

Solving Facility Rearrangement Problem Using a Genetic Algorithm and a Heuristic Local Search

Atsushi Suzuki*

Takasaki City University of Economics, Takasaki, Japan

Hisashi Yamamoto

Tokyo Metropolitan University, Tokyo, Japan

(Received: February 15, 2012 / Revised: April 16, 2012 / Accepted: April 19, 2012)

ABSTRACT

In this paper, a procedure using a genetic algorithm (GA) and a heuristic local search (HLS) is proposed for solving facility rearrangement problem (FRP). FRP is a decision problem for stopping/running of facilities and integration of stopped facilities to running facilities to maximize the production capacity of running facilities under the cost constraint. FRP is formulated as an integer programming model for maximizing the total production capacity under the constraint of the total facility operating cost. In the cases of 90 percent of cost constraint and more than 20 facilities, the previous solving method was not effective. To find effective alternatives, this solving procedure using a GA and a HLS is developed. Stopping/running of facilities are searched by GA. The shifting the production operation of stopped facilities into running facilities is searched by HLS, and this local search is executed for one individual in this GA procedure. The effectiveness of the proposed procedure using a GA and HLS is demonstrated by numerical experiment.

Keywords: Facility Rearrangement Problem, Facility Planning, Genetic Algorithm, Local Search

* Corresponding Author, E-mail: asuzuki@tcue.ac.jp

1. INTRODUCTION

With changes in the economic and social environment, the volume of production is decreased adapting to the decreasing of demands. If the decrease is a temporary phenomenon, it will be adapted to decrease the production volume by modifying production scheduling and/or human resource assignment. If the modification cannot satisfy the cost constraint, some facilities are stopped and costs for those facilities are reduced. In this study, this problem is represented as a facility rearrangement problem (FRP).

In the area of the facility layout research, dynamic facility layout problems (DFLPs) were formulated and many solving methods have been proposed (Urban, 1993; Conway and Venkataramanan, 1994). In this approach, the number of operating facilities is given, and the material flow values between facilities are changed in each

period. Those approaches could not consider operating or stopping of facilities and shifting of operations at stopped facilities.

To consider these matters, the facility rearrangement problem model was built and evolutionary solving methods have been proposed by Suzuki *et al.* (2006, 2009). FRP is a problem for decision of stopped facilities to maximize production capacity of running facilities under the decreased cost constraint in order to adapt decreasing of the demand of products. FRP can be formulated into an Integer Programming model for maximizing of the total production capacity under the constraint of the total facility operating cost.

FRP can be divided into two sub-problems: the decision problem of stopped/running facilities and the integration problem of production operations in stopped facilities to running facilities. The integration problem can be dealt as a sub problem of the decision problem.

In a previous study (Suzuki *et al.*, 2009), this integration problem was solved by enumeration and the decision problem is solved by using an evolutionary procedure. In the cases of 90% of cost constraint and more than 20 facilities, the previous solving method was not effective. To solve such a problem, a two-step genetic algorithm (GA) was proposed (Suzuki and Yamamoto, 2010).

This paper proposes a solving procedure using a GA and a heuristic local search (HLS). Stopping/running of facilities is searched by GA. The shifting of production operation of stopped facilities into running facilities is searched by HLS, and this search is executed for one individual in this GA procedure. This procedure using GA and HLS can find more effective alternatives than the two-step GA (Suzuki and Yamamoto, 2010). The effectiveness of the proposed procedure using GA and HLS is demonstrated in a numerical experiment.

2. PROBLEM DEFINITION

2.1 Problem Assumption

In this study, the following problem is considered. In this period, a product is being manufactured by using n facilities. For the next period, the facility operating cost is reduced from managerial estimation. The cost constraint cannot be satisfied by modifying the operating scheduling and human resource assignment, and stopping of some facilities is considered. For the case of recovering of demands, production capacity should be maximized under the total cost constraint, as the maximum value of cost C_{\max} .

When a facility is stopped, the production operation in the facility is shifted into other running facility, and this shifting is called integration in this study. In the case of integration of facility i to facility j ($i \neq j$), the production efficiency decreases to $r(i, j)$ ($0 \leq r(i, j) \leq 1$). It means that the production process at facility i is optimized to the facility and the same efficiency will be not put out at other facility (facility j). Naturally, if facility j is similar to facility i in specifications and operation conditions, it may become $r(i, j) = 1$.

Under the assumption mentioned above, the conditions of running or stopping of facility should be decided to maximize the total production capacity.

2.2 Formulation

The objective function is an equation of total capacity of operating facilities and this should be maximized.

$$\max \sum_{j=1}^n p(j) x(j) \quad (1)$$

subject to

$$\sum_{j=1}^n (c_f(j) x(j) + c_v(j) p(j)) \leq C_{\max} \quad (2)$$

$$p(j) = \begin{cases} s(j), & \text{if } s(j) \leq q_{\max}(j) \\ q_{\max}(j), & \text{if } s(j) > q_{\max}(j) \end{cases}, j=1, \dots, n \quad (3)$$

$$s(j) = \sum_{i=1}^n q(i) r(i, j) y(i, j) + q(j) x(j), j=1, \dots, n \quad (4)$$

$$\sum_{j=1}^n y(i, j) + x(i) = 1, i=1, \dots, n \quad (5)$$

$$x(i) \in \{0, 1\}, i=1, \dots, n \quad (6)$$

$$y(i, j) \in \{0, 1\}, i, j=1, \dots, n, i \neq j \quad (7)$$

$$y(i, j) = 0, \text{ if } i = j; i, j=1, \dots, n \quad (8)$$

For $i=1, \dots, n, j=1, \dots, n,$

$c_f(j)$: fixed cost of facility j ,

$c_v(j)$: variable cost of facility j ,

$p(j)$: volume of production in facility j (after integration),

$q(i)$: volume of production in facility i (before integration),

$q_{\max}(j)$: maximum value of capacity of facility j ,

$r(i, j)$: decreasing ratio by integration from facility i to facility j .

For example, if facility i is integrated into facility j , the new production of facility j can be expressed in the following Eq.:

$$p(j) = \begin{cases} q(j) + q(i), & q_{\max}(j) - q(j) \geq q(i) / r(i, j) \\ q(j) + (q_{\max}(j) - q(j)) \times r(i, j), & \text{otherwise} \end{cases} \quad (9)$$

In this model, the condition of facility i (running or stopping) and shifting of production operation from facility i to facility j must be decided. For representation of condition of facility and integration of facilities, Eqs. (10) and (11) are used:

$$x(i) = \begin{cases} 1, & \text{facility } i \text{ is running} \\ 0, & \text{facility } i \text{ is stopped} \end{cases} \quad (10)$$

$$y(i, j) = \begin{cases} 1, & \text{facility } i \text{ is integrated to facility } j \\ 0, & \text{facility } i \text{ is not integrated to facility } j \end{cases} \quad (11)$$

If $x(i) = 1$, facility i is not integrated, and if $x(i) = 0$, facility i must be integrated to any one of other facilities, as the following Eq.:

$$\sum_{j=1}^n y(i, j) = \begin{cases} 0, & \text{if } x(i) = 1 \\ 1, & \text{if } x(i) = 0 \end{cases}, i=1, \dots, n \quad (12)$$

The cost reducing ratio R_c can be given by C_{\max} and current cost as Eqs. (13) and (14):

$$Rc = C_{\max} / C_{\text{init}} \quad (13)$$

$$C_{\text{init}} = \sum_{j=1}^n (c_f(j) + q(j)c_v(j)) \quad (14)$$

The cost of rearranged facilities is estimated by Eq. (15):

$$\text{Cost} = \sum_{j=1}^n (c_f(j)x(j) + c_v(j)p(j)) \quad (15)$$

In the cases of $n < 20$ and $Rc \geq 0.90$, excellent alternatives can be found by using evolutionary improvement procedure (Suzuki *et al.*, 2009). However, in the cases of $n \geq 20$ and $Rc < 0.90$, the improvement procedure cannot find feasible alternatives because the search area is expanded and the combination of integration alternatives are increased. To find excellent alternatives, new search methods are necessary. In this paper, a procedure using a GA and a HLS is proposed.

3. SOLVING PROCEDURE

3.1 Search Strategy

In section 2.2 of this paper, FRP can be represented as a 0-1 (binary) programming model. In this manner, simultaneous searching of $x(i)$ and $y(i, j)$ is not effective because searching of infeasible area is often happened.

In this study, this problem is dealt as searching of $x(i)$ and searching of $y(i, j)$ is dealt as sub problem under the one set of $x(i)$ (Figure 1).

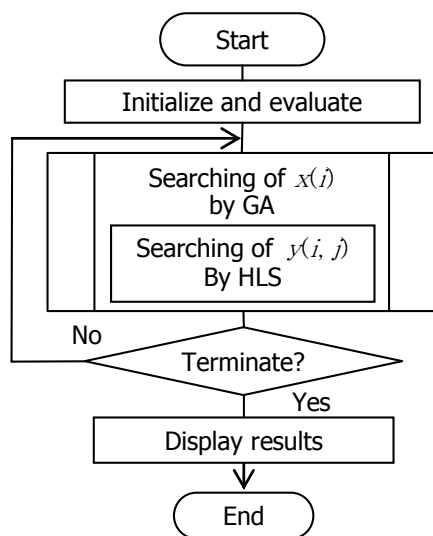


Figure 1. Outline of search procedure. GA: genetic algorithm, HLS: heuristic local search.

In the proposed procedure, $x(i)$ ($i=1, \dots, n$) are searched by GA and $y(i, j)$ ($i, j=1, \dots, n$) are searched

by HLS. The flowchart of GA procedure is shown in Figure 2.

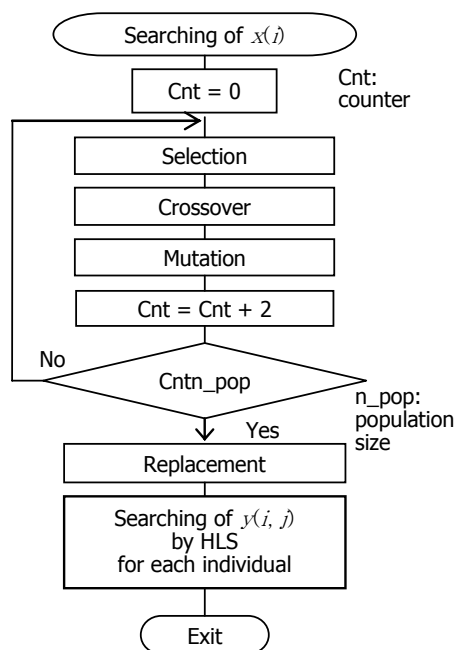


Figure 2. Procedure for searching of $x(i)$ by genetic algorithm. HLS: heuristic local search.

At the initialize step, n_{pop} (population size) individuals are generated. Each value of $x(i)$ is given by probability of $x(i)=1$ based on Rc value. Then values of $y(i, j)$ for each individual are generated based on $x(i)$ values of the individual to be a feasible integration alternative. Each individual is evaluated by objective function and cost values.

3.2 GA Procedure

For searching of $x(i)$, a simple GA is used in this paper. The GA procedure for searching $x(i)$ is shown in Figure 2 and the setting for this GA procedure is shown Table 1.

In the step of selection, two individuals are selected from current population with roulette selection based on the selection probability value of the individual. The fitness of Individual h is calculated by Eq. (16):

$$\text{Fitness}(h) = \frac{\text{OFV}(h) \times C_{\max} / \text{Cost}(h)}{\sum (\text{OFV}(h) \times C_{\max} / \text{Cost}(h))} \quad (16)$$

The selection probability of Individual h , $P_{\text{select}}(h)$ is estimated as Eq. (17):

$$P_{\text{select}}(h) = \frac{\text{Fitness}(h)}{\sum \text{Fitness}(h)} \quad (17)$$

Table 1. Setting for genetic algorithm procedure

Operations	Setting
Initialize	(Probability of $x(i) = 1$) = R_c
Selection	Roulette selection
Crossover	Single point
Mutation	Inverse 1 to 0 at any $x(i)$ on mutation probability If all $x(i) = 1$ then any $x(i) = 0$ If all $x(i) = 0$ then any $x(i) = 1$ If $\sum x(i) \leq C_{max} / C_{init} \times 0.5n$ then any $x(i) = 1$
Elitism preservation	The best one individual of current population is preserved into next population
Fitness	Fitness of individual h $= \frac{OFV(h) \times C_{max} / Cost(h)}{\sum (OFV(h) \times C_{max} / Cost(h))}$

In the step of crossover, a simple (single point) crossover is operated into two selected individuals as exchanging of two vectors of $x(i)$ cut at random point of $i = 1, \dots, n - 1$.

The mutation is performed as inverting 1 to 0 at any $x(i)$ on mutation probability. If all $x(i) = 0$ then any $x(i) = 1$. If all $x(i) = 1$ then $x(i) = 0$. As other operation, if $\sum x(i) \leq C_{max} / C_{init} \times n \times 0.5$ then any $x(i) = 1$.

Selection, crossover and mutation are repeated until the number of the generated new individuals (as children) = n_pop (population size).

The current population is replaced by new individuals. If elitism preservation is operated, the best individual(s) is adopted as the top of population of the next generation. In this paper, the best one individual is kept at next as the elitism preservation. The individuals of new population are evaluated for objective function value (OFV), cost, fitness and selection probability. This procedure is repeated until termination condition.

3.3 HLS Procedure

The HLS procedure is executed for searching of integration alternatives from the stopped facilities into any running facilities under restriction based on the values of $x(i)$. When $5 < n_0 < n - 3$ HLS is carried out ($n_0 = n - \sum_{i=1}^n x(i)$), in other cases enumeration is executed as Figure 3.

In the first step of the HLS procedure (Figure 4), one alternative of integration, $y(i, j)$ ($i, j = 1, \dots, n$) is randomly generated as $Y_{current}$ (a current alternative). In the next step, the neighbor Y_{nb} is generated. For example, values of $x(i)$ ($i = 1, \dots, 6$) are given as follows:

$$(x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6) = (1 \ 0 \ 1 \ 1 \ 0 \ 1)$$

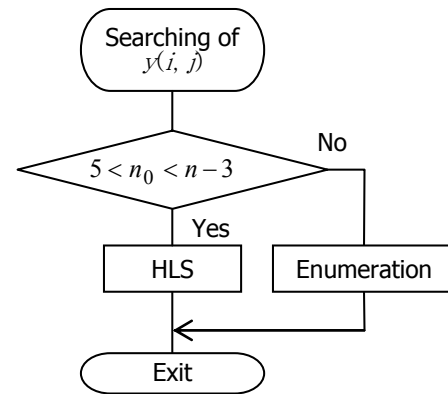


Figure 3. Procedure of searching of $y(i, j)$. HLS: heuristic local search.

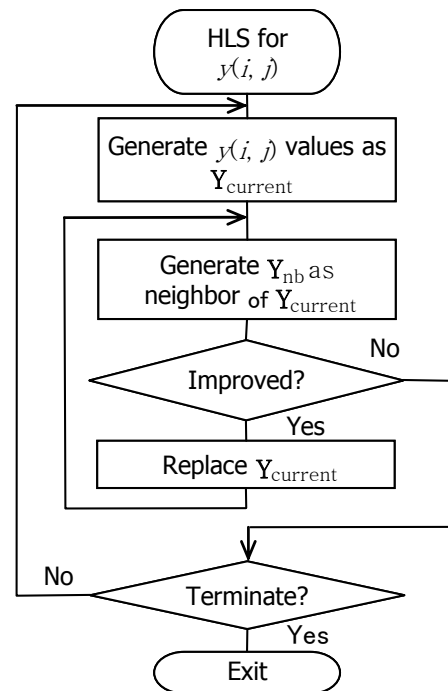


Figure 4. Heuristic local search (HLS) for $y(i, j)$.

These values means that facilities 2 and 5 are stopped and other facilities are running. It is assumed that the following one integration alternative is generated from this individual as a current alternative:

$$Y_{current} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

This $Y_{current}$ means that the production operations in facilities 2 and 5 are shifted into facilities 1 and 4, respectively. As one example of neighbor, the following

Y_{nb} can be generated:

$$Y_{nb} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

If this Y_{nb} improves $Y_{current}$, $Y_{current}$ is replaced by Y_{nb} and this operation is iterated. When it is not improved, the searching is restarted, and a new current alternative is generated; and such operation is iterated until termination condition. In this study, this iteration is performed as the maximum time until 1/100 of all calculation time.

4. NUMERICAL EXPERIMENT

4.1 Numerical Example

For demonstration of the proposed algorithm, a numerical experiment is executed. Problem data sets are used in Suzuki and Yamamoto (2010).

As experiment, 42 data sets of example problem, that is 7 problems \times 6 rates of cost decreasing ($Rc = C_{max} / C_{init}$), were solved by GA+HLS and the two-step GA (Suzuki and Yamamoto, 2010). The calculation by the first setting and the second setting was carried out by the same computation time for solving of one example problem at 3,600 seconds ($n = 20$), 3,960 seconds ($n = 22$), 4,320 seconds ($n = 24$), 4,680 seconds ($n = 26$), 5,040 seconds ($n = 28$). In an example problem, calculations are executed five times because the proposed procedure is one of probabilistic search. The parameters are set as crossover probability $Pc = 1.00$, mutation probability $Pm = 0.1$, population size $n_{pop} = 20$. These values are based on a previous study by Suzuki and Yamamoto (2010) and preliminary experiment. The termination condition is the computation time mentioned above.

4.2 Results of Experiment

The numerical experiment was executed in the following computational environment: Pentium4 CPU 3.60 GHz, 2.00 GB RAM, Microsoft Windows XP Pro SP2, BCC++ 5.5. The result of experiment is listed in Table 2.

In all cases, OFVs and cost values by GA+HLS are excellent to values by 2-step GA. In the cases of $Rc = 0.95$ and 0.90 , OFV and cost are similar values. In the cases of $Rc = 0.50, 0.60$, and 0.70 , OFVs by GA+HLS are very excellent than 2-step GA. It is demonstrated that the proposed method, GA+HLS, is superior to 2-step GA of the previous method.

Table 2. Comparison of by GA+HLS and 2-step GA (mean of OFVs and cost values)

n	Rc	GA+HLS		2-step GA	
		OFV	Cost	OFV	Cost
20	0.95	2120.0	37625.0	2120.0	37630.0
	0.90	2120.0	36539.0	2114.9	38830.5
	0.80	2117.6	35931.0	2095.9	34228.6
	0.70	2090.2	32441.9	2053.7	29072.0
	0.60	2063.6	27260.6	2008.3	25843.4
	0.50	1937.3	22900.7	1760.6	22639.0
22	0.95	2224.0	39815.0	2224.0	43905.0
	0.90	2224.0	39185.0	2222.5	42935.0
	0.80	2222.5	39412.5	2199.7	37057.0
	0.70	2205.4	35526.5	2156.8	32008.0
	0.60	2159.2	30741.5	2100.0	27340.0
	0.50	2070.8	25597.6	2020.2	24151.6
24	0.95	2496.0	46532.0	2496.0	49012.0
	0.90	2496.0	46585.0	2489.7	47682.0
	0.80	2491.5	44354.0	2458.2	42704.0
	0.70	2462.3	39635.0	2394.5	36284.6
	0.60	2416.0	33631.2	2357.2	31282.0
	0.50	2296.8	28544.2	2199.7	28086.8
26	0.95	2554.0	50007.0	2552.5	52240.0
	0.90	2554.0	50175.0	2540.0	50180.0
	0.80	2540.0	47556.0	2510.7	44162.0
	0.70	2506.0	41565.5	2453.5	38335.0
	0.60	2438.4	35726.8	2387.5	33254.6
	0.50	2370.6	29726.1	2279.6	27466.4
28	0.95	3041.0	55075.0	3041.0	59520.0
	0.90	3041.0	54785.0	3029.7	55141.0
	0.80	3027.4	51857.9	2985.8	49771.4
	0.70	2978.0	45374.7	2913.3	42396.0
	0.60	2903.6	39354.0	2814.8	36448.2
	0.50	2754.7	32977.2	2643.1	30501.2

GA: genetic algorithm, HLS: heuristic local search, OFV: objective function value.

5. CONCLUSION

This paper proposed a procedure using a genetic algorithm and a heuristic local search to solve the facilities rearrangement problem. The proposed procedure featured the generation method for initial population, a new estimation method of fitness, selection by roulette selection, simple crossover operation, and mutation by 0/1 inverting operation. For the search of integration of production operation in stopped facilities, the heuristic local search is used. The result of the numerical experi-

ment demonstrated the effectiveness of this proposed procedure.

As a future study, use of the searching of $y(i, j)$ other metaheuristics methods, for example, simulated annealing and tabu search method, should be studied. Furthermore, the relationship of problem data characteristics and the effectiveness of solving method should be also considered.

ACKNOWLEDGMENTS

This research was supported in part by the Grant-in-Aid for Scientific Research 23510174 from Japan Society of Promotion of Science (JSPS).

REFERENCES

- Urban, T. L. (1993), A heuristic for the dynamic facility layout problem, *IIE Transactions*, **25**, 57-63.
- Conway, D. G. and Venkataramanan, M. A. (1994), Genetic search and the dynamic facility layout problem, *Computers and Operations Research*, **21**, 955-960.
- Suzuki, A., Yamamoto, H., and Kainuma, Y. (2006), Facility integration problem for resource recycle, *Proceedings of International Workshop on Institutional View of SCM*, Tokyo, Japan, 329-334.
- Suzuki, A., Yamamoto, H., and Tsujimura, Y. (2009), Facility rearrangement problem and evolutionary algorithms under variable demands, *Proceedings of Meeting of Division C of the Institute of Electrical Engineers of Japan*, Tokushima, Japan, 1198-1203.
- Suzuki, A. and Yamamoto, H. (2009), Evolutionary algorithms for solving facility rearrangement problem, *Proceedings of 10th Asia Pacific Industrial Engineering and Management System Conference*, Kitakyushu, Japan, 1292-1296.
- Suzuki, A. and Yamamoto, H. (2010), A two-step genetic algorithm for solving facility rearrangement problem, *Proceedings of 11th Asia Pacific Industrial Engineering and Management System Conference*, Melaka, Malaysia.