposal centers in reverse logistics. A mathematical programming modeling is formulated for representing the ISC design problem and it is implemented in hybrid genetic algorithm (HGA) approach. Various comparisons using the proposed HGA approach and other competing approaches are performed in numerical experiments. The experimental results show the proposed HGA approach outperforms the others for solving the ISC design problem.

Keywords: integrated supply chain design (ISC), handling process, forward logistics (FL), reverse logistics (RL), hybrid genetic algorithm (HGA)

1. INTRODUCTION

Recently, the reduction of product life cycle, the reinforcement of environmental regulation, and the increase of customer requirement in manufacturing industry have caused manufacturers to effectively construct the integrated supply chain (ISC) which combines forward logistics (FL) with reverse logistics (RL). To construct the ISC design problem, various facilities such as suppliers, manufacturers, distribution centers, and retailers (or customers) in FL and collection centers, recovery (or remanufacturing, refurbishing) centers, secondary markets and waste disposal centers in RL are taken into consideration. Since the ISC design problem has larger complexity in its network structure than FL or RL design problem, locating global optimal solution in the former is more difficult than that in the latter. Therefore, many researchers have studied various types of the ISC design problems to locate their global optimal solutions effectively (Fleischmann *et al.* 2000; Georgiadis and Besiou, 2008; Wang and Hsu, 2010; Amin and Zhang, 2012, 2013; Chen et al. 2015).

Fleischmann et al. (2000) suggested an ISC design problem combining FL with RL. The suggested ISC design problem considers various facilities which are consisted of supply, production, distribution and customer for product production in FL and selection, reprocessing, redistribution, reuse and disposer market for returned product in RL. Similar to Fleischmann et al. (2000), Wang and Hsu (2010) also suggested an ISC design problem. They considered suppliers, manufacturers, distribution centers and customers in FL and recycler and landfill area in RL. In the ISC design problem, especially, the distribution center has two functions, that is, the first function is to distribute products from manufacturer toward customer and the second function is to collect the used products from customers and then send them to recycler. The recycler disassembles them into reusable and unusable materials. The reusable materials are sent to manufacturer for producing product, but the unusable materials are sent to landfill area for burying them.

The ISC design problem suggested by Amin and Zhang (2012) considered suppliers, manufacturers, distribution centers and retailer in FL and disassembly center, refurbishing center and disposal center in RL. In this ISC design problem, they used two types of parts in FL and RL for producing product. First type is to use the new parts from suppliers and the second type is to use the returned parts from refurbishing center. Chen, et al. (2015) proposed the ISC design problem with various handling processes in RL. For the various handling processes, recycling center in RL tests the returned products from customer and then classifies them into reusable and unusable products. The reusable products are sent to manufacturer in FL and the unusable products are disassembled into reusable and unusable materials. The reusable materials are sent to supplier in FL and the unusable materials are sent to waste disposal center in RL.

The above mentioned conventional studies show that the various handling processes in RL are required for effectively constructing the ISC design problem. This requirement can also satisfy the facing problem of many manufacturers such as the reduction of product life cycle, the reinforcement of environmental regulation, and the increase of customer requirement in manufacturing industry. Therefore, in this paper, we also propose an ISC design problem with various handling processes in RL. The main difference between our ISC design problem and the conventional ones is that the latter has reproduction activities in FL and waste disposal activities in RL using the returned products from customers (or retailers), but the former has an additional activity for reselling the recovered products in RL including all activities of the latter.

For our ISC design problem, part suppliers, module manufacturers, product manufacturers, distribution centers and retailers in FL and customers, collection centers, recovery centers, secondary markets and waste disposal centers in RL are taken into consideration. For the various handling processes, the returned products from customers are classified and disassembled into four types (recoverable products, unrecoverable products, recoverable modules and recoverable parts). For representing the proposed ISC design problem, a mathematical modeling is formulated and it is implemented in hybrid genetic algorithm (HGA) approach. Various comparisons using the proposed HGA approach and other competing approaches are performed in numerical experiments.

2. ISC DESIGN PROBLEM

For the proposed ISC design problem, the various facilities in FL and RL are used and the conceptual network model is described in Figure 1.

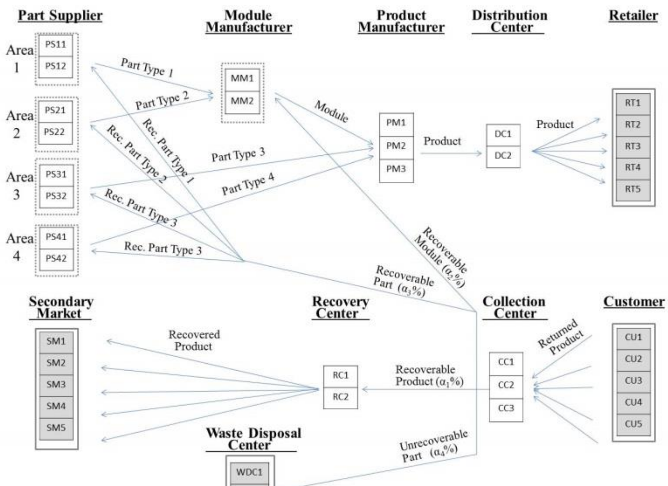

Figure 1: Conceptual network model for proposed ISC design problem

In Figure 1, for FL, part suppliers in areas 1 and 2 send part types 1 and 2 to module manufacturers and part suppliers in areas 3 and 4 send part types 3 and 4 to product manufacturers. Module manufacturers assemble modules using part types 1 and 2 and then send the module to product manufacturers. Product manufactures make products using the modules from module manufacturers and the part type 3 and 4 from part suppliers in areas 3 and 4. The products made at product manufacturers are sent to retailers through distribution centers. Finally, customers buy the products at retailers. For RL, the returned products from customers are collected at collection centers. They are tested and classified into recoverable and unrecoverable products. The recoverable products with α_1% are sent to recovery centers and then resell them at secondary markets after their all functions are recovered at recovery centers. The unrecoverable products are disassembled into recoverable and unrecoverable modules. The recoverable modules with α_2% are sent to module manufacturers. The unrecoverable modules are disassembled into recoverable and unrecoverable parts. The recoverable parts with α_3% are sent to module manufacturers and the unrecoverable parts with α_4% are sent to waste disposal centers. The all functions of collection centers are summarized in Figure 2.

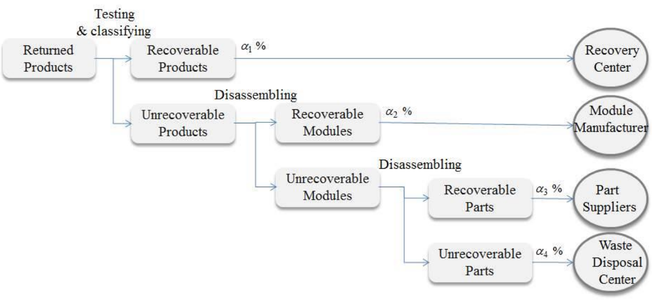

Figure 2: All functions of collection centers

3. MATHEMATAL FORMULATION

Before representing the proposed ISC design problem using mathematical formulation, some assumptions are considered as follows:

- Only single item production is considered.
- All numbers of part suppliers in each area, module manufacturers, product manufacturers, distribution centers, retailers, customers, collection centers, recovery centers, secondary markets, and waste disposal centers are already known. Of the facilities, all numbers of retailers, customers, secondary markets and waste disposal centers are always opened, whereas, part suppliers at each area, module manufacturers, product manufacturers, distribution centers, collection centers, and recovery centers should be opened alone at each stage.
- The fixed costs of all facilities considered at each stage are different each other and already known.
- The unit handling costs of all facilities considered are already known and they are identical with each other at the same stage.
- Unit transportation costs between each facility are different each other and already known.
- The handling capacity of each facility considered at same stage is the same or greater than that of the facility considered at the previous stage.
- Collection centers collect the returned products from all customers with 100% rate.
- All recoverable modules and parts sent from collection centers in RL have the same quality as the modules and parts from module manufactures and part suppliers in FL.

Mathematical formulation based on the assumptions mentioned above is developed. Firstly, index sets, parameters and decision variables are set.

Index sets:

a : index of area; $a \in A$

b : index of part supplier; $b \in B$

c : index of module manufacturer; $c \in C$

d : index of product manufacturer; $d \in D$

e : index of distribution center $e \in E$

f : index of retailer/customer; $f \in F$

g : index of collection center; $g \in G$

h : index of recovery center; $h \in H$

i : index of secondary market; $i \in I$

j : index of waste disposal center; $j \in J$

Parameters:

FSS_{ba} : fixed cost at part supplier b of area a

FMM_c : fixed cost at module manufacturer c

FPM_d : fixed cost at product manufacturer d

FDC_e : fixed cost at distribution center e

FCC_g : fixed cost at collection center

FRC_h : fixed cost at recovery center h

HSS : unit handing cost at part supplier

HMM : unit handing cost at module manufacturer

HPM : unit handing cost at product manufacturer

HDC : unit handing cost at distribution center

HCC : unit handing cost at collection center

HRC : unit handing cost at recovery center

TSM_{bac} : unit transportation cost from part supplier b of area a to module manufacturer c

TSP_{bad}: unit transportation cost from part supplier b of area a to product manufacturer d

TMP_{cd} : unit transportation cost from module manufacturer c to product manufacturer d

TPD_{de} : unit transportation cost from product manufacturer d to distribution center e

TDR_{ef} : unit transportation cost from distribution center e to retailer/customer f

TCC_{fg} : unit transportation cost from customer f to collection center g

TCM_{gc} : unit transportation cost from collection center g to module manufacturer c

TCS_{gba} : unit transportation cost from collection center g to part supplier b of area a

TCR_{gh} : unit transportation cost from collection center g to recovery center h

TCW_{gj} : unit transportation cost from collection center g to waste disposal center j

TRS_{hi} : unit transportation cost from recovery center h to secondary market i

Decision variables:

s_{ba} : handling capacity at part supplier b of area a

m_c : handling capacity at module manufacturer c

p_d : handling capacity at product manufacturer d

d_e : handling capacity at distribution center e

k_f : handling capacity at retailer/customer f

c_g : handling capacity at collection center g

r_h : handling capacity at recovery center h

s_i : handling capacity at secondary market i

w_j : handling capacity at waste disposal center j

$$xs_{ba} = \begin{cases} 1, & \text{if part supplier } b \text{ at area } a \text{ is opened} \\ 0, & \text{otherwise} \end{cases}$$

$$xm_c = \begin{cases} 1, & \text{if module manufacturer } c \text{ is opened} \\ 0, & \text{otherwise} \end{cases}$$

$$xp_d = \begin{cases} 1, & \text{if product manufacturer } d \text{ is opened} \\ 0, & \text{otherwise} \end{cases}$$

$$xd_e = \begin{cases} 1, & \text{if distribution center } e \text{ is opened} \\ 0, & \text{otherwise} \end{cases}$$

$$xc_g = \begin{cases} 1, & \text{if collection center } g \text{ is opened} \\ 0, & \text{otherwise} \end{cases}$$

$$xr_h = \begin{cases} 1, & \text{if recovery center } h \text{ is opened} \\ 0, & \text{otherwise} \end{cases}$$

Objective function is to minimize the total cost (*TC*) which is consisted of the total sum of the fixed costs at each facility (*TF*), the total sum of the handling costs at each facility (*TH*) and the total sum of the transportation costs between each facility (*TT*). To satisfy the objective function, various constraints are taken into consideration.

$$\text{minimize } TC = TF + TH + TT \tag{1}$$

$$TF = \sum_a \sum_b (FSS_{ba} \cdot xs_{ba}) + \sum_c (FMM_c \cdot xm_c) + \sum_d (FPM_d \cdot xp_d) + \sum_e (FDC_e \cdot xd_e) + \sum_g (FCC_g \cdot xc_g) + \sum_h (FRC_h \cdot xr_h) \tag{2}$$

$$TH = HSS \sum_a \sum_b (s_{ba} \cdot xs_{ba}) + HMM \sum_c (m_c \cdot xm_c) + HPM \sum_d (p_d \cdot xp_d) + HDC \sum_e (d_e \cdot xd_e) + HCC \sum_g (c_g \cdot xc_g) + HRC \sum_h (r_h \cdot xr_h) \tag{3}$$

$$\begin{aligned} TT = {} & \sum_a \sum_b \sum_c (TSM_{bac} \cdot s_{ba} \cdot xs_{ba} \cdot xm_c) + \sum_a \sum_b \sum_d (TSP_{bad} \cdot s_{ba} \cdot xs_{ba} \cdot xp_d) + \\ & \sum_c \sum_d (TMP_{cd} \cdot m_c \cdot xm_c \cdot xp_d) + \sum_d \sum_e (TPD_{de} \cdot p_d \cdot xp_d \cdot xd_e) + \\ & \sum_e \sum_f (TDR_{ef} \cdot d_e \cdot xd_e) + \sum_f \sum_g (TCC_{fg} \cdot c_g \cdot xc_g) + \\ & \sum_g \sum_c (TCM_{gc} \cdot c_g \cdot xc_g \cdot xm_c) + \sum_g \sum_b \sum_a (TCS_{gba} \cdot c_g \cdot xc_g \cdot xs_{ba}) + \\ & \sum_g \sum_h (TCR_{gh} \cdot c_g \cdot xc_g \cdot xr_h) + \sum_g \sum_j (TCW_{gj} \cdot c_g \cdot xc_g) + \\ & \sum_h \sum_i (TRS_{hi} \cdot r_h \cdot xr_h) \end{aligned} \tag{4}$$

subject to

$$\sum_b xs_{ba} = 1, \quad \forall a \in A \tag{5}$$

$$\sum_c xm_c = 1 \tag{6}$$

$$\sum_d xp_d = 1 \tag{7}$$

$$\sum_e xd_e = 1 \tag{8}$$

$$\sum_g xc_g = 1 \tag{9}$$

$$\sum_h xr_h = 1 \tag{10}$$

$$\sum_b (s_{ba} \cdot xp_{ba}) - \sum_c (m_c \cdot xm_c) = 0, \quad \forall a \in A \tag{11}$$

$$\sum_b (s_{ba} \cdot xp_{ba}) - \sum_d (p_d \cdot xp_d) = 0, \quad \forall a \in A \tag{12}$$

$$\sum_c (m_c \cdot xm_c) - \sum_d (p_d \cdot xp_d) = 0 \tag{13}$$

$$\sum_d (p_d \cdot xp_d) - \sum_e (d_e \cdot xd_e) = 0 \tag{14}$$

$$\sum_e (d_e \cdot xd_e) - \sum_f k_f = 0 \tag{15}$$

$$\sum_f k_f - \sum_g (c_g \cdot xc_g) = 0 \tag{16}$$

$$\sum_h (r_h \cdot xr_h) - \sum_i s_i = 0 \tag{17}$$

$$\sum_h (r_h \cdot xr_h) - \alpha_1 \sum_g (c_g \cdot xc_g) \geq 0 \tag{18}$$

$$\sum_c (m_c \cdot xm_c) - \alpha_2 \sum_g (c_g \cdot xc_g) \geq 0 \tag{19}$$

$$\sum_a \sum_b (s_{ba} \cdot xs_{ba}) - \alpha_3 \sum_g (c_g \cdot xc_g) \geq 0 \tag{20}$$

$$\sum_j w_j - \alpha_4 \sum_g (c_g \cdot xc_g) \geq 0 \tag{21}$$

$$xs_{ba} = \{0, 1\}, \quad \forall a \in A, b \in B \tag{22}$$

$$xm_c = \{0, 1\}, \quad \forall c \in C \tag{23}$$

$$xp_d = \{0, 1\}, \quad \forall d \in D \tag{24}$$

$$xd_e = \{0, 1\}, \quad \forall e \in E \tag{25}$$

$$xc_g = \{0, 1\}, \quad \forall g \in G \tag{26}$$

$$xr_h = \{0, 1\}, \quad \forall h \in H \tag{27}$$

$$s_{ba}, m_c, p_d, d_e, k_f, c_g, r_h, s_i, w_j \geq 0, \ \forall a \in A, b \in B, c \in C, \forall d \in D, \forall e \in E, \forall f \in F, \forall g \in G, \forall h \in H, \forall i \in I, \forall j \in J \tag{28}$$

Equation (1) represents the objective function which is consisted of *TF*, *TH* and *TT* in equations (2), (3) and (4). Equation (5) shows that only one part supplier in each area should be opened. As a same meaning, Equations (6) to (10) restrict the opening status of module manufacturers, product manufacturers, distribution centers, collection centers and recovery centers at each stage. Equation (11) shows the sum of handling capacity of the part supplier opened at each area is the same as that of the module manufacturer opened. Same meanings are described in Equations (12) to (17). Equation (18) means that the sum of handling capacity of the recovery center opened is always greater than that of handling capacity with α_1% of the collection center opened. Equations (19) to (21) also restrict the sum of handling capacities of the module manufacturer, part supplier, and waste disposal center. Equations (22) to (27) show the opening/closing decision at each stage. Equation (28) indicates that all parameters should have non-negative values.

4. HYBRID GENETIC ALGORITHM APPROACH

The mathematical formulation designed in section 3 is implemented in the proposed HGA approach. The proposed HGA approach combines conventional GA approach with a revised cuckoo search (CS) approach. Original HGA approach using GA and CS approach was proposed by Kanagaraj et al. (2013). The main idea of the original HGA approach is to apply Levy flight to only one individual of the offspring resulting from GA operators (selection, crossover and mutation). If the fitness value of the one individual after applying Levy flight is more effective than that of the individual ramdomly seleted in the offpsring, then the former is inserted into new population for the next generation. Unfortunately, however, this trial is done by only one time at each geneation, which may reduce a possibility for continuously producing respective individuals.

Threfore, in this paper, we propose a new type of HGA combining conventional GA approach with a revised (CS) approach. The detailed implementing procedures are as follows:

Step 1: GA approach
  Step 1.1: Initial population
    Randomly generate initial population *P*(*t*).
  Step 1.2: GA operator
    Elitist selection scheme in enlarged sampling space (Gen and Cheng 1997), one-point crossover (1X) operator (Michalewicz, 1994), and random mutation operator (Yun et al. 2013) are used for selection, crossover and mutation operators, respectively. After adapting GA operator, an offspring *O*(*t*) is produced.
Step 2: Revised CS approach
  Step 2.1: Levy flight
    Apply Levy flight to the one individual randomly selected in *O*(*t*) and calculate the fitness value of the individual F_{new}.
  Step 2.2: Updating the solution
    Compare the F_{new} with the fitness value (F_{old}) of the another one individual randomly selected in *O*(*t*). If the F_{new} is more efficient than the F_{old}, then the individual of F_{new} is inseted into new population for next generation.
  Step 2.3: Continuous updating the solution.
    Repeat Steps 2.1 and 2.2 as many as the number of the individuals of *O*(*t*).

Table 1: Three scales of the ISC design problem

Scale	Part Supplier 1	Part Supplier 2	Part Supplier 3	Part Supplier 4	Module Manufacturer	Product Manufacturer	Distribution Center	Retailer (=customer)	Collection Center	Recovery Center	Secondary Market	Waste Disposal Center
1	3	3	3	3	2	3	2	6	3	2	6	1
2	9	9	9	9	7	9	7	12	9	7	12	3
3	21	21	21	21	18	21	18	24	21	18	24	5

5. NEMERICAL EXPERIMENT

In numerical experiment, three scales of the ISC design problem are presented and they have various sizes of facility as shown in Table 1. For each scale, 3,000 products are produced in FL and handled in RL. The rates at collection center for handling the returned products from customer is as follows: $\alpha_1 = 60\%, \alpha_2 = 20\%, \alpha_3 = 10\%$ and $\alpha_4 = 10\%$, for recoverable products, recoverable modules, recoverable parts and unrecoverable parts, respectively.

For various comparisons, three conventional approaches are used and their brief descriptions are shown in Table 2. All approaches except for Lingo are programmed by MATLAB version 2014b and ran under a same computation environment (IBM compatible PC 1.3 Ghz processor-Intel core I5-1600 CPU, 4GB RAM, OS-X EI). The parameter settings for GA, HGA and pro-HGA are as follows: total numbers of generations is 2,000, 5,000 and 10,000 for the scales 1, 2, and 3, respectively, population size 30, crossover rate 0.5, and mutation rate 0.2. Total 30 trials are independently done for eliminating the randomness in the search processes of the GA, HGA and pro-HGA.

The performances of each approach are compared using various measures of performance as shown in Table 3. In Table 3, the values of all measures, except for best setting, are obtained after total 30 independent trails.

Table 2: Approaches for comparison

Approach	Description
GA	Conventional GA
HGA	HGA by Kanagaraj et al. (2013)
pro-HGA	Proposed HGA in this paper
Lingo	Nonlinear programming solver by Lindo Systems (2015)

Table 3: Measures of performance

Measure	Description
Best solution	Best solution among all solutions
Average solution	Average value of all solutions
Average iteration	Average number of generations
Average time	Average CPU time (unit: sec.)
Best setting	Facility number opened in each stage when best solution is located

For the three scales of the ISC design problem, the computation results using all approaches are shown in Table 4. In the scale 1 of Table 4, all the approaches including Lingo locate the same value (315333) in terms of the best solution. However, the pro-HGA is slightly efficient than the GA and HGA in terms of the average solution. The average time shows that the GA is the quickest and the pro-HGA the slowest. In the scale 2, the performance of the pro-HGA is superior to those of the others in terms of the best solution and superior to those of the GA and HGA in terms of the average solution. However, in terms of the average iteration and average time, the GA and HGA are more efficient than the pro-HGA.

Table 4: Computation result for each scale

	Scale 1				Scale 2				Scale 3			
	GA	HGA	pro-HGA	Lingo	GA	HGA	pro-HGA	Lingo	GA	HGA	pro-HGA	Lingo
Best Solution	**315333**	**315333**	**315333**	**315333**	**317430**	**316768**	**316019**	**318382**	**321298**	**321311**	**320146**	**322368**
Average solution	316893	316524	315333	-	320821	320515	318045	-	326006	325776	323015	-
Average iteration	12	23	23	-	41	33	61	-	452	57	985	-
Average Time	8.3	8.5	13.8	-	12.0	12.7	23.8	-	31.0	31.5	62.0	-

Table 5: Facility number opened at each stage in the scale 3

	Part Supplier 1	2	3	4	Module Manufacturer	Product Manufacturer	Distribution Center	Collection Center	Recovery Center
GA	18	3	14	18	13	4	11	4	18
HGA	17	8	17	21	17	20	13	18	11
pro-HGA	14	18	15	15	1	3	14	3	9
Lingo	8	19	14	14	8	21	15	15	13

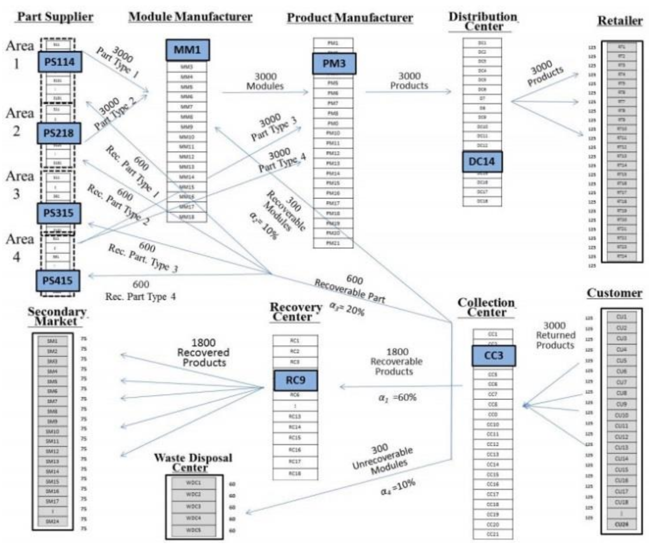

Figure 3: Opening/closing decisions at each stage using pro-HGA for scale 3

Similar results are also shown in the Scale 3, that is, the pro-HGA shows to be more efficient in terms of the best solution and average solution than the GA, HGA and Lingo, but in terms of the average iteration and average time, the former does not show any benefit than the latter.

Especially, for the scale 3, Table 5 and Figure 3 show the facility numbers opened at each stage in terms of the best setting. In Figure 3, the opened numbers (PS114, PS 218, PS315 and PS415) of part suppliers at each area mean that the part supplier numbers 14, 18, 15, and 15 at each area are opened, respectively.

For more detailed comparison, Figure 4 shows the convergence behaviors of the GA, HGA and pro-HGA until the number of generations is reached to 2,000. In Figure 4, all the approaches shows a rapid and various convergence behaviors at initial generations. Also the GA and HGA shows better convergence behaviors than the pro-HGA until generation number is reached to about 1,000. However, after about 1,000 generations, the GA and HGA do not show any convergence behavior, but the pro-HGA shows a better convergence behaviors.

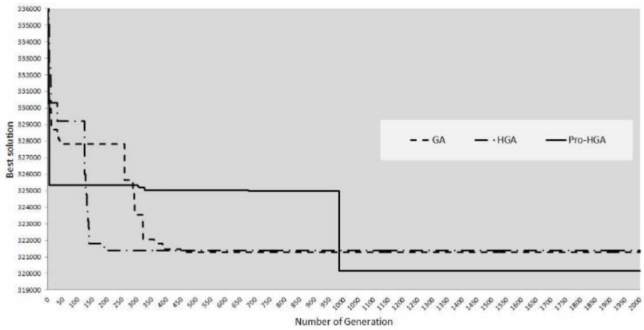

Figure 4: Convergence behaviors of GA, HGA and pro-HGA for scale 3

According to the analysis of computation results mentioned above, we can reach the following conclusion.

i) The pro-HGA is able to locate more efficient solutions than the GA and HGA for larger-scaled ISC design problems such as the scales 2 and 3.

ii) In the comparison between the HGA and pro-HGA, the latter outperforms the former, in spite of they all have hybrid search schemes using GA and CS. This implies that the search scheme of the pro-HGA is more efficient than that of the HGA. Especially, in the comparison between the pro-HGA and Lingo, the performance of the pro-HGA is superior to that of Lingo, which means that Lingo does not always guarantee the finding of optimal solution during it search process, although it is a well-known and commercial version for solving nonlinear programming.

iii) In the comparison of search time, the pro-HGA does not show any benefit when compared with the GA and HGA. This means that the pro-HGA has a reinforced search scheme using the revised CS approach in the Step 2 of Section 4 and the reinforced search scheme requires more times for locating optimal solution than that of the HGA.

6. CONCLUSION

In this paper, we have suggested an ISC design problem. The suggested ISC design problem has various handling processes in RL. For the various handling processes, collection center has multiple functions. Firstly, it tests the returned products from customers and then classifies them into recoverable and unrecoverable products.

The recoverable products are sent to recovery center and the unrecoverable products are disassembled into recoverable parts and modules and unrecoverable parts. Each part and module is sent to part supplier, module manufacturer and waste disposal center. Especially, the suggested ISC design problem has considered the reselling activity in secondary market using the recovered products from recovery center. This activity has not been considered in the conventional studies (Fleischmann *et al*. 2000; Georgiadis and Besiou, 2008; Wang and Hsu, 2010; Amin and Zhang, 2012, 2013; Chen *et al*. 2015).

The suggested ISC design problem has been represented by a mathematical formulation and implemented in the proposed HGA approach. In numerical experiments, three types of the ISC design problem have been presented to compare the performances of the proposed HGA approach and some other conventional approaches. The experimental results prove that the proposed HGA approach is more efficient in most of measures of performance than the other conventional approaches.

For our future study, larger-sized ISC design problems will be considered and more various approaches will be presented and their performances will be compared with that of the proposed HGA approach.

ACKNOWLEDGEMENT

This work is partly supported by JSPS: Grant-in-Aid f or Scientific Research (C; No. 15K00357).

REFERENCES

Amin, S. H. and Zhang, G. (2012) An integrated model for closed-loop supply chain configuration and supplier selection: Multi-objective approach, *Expert Systems with Applications*, **39**, 6782-6791.

Amin, S. H. and Zhang, G. (2013) A multi-objective facility location model for closed-loop supply chain network under uncertain demand and return, *Applied Mathematical Modelling*, **37**, 4165-4176.

Chen, T. T., Chan, 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.

Fleischmann, M., Krikke, H. R., Dekker, R. and Flapper, S. D. P. (2000) A characterization of logistics networks for product recovery, *Omega*, **28(6)**, 653-666.

Gen, M. and Cheng, R. (1997) *Genetic algorithms and engineering design*, John Wiley & Son, New York

Georgiadis P. and Besiou, M. (2008) Sustainability in electrical and electronic equipment closed-loop supply chains: A System Dynamics approach, *Journal of Cleaner Production*, **16**, 1665-1678.

Kanagaraj, G., Ponnambalam, S. G. and Jawahar, N. (2013) A Hybrid cuckoo search and genetic algorithm for reliability-redundancy allocation problems, *Computers & Industrial Engineering*, *66(4)*, 1115-1125.

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

Michalewicz, Z. (1994). *Genetic Algorithms + Data Structures = Evolution Program*, Spring-Verlag.

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

Yun, Y. S., Gen, M. and Hwang, R. K. (2013) Adaptive genetic algorithm to multi-stage reverse logistics network design for product resale, *Information: An International Interdisciplinary Journal*, **15(12)**, 6117-6138.