MULTI-OBJECTIVE OPTIMIZATION FOR MIXED-INTEGER PROGRAMS THROUGH HYBRID GENETIC ALGORITHM

WITH VALUE FUNCTION MODELED BY NEURAL NETWORKS

Yoshiaki Shimizu

Toyohashi University of Technology
1-1 Hibarigaoka, Tenpaku-cho, Toyohashi-city, Aichi 441-8580, Japan
E-mail: shimizu@sandy.tutpse.tut.ac.jp

Abstract - With the aim of developing a flexible optimization method for managing conflict resolution, this paper concerns itself with multi-objective mixed-integer programs. For this purpose, we have proposed an intelligence supported approach that combines genetic algorithm with mathematical programs (hybrid genetic algorithm) to derive the best-compromise solution. Also we have developed a novel modeling method of value function using neural networks, and incorporated it into the approach which employs a simulated repair operation of DNA. As a result, we can provide a practical and effective method in which the hybrid strategy maintains its advantages of relying on good matches between the solution methods and the problem properties such as a genetic algorithm for unconstrained combinatorial optimization and a mathematical program for constrained continuous ones. Finally, by taking an example for site location problems of hazardous wastes, we have examined the effectiveness of the proposed approach numerically.

Keywords: Combinatorial Optimization, Hybrid Genetic Algorithm, Neural Networks

1 INTRODUCTION

As a goal for production systems at the next generation, concepts such as sustainable development, ecological economics and cleaner production are being commonly accepted these days. For the further progress of the society, it is very important to develop a reliable method that can reveal multiple attributes imbedded in such goal, and make a rational decision based on it. With such a point of view, we have paid special concerns to developing a practical solution method for multi-objective mixed-integer programs (MOMIP). This is because we can formulate a variety of problems that appears in production systems as MOMIP. Then to derive the Pareto optimal solution (POS) set of MOMIP, we propose a hierarchical approach that combines genetic algorithm (GA) with mathematical programming (MP), and call it hybrid GA (HybGA).

First, we present how HybGA can work to derive a best-compromise solution (BCS) on the POS set using a value function modeled by neural networks (NN). Then, we propose HybGA/MOR that introduces a new operation named repair operation into GA to improve the method. There, we weigh on practice rather than mathematical rigidity of the optimality so that we can cope with large-scale real world problems. Finally, taking a typical site location problem regarding hazardous waste disposal as an example, we will examine the effectiveness of the proposed method.

2 HYBRID GA FOR SOLVING MOMIP

2.1 Preliminary Consideration

We will consider about the following problem hereinafter.

$$\text{(p.1)}\quad \text{Minimize}\left\{f_1(\boldsymbol{x},\boldsymbol{z}),\quad f_2(\boldsymbol{x},\boldsymbol{z}),\quad \cdots,\quad f_N(\boldsymbol{x},\boldsymbol{z})\right\}$$

$$\textit{subject to}\begin{cases} g_i(\boldsymbol{x},\boldsymbol{z})\le 0\,, & (i=1,\ldots,I1)\\ h_i(\boldsymbol{x},\boldsymbol{z})=0\,, & (i=1,\ldots,I2)\\ \boldsymbol{x}\ge\boldsymbol{0}:\text{real}, \boldsymbol{z}\ge\boldsymbol{0}:\text{integer}\,, & \end{cases}$$

where the phrase "Minimize *N* objectives simultaneously" means that the solution of (p.1) needs articulate satisfaction levels by a decision maker (DM) among multiple objectives. We denote its *i-th* element by $f_i(\boldsymbol{x},\boldsymbol{z})$ whick is incommensurable and conflicts with some other objectives. As well as concern on multiple objectives, we should note the existence both of integer variables and real ones. This means the problem will refer to MOMIP.

Compared with many applications associated with MIP under single objective, only a few have been studied about MOMIP previously. On the other hand, solution of combinatorial problems like MIP, GA is popularly known as an efficient and practical method through a variety of applications. Additionally, the multi-start nature of the searching algorithm is suitable to cope with multi-objective problems where obtaining the POS set has an important meaning. Accordingly, several methods (Schaffer, 1985, Goldberg, 1989, Venugopal and Narendran, 1992, Fonseca and Fleming, 1993, Louis and Rawlings, 1993, Tamaki et al., 1995, Murata et al.,1995) have been developed to derive the POS set using GA. (We call them generically MOGA.)

However, we show it quite impractical to apply any of MOGA to MOMIP. Hence, to derive rationally the POS set to which BCS must belong, we propose the following hierarchical formulation (Shimizu, 1999a).

$$\text{(p.2)}\quad \underset{\boldsymbol{z}\ge\boldsymbol{0}:\,\text{integer}}{\text{Minimize}}\ f_p(\boldsymbol{x},\boldsymbol{z})$$

$$\textit{subject to}$$

$$\underset{\boldsymbol{x}\ge\boldsymbol{0}:\,\text{real}}{\text{Minimize}}\ f_p(\boldsymbol{x},\boldsymbol{z})$$

$$\textit{subject to}\begin{cases} f_i(\boldsymbol{x},\boldsymbol{z})\le f_i^*+\varepsilon_i\,, & (i=1,\ldots,N, i\ne p)\\ g_i(\boldsymbol{x},\boldsymbol{z})\le 0\,, & (i=1,\ldots,I1)\\ h_i(\boldsymbol{x},\boldsymbol{z})=0\,, & (i=1,\ldots,I2)\end{cases}$$

In the above, the slave problem refers to the usual ε−constrained problem (Cohen, 1978) possible to achieve the Pareto optimality even in the nonconvex case. Here, $f_p(\cdot)$ denotes a principal objective function, f_i^* optimal value of the *i-th* objective, and ε_i its amount of degradation. As an practical solution method, we propose a hierarchical scheme which applies GA to the master problem and MP to the slave problem (HybGA). By taking such a scheme, we can deal with the constrained optimization with respect to real variables by an appropriate MP, and with the unconstrained one with respect to integer variables by GA. Thus, we can keep the advantages in solution that may come from good matches of solution methods with the properties of problems. By solving the above problem for a variety of ε_i repeatedly, we can obtain the POS set. In fact, we verify its effectiveness compared with the conventional MOGA methods through numerical experiments.

However, since our final goal is to solve (p.1), DM must choose his/her BCS among the candidate solutions on the POS set through an appropriate tradeoff analysis. Eventually, such tradeoff analysis will refer to a process to adjust the attained level of each objective according to the DM's preference. In other words, we can obtain our BCS through finding out the most preferable amounts of degradation, that is, values of $(\boldsymbol{\varepsilon}_{-p}, f_p(\boldsymbol{x},\boldsymbol{z}))$. Here $\boldsymbol{\varepsilon}_{-p}$ denotes every ε_i except for the *p-th* element.

2.2 Modeling Method of Value Function

To rank the candidate solutions according to the DM's preference, two issues known as prior articulation and progressive one are applicable. The former takes a stage-wise process which separates the identifying stage of the value function from the optimizing stage, and the later fuses these, and proceeds decision interactively in the course of the searching stage. In the present case, prior articulation is essential for the numerous evaluations required by GA to solve the master problem. As a prior articulation for solving multi-objective optimization, utility theory (Keeney and Raiffa, 1976) has been extensively applied. In many technical applications, however, it has been rarely practical since we need examine certain mathematical conditions such like utility independence and/or preferential independence beforehand.

To avoid such tedious examinations, a new method using NN is proposed recently without going into the mathematical details (Malakooti and Zhou, 1994). The authors try to model the value function of DM, $V(\boldsymbol{f})$ by using a feedforward NN as follows. First, suppose utopia values $\boldsymbol{f}^{\mathrm{U}}$ and nadir values $\boldsymbol{f}^{\mathrm{N}}$ of the objective functions. Next, gather some training data from the question such that: if utopia $\boldsymbol{f}^{\mathrm{U}}$ takes the best score (say, $V(\boldsymbol{f}^{\mathrm{N}})$ =1.0) and nadir $\boldsymbol{f}^{\mathrm{N}}$ the worst ($V(\boldsymbol{f}^{\mathrm{N}})$= -1.0), then what is the score for a certain $\boldsymbol{f}^i$ locating between them ($V(\boldsymbol{f}^i)=a_i$, $-1 \leqq a_i \leqq 1$). Finally, using every $\boldsymbol{f}^i$ as input and a_i as output (i=1,..., k), train NN. As supposed easily, such assessment that requires DM to evaluate every trial point directly by the score is very difficult. It is also hard to keep consistency between the responses since there are only two definite standards, that is, a utopia point and a nadir point for every evaluation. So as the number of objectives increases just a little, it will become extremely hard to get a meaningful set of training data. Hereinafter we will call this direct method.

Instead, we propose another NN modeling method which we call relative method as follows (Shimizu, 1999b). In the relative method, training data will be gathered through pair comparisons like AHP (Analytic Hierarchy Process, Saaty, 1980). By taking a pair of trial points, DM is asked to express his/her relative preference of one to another. Just like AHP, such responses will be accomplished by using linguistic statements, and then transformed into the score given in **Table 1**. We can make the load of DM paying for the responses much less than that of the direct method since the relative comparison is generally much easier than the direct one.

Table 1 Conversion table

Linguistic statements	a_{ij}
equally	1
moderately	3
strongly	5
Demonstrably	7
Extremely	9
Intermediate judgement between the two adjacents	2,4,6,8

$i \diagdown j$	$\boldsymbol{f}^1$	$\boldsymbol{f}^2$	$\boldsymbol{f}^3$	...	$\boldsymbol{f}^k$
$\boldsymbol{f}^1$	1	a_{12}	a_{13}	...	a_{1k}
$\boldsymbol{f}^2$		1	a_{23}	...	a_{2k}
$\boldsymbol{f}^3$			1	...	a_{3k}
...	$a_{ji}=1/a_{ij}$			...	...
$\boldsymbol{f}^k$					1

Fig.1 Pair comparison matrix

After doing such $k(k-1)/2$ pair comparisons covering k trial points, we can obtain a pair comparison matrix as shown in **Fig. 1**. Its i-j element a_{ij} represents the degree of preference (score in Table 1) of $\boldsymbol{f}^j$ compared with $\boldsymbol{f}^i$ admitting that $a_{ii}=1$ and $a_{ji}=1/a_{ij}$. According to the theory of AHP, we are also easy to examine the consistency of such pair comparisons only by calculating the consistency index by $(r_{\max} - k) / (k - 1)$. Here, $r_{\max}$ denotes the maximum eigen value of the pair comparison matrix. It is empirically known that if the index value exceeds 0.1, there are involved undue responses in the matrix. In such a case, we need to revise certain scores to recover from the inconsistency.

After all, each element of the matrix provides totally k^2 training data for NN that has a feedforward structure consisting of three layers (*i.e.* $2N$ inputs, one output and an appropriate number of hidden nodes). Using some test problems, we ascertain this relative method can model a few typical value functions correctly by a reasonable number of pair comparisons.

Based on the value function thus modeled, we can rank any candidate solution as follows. First, suppose the NN model is a function from $2N$ dimensional space to a scalar one, *i.e.* V_{NN}: $\{\boldsymbol{f}^i(\boldsymbol{x},\boldsymbol{z}), \boldsymbol{f}^j(\boldsymbol{x},\boldsymbol{z})\} \in \mathrm{R}^{2\mathrm{N}} \rightarrow a_{ij} \in \mathrm{R}^1$. Then, fixing a half of the inputs at an appropriate reference point, say $\boldsymbol{f}^R$, we can calculate the score v_{iR} for any candidate $\boldsymbol{f}^i(\boldsymbol{x},\boldsymbol{z})$ ($i=1,\ldots,k$) by Eq.(1), and rank the preference of $\boldsymbol{f}^i(\boldsymbol{x},\boldsymbol{z})$ easily depending on the magnitude of v_{iR}.

$$V_{\mathrm{NN}}(\boldsymbol{f}^{i}(\boldsymbol{x},\boldsymbol{z});\ \boldsymbol{f}^{R}) = v_{iR}\ ,\ (i=1,\ldots,k) \qquad (1)$$

In summary, particular advantages of the relative method over the direct one are less load paid for the responses, and the existence of a proof method to check the consistency of the responses. Moreover, in contrast to the approach of utility theory, it needs not to examine the mathematical assumptions a priori, and can cope flexibly with the changes of decision making environment just by recalculating the weights of NN. Since the tradeoff analysis will be taken place just among a few major objectives in usual systems, we can use the proposed approach conveniently and widely.

2.3 Hybrid Scheme of GA with Repair Operation

To apply HybGA along with the framework depicted in **Fig.2**, we reformulate (p.2) as follows.

$$\text{(p.3)}\quad \underset{\boldsymbol{z}:\,\text{integer},\ \varepsilon_{-p}\geq 0}{\text{Maximize}}\ V_{\mathrm{NN}}\left(\boldsymbol{\varepsilon}_{-p}\ ,\ f_{p}(\boldsymbol{x},\boldsymbol{z});\ \boldsymbol{f}^{R}\right)$$

$$\textit{subject to}$$

$$\underset{\boldsymbol{x}\geq 0:\,\text{real}}{\text{Minimize}}\ f_{p}(\boldsymbol{x},\boldsymbol{z})$$

$$\textit{subject to}\begin{cases} f_{i}(\boldsymbol{x},\boldsymbol{z})\leq f_{i}^{*}+\varepsilon_{i}, & (i=1,\ldots,N,i\neq p) \\ g_{i}(\boldsymbol{x},\boldsymbol{z})\leq 0, & (i=1,\ldots,I1) \\ h_{i}(\boldsymbol{x},\boldsymbol{z})\leq 0, & (i=1,\ldots,I2) \end{cases}$$

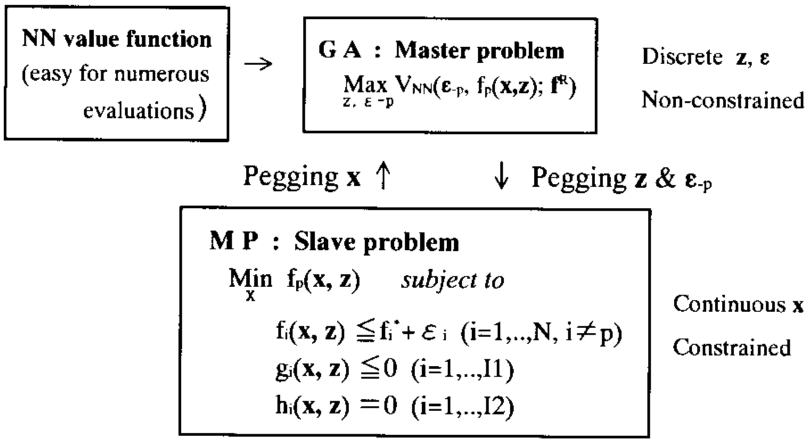

NN value function
(easy for numerous evaluations)
G A : Master problem
$\underset{z,\ \varepsilon_{-p}}{\text{Max}}\ V_{NN}(\varepsilon_{-p},\ f_p(\mathbf{x},\mathbf{z});\ \mathbf{f}^R)$
Discrete z, ε
Non-constrained
Pegging x
Pegging z & ε-p
M P : Slave problem
$\underset{x}{\text{Min}}\ f_p(\mathbf{x},\mathbf{z})$ subject to
$f_i(\mathbf{x},\mathbf{z}) \leqq f_i^* + \varepsilon_i$ (i=1,..,N, i≠p)
$g_i(\mathbf{x},\mathbf{z}) \leqq 0$ (i=1,..,I1)
$h_i(\mathbf{x},\mathbf{z}) = 0$ (i=1,..,I2)
Continuous x
Constrained

Fig.2 Outline of the proposed approach

Below, we will describe about the GA employed for the master problem of (p.3). This is basically a simple GA (Goldberg, 1989) except for the repair operation mentioned in the later.

(1) Representation of chromosome

As shown in **Fig. 3**, we adopt a binary representation where the first half of the chromosome corresponds to the integer variables, and the latter to the quantized amounts of degradation regarding ε−constraints except for the principal objective. Oppositely, they are decoded respectively as follows.

$$z_i = \sum_{j=0}^{J} 2^j s_{ij}\ ,\ (i = 1,\dots ,M\ ) \tag{2}$$

$$\varepsilon_i = \sum_{j=0}^{J'} 2^j s_{ij} \Delta\varepsilon_i\ ,\ \ (i = 1,\dots ,N\ ,i \neq\ p) \tag{3}$$

where each s_{ij} denotes 0-1 variable representing the binary type of allele, and $\Delta\varepsilon_i$ a unit degradation (a grain of quantization).

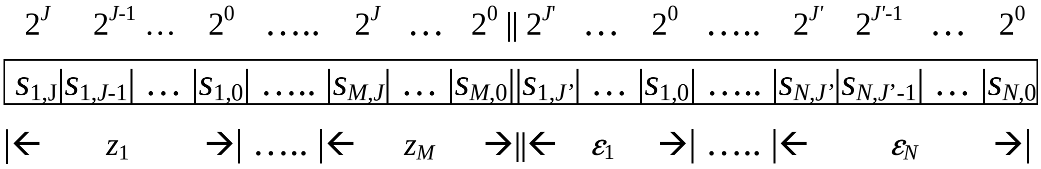

2^J 2^{J-1} … 2^0 ….. 2^J … 2^0 || $2^{J'}$ … 2^0 ….. $2^{J'}$ $2^{J'-1}$ … 2^0
$s_{1,J}$ | $s_{1,J-1}$ | … | $s_{1,0}$ | ….. | $s_{M,J}$ | … | $s_{M,0}$ || $s_{1,J'}$ | … | $s_{1,0}$ | ….. | $s_{N,J'}$ | $s_{N,J'-1}$ | … | $s_{N,0}$
z_1 ….. z_M || ε_1 ….. ε_N

Fig.3 Structure of the employed chromosome

The binary coding for ε_i like Eq. (3) is rather reasonable because human beings usually have a certain resolution identifying the difference of their preference between the two solutions. On the other hand, the binary coding for real variables causes a tradeoff problem between the efficiency and the accuracy. The longer the chromosome becomes, the less efficient the GA becomes. In another words, an accurate solution needs small grain which makes the chromosome long. The proposed approach will not be worried about this problem at all since we optimize the real variables based on MP.

(2) Reproduction : roulette wheel strategy couple with the elitist policy

(3) Crossover : one-point crossover per part as shown in **Fig. 4**.
(substantially two-points crossover)

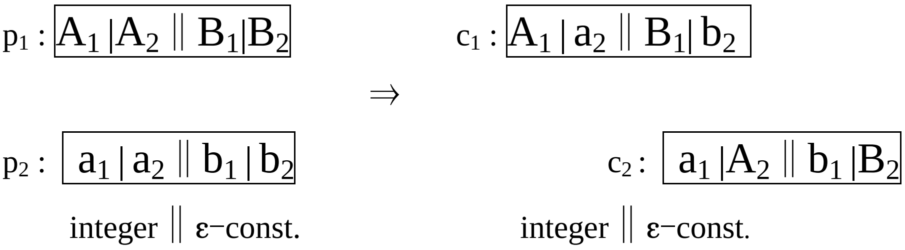

p_1 : A_1 | A_2 || B_1 | B_2
c_1 : A_1 | a_2 || B_1 | b_2
⇒
p_2 : a_1 | a_2 || b_1 | b_2
c_2 : a_1 | A_2 || b_1 | B_2
integer || ε−const.
integer || ε−const.

Fig.4 Employed crossover rule

(4) Mutation : binary bit entry flip (*i.e.* 0/1 or 1/0)

(5) Evaluation of fitness : transformed value of the output of NN model by a three-order algebraic equation like Eq.(4)

$$\text{Fitness}= \{V_{\mathrm{NN}}(\boldsymbol{\varepsilon}_{-p}, f_p; \boldsymbol{f}^R) - \underline{U})/\underline{U}\}^3 \tag{4}$$

where $\underline{U}$ denotes a certain nominal value. (Actually, we choose the minimum of V_{NN} at each generation as $\underline{U}$.)

(6) Repair operation : In the slave problem of (p.3), for a given set of $\boldsymbol{\varepsilon}$, there often occurs some ε−constraints become inactive. If this is true for a certain, say *i-th* objective, the master problem is to be unduly evaluated since the inequality, $V_{\mathrm{NN}}(\underline{\cdot}, \underline{\varepsilon}_i; \boldsymbol{f}^R) \leqq V_{\mathrm{NN}}(\underline{\cdot}, f_i(\underline{\boldsymbol{x}}^*, \underline{\boldsymbol{z}}); \boldsymbol{f}^R)$ is satisfied. Here, $\underline{\boldsymbol{x}}^*$ denotes the optimal of the slave problem obtained for the pegged $\boldsymbol{\varepsilon}_{-p}$, and $\boldsymbol{z}$. Repair operation can work to avoid such undue evaluation. When discrepancy of the inactive ε−constraint exceeds one grain of the quantization (*i.e.* $f_i^* + \underline{\varepsilon}_i - f_i(\underline{\boldsymbol{x}}^*, \underline{\boldsymbol{z}}) > \Delta\varepsilon_i$), the corresponding part of the chromosome should be rewritten so that $\underline{\varepsilon}_i < f_i(\underline{\boldsymbol{x}}^*, \underline{\boldsymbol{z}}) - f_i^* + \Delta\varepsilon_i$ will hold. Since this makes the constraint active within a unit grain, the Pareto optimality can be satisfied more likely at each generation compared with the approach without this repair operation.

To explain this, **Fig.5** is helpful. In the two-objective problem, suppose that we obtain two different solutions denoted by ② and ②' respectively under the ε−constraint such like $f_2(\boldsymbol{x},\boldsymbol{z}) \leqq f_2^* + \varepsilon_2$. There, ②' is active for the ε−constraint, while ② inactive. In terms of the foregoing description of the chromosome, ② is to be evaluated substantially at point ②'. Hence, nevertheless ② is truly preferable to ②', it is likely to be selected as the individual with low fitness. Such irrationality will be avoided if ② is evaluated at ②''. This is equivalent to remove the surplus and adjust the ε−constraint when the inactive amount exceeds the unit grain $\Delta\varepsilon_2$. We can view such dealing as an analogy of natural life that will repair the damaged string of DNA by copying from the other normal one.

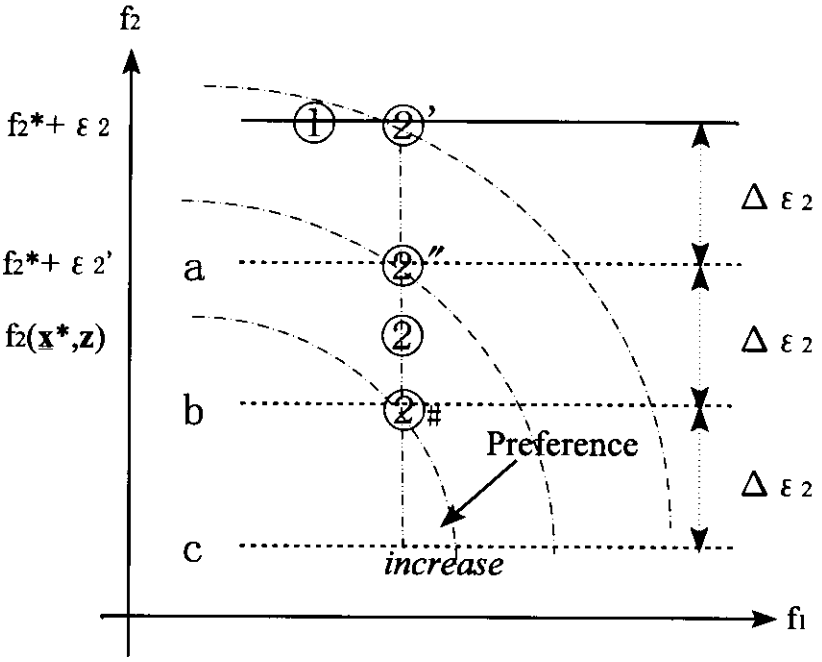


Fig.5 Illustration of the repair operation

By doing that, since every solution locating between the dotted line a and b is viewed same as ②", and also every solution between b and c as ②$^{\#}$, we can restrict the search space (numbers of combination) in the master problem. Again, the following simple example will help explain this effect more clearly. If the schema of a certain active constraint is supposed to be [0 | 1 | 0], then each element of the set { [0 | 1 | 1], [1 | 0 | 0], [1 | 0 | 1], [1 | 1 | 0], [1 | 1 | 1]} is viewed damaged, and rewritten (repaired) as [0 | 1 | 0]. Thus the repair operation can bring about the capability to improve the convergence of the search besides the likeliness of the Pareto optimality.

3 FORMULATION OF A TYPICAL OF SITE LOCATION PROBLEM

Associated with what is known as the environmental problems, location problem of hazardous waste disposal site is becoming very important these days. Since the resolution of conflicting objectives between economy and risk is common to this kind of NIMBY (Not In My Back Yard) problem, consideration as the multi-objective optimization is quite amenable as a practical concern. With such a point of view, we take a site location problem whose basic but general scheme is shown in **Fig. 6**. The problem thereat can be described such that : for rational disposal of the hazardous waste generated at *L* sources, choose the suitable sites among the *M* candidates. This problem can be formulated as follows.

$$\text{(p.4)}\quad \underset{\boldsymbol{z},\,\varepsilon_1}{\text{Maximize}}\ V_{\text{NN}}\left(\varepsilon_1,\ f_2=\sum_{i=1}^{M}\sum_{j=1}^{L}C_{ij}x_{ij}+\sum_{i=1}^{M}F_i z_i;\ \boldsymbol{f}^{R}\right)-P\cdot\text{MAX}\left(0,\sum_{i=1}^{M}z_i-K\right)$$

subject to

$$\underset{\boldsymbol{x}}{\text{Minimize}}\quad f_2=\sum_{i=1}^{M}\sum_{j=1}^{L}C_{ij}x_{ij}+\sum_{i=1}^{M}F_i z_i$$

$$\textit{subject to}\ \begin{cases} f_1=\displaystyle\sum_{i=1}^{M}\sum_{j=1}^{L}R_{ij}x_{ij}+\sum_{i=1}^{M}Q_iB_iz_i\le f_1^{*}+\varepsilon_1 & (5)\\ \displaystyle\sum_{i=1}^{M}x_{ij}\ge D_j\ ,(j=1,\dots,L) & (6)\\ \displaystyle\sum_{j=1}^{L}x_{ij}\le B_iz_i\ ,(i=1,\dots,M) & (7)\end{cases}$$

In the above, f_1 and f_2 denote the objective functions evaluating cost and risk respectively. They are functions of the amount of waste shipped from source j to site i, $x_{ij}(\geqq 0)$, and 0-1 variable z_i ($\in\{0,1\}$) which takes 1 if the *i-th* site is chosen and 0 if otherwise. Moreover, D_j denotes demand at the *j-th* source, B_i capacity at the *i-th* site. Then Eq. (6) describes that the waste is shippable at each source, and Eq. (7) disposable at each site. Moreover, K is an upper bound of the permitted construction. This constraint was handled by a penalty term in the objective function at the master problem where P denotes a penalty coefficient, and MAX(・) requires taking the greatest among the elements in the parenthesis.

On the other hand, C_{ij} denotes shipping cost from j to i per unit amount of waste, and F_i fixed-charge cost of site i. R_{ij} denotes the risk portion accompanied with transportation per unit amount from j to i. It should be a function of distance, population density along the traveling route, and other specific factors in real application. Likewise, Q_i represents the fixed-portion of risk at the *i-th* site per unit capacity, and is considered to be a function of population density around the site, and some other specific factors in real applications. Since the system equations and two objectives are all linear functions of the decision variables, it is easy to solve the slave problem using linear programming even if the problem size becomes very large.

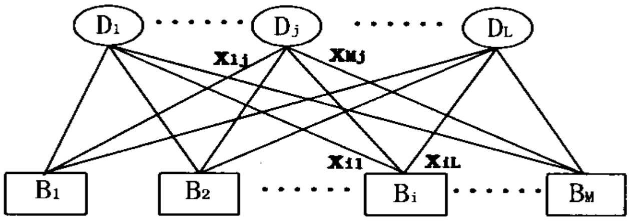


Fig. 6 Basic scheme of site location problems

4 NUMERICAL EXPERIMENTS

To examine the effectiveness of the proposed method, we carried out numerical experiments with the problem size like M=8, L=6 and K=3 by supposing the author as DM. We gave appropriate normalized values to the system parameters just for convenience' sake of the numerical experiments in the present study.

First referring to the pay-off matrix obtained through solving the mixed-integer linear programming problem under each objective function repeatedly, we gave five points as trials for the pair comparisons, and obtained the pair comparison matrix like **Fig. 7**. Then checking the consistency of the pair comparisons from the consistency index (7.014E-2 <0.1), we obtained the NN model of value function using each element of the matrix as the training data. In the case of 10 nodes in the hidden layer, we obtained the NN model with accuracy like shown in **Fig. 8** where the outputs are close to the true values (training data) almost everywhere (average square root error in the modeling was 1.83E-2.).

	ref (1)	ref (2)	ref (3)	utopia	Nadir
ref (1)	1	3	1/3	1/5	5
ref (2)	$a_{ji}=1/a_{ij}$	1	1/5	1/6	3
ref (3)			1	1/3	5
utopia				1	9
nadir					1

Fig. 7 Pair comparison matrix of the example

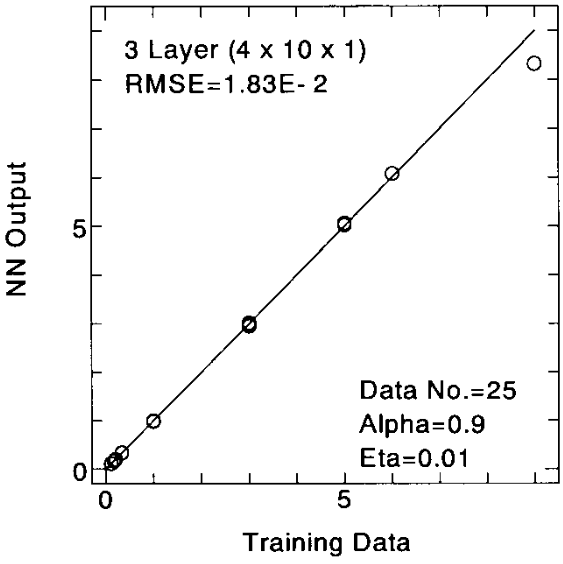
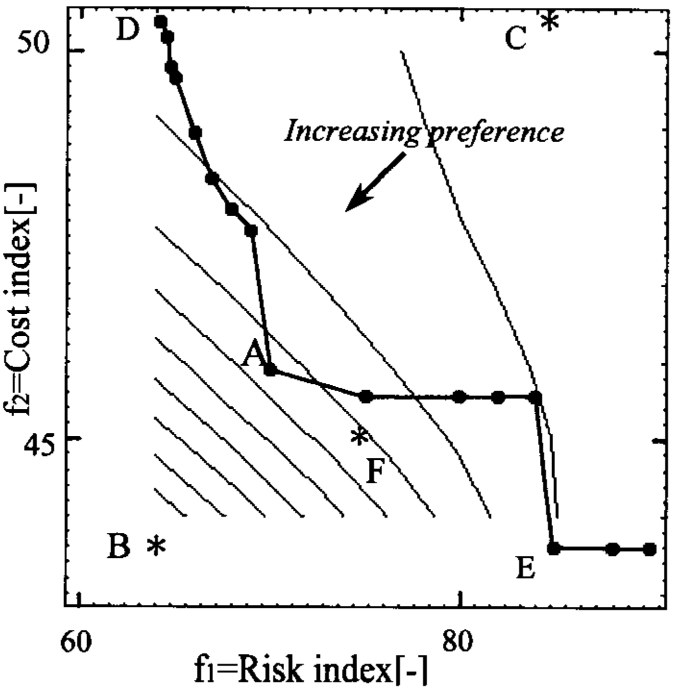


Fig.8 Performance of the NN modeling

After all these steps, we derived the BCS at 14 generations by evaluating the value function totally 201 times under the following conditions of GA. We set the crossover rate as 0.1, mutation rate as 0.01, and population size as 50 for the chromosome 11 bits long. We also supposed that GA has converged when the number of individuals with the best fitness at each generation exceeds that of 90% of the total population. It took within a second by using a standard type of work station (SONY NEWS 4000E).

From **Fig. 9** where POS set are imposed on the contour of value function of the NN model, we can ascertain that the present BCS locating at point A is quite reasonable. That is, it locates on the POS set and also has the highest value of the value function at the same time. This will verify the effectiveness of the proposed method numerically.



A: Best-compromise, B: Utopia, C: Nadir

D: Ref.(1), E:Ref(2); F:Ref(3)

—●— : Pareto optimal set

Fig. 9 Feature of result in objective space

Furthermore, to examine the effectiveness of the repair operation mentioned in § 2.3, we compared the performance to solve (p.4) among the following three approaches.

(1) no concern ; apply the usual HybGA.

(2) penalty approach ; to compensate the degradation due to the inactive ε-constraints, add a penalty term in the master problem such like,

$$P\left\{ \sum_{i=1}^{N} \left[\left(\underline{\varepsilon}_i - f_i(\underline{\boldsymbol{x}}^*, \underline{\boldsymbol{z}}) \right) / \Delta\, \varepsilon_i \right] \right\} \tag{8}$$

(3) HybGA with the repair operation

We summarized the result in **Table 2** where the partition number is given as $\Sigma^{J'}_{j=0}\ 2^j$. Since it increases together with *J'*, for the experiment, we can change the size or search space in the master problem through the partition number or *J'*. The larger such values become, the more rapidly possible combinations that make the solution difficult will increase. As shown in the table, performance of the proposed approach is superior to the others. Also, it could cope with the expansion of the size while the others always could not attain at the best compromise solution and/or required more evaluations till convergence. This is why the repair operation can reduce the expansion of search space effectively as well as decrease the number of death of the possibly superior individuals through the proper evaluation of the fitness as explained already. Even more, observing such a tendency that computation load is declining with the expansion, we can expect to apply this approach to large-scale real world problems.

Table 2 Comparison of evaluation among three methods

Partition No. (*J'*)	7 (2)	15 (3)	31 (4)	63 (5)	511 (8)	4095 (11)	131071 (16)
(1) No concern	496	566	~~688~~	~~889~~	763	~~1181~~	~~1128~~
(2) Penalty	534	635	~~704~~	908	~~1050~~	~~1226~~	~~1146~~
(3) Repair	**481**	**539**	**738**	**692**	**679**	**839**	**885**

~~1234~~ unattainable to the best-compromise solution

Population=150, Crossover rate = 0.1, Mutation rate=0.01

5 CONCLUSION

Flexible and intelligent supporting methods for decision making will be highly desired in the coming century when technical problem-solving will become more and more complicated and inter-related. In such a situation, we can formulate a variety of problems as multi-objective mixed-integer programs (MOMIP). Taking a site location problem of hazardous waste as an eligible case study, in this paper, we concern ourselves with the solution of MOMIP while weighing on the practice rather than the mathematical rigidity. Then, to solve the problem practically, we have presented a hierarchical approach composed of GA and mathematical programming in the aid of neural network (NN) for modeling the value function and the repair operation in genetic search. As a result, we can provide an effective method which can derive the best compromise solution practically while the conventional methods known as multi-objective genetic algorithms remain at the level computing the Pareto optimal solution set. Furthermore, by virtue of the identification process of the value function, the proposed method is superior to the goal programming (GP) which has a stiff structure to perform the tradeoff analysis.

Though the present study revealed the effectiveness concerning only with a small problem, the proposed approach is promising for real world applications due to the high ability of each solution method (i.e. genetic algorithm and mathematical program), and the practical modeling method of value function using NN. Real-world applications to make such assertion certain is left for further studies.

REFERENCES

Cohen,J.L.(1978) Multi-objective Programming and Planning. Academic Press, New York.

Fonseca,C.M. and Fleming, P.J. (1993) Genetic algorithm for multi-objective optimization: formulation, discussion and generalization. **Proc. 5th Int. Conf. on Genetic Algorithm and Their Applications**, Morgan Kaufmann Publisher, Chicago, USA, 416-423, 1993.

Goldberg, D.E. (1989) Genetic Algorithm in Search, Optimization, and Machine Learning. Addison-Wesley, Mass..

Keeney, R.L. and Raiffa, H. (1976) Decision with Multiple Objective Preference and Value Tradeoffs. John Wiley & Sons, New York.

Louis,S.J. and Rawling, G.J.E. (1993) Pareto optimality, ga-easiness and deception. **Proc. 5th Int. Conf. on Genetic Algorithm and Their Applications**, Morgan Kaufmann Publisher, Chicago, USA, 118-123, 1993.

Malakooti, B. and Zhou, Y. Q. (1994) Feed forward artificial neural networks for solving discrete multiple criteria decision making problems. **Manage. Sci. 40**, 1542-1561.

Murata,T., Ishibuchi, H. and Tanaka, H. (1995) Flowshop scheduling by genetic algorithm and its application to multi-objective problems. **Keisoku Jidoseigyo Gakkai Ronbunshu 31**, 583-590 *(in Japanese)*.

Saaty,T.L. (1980) The Analytic Hierarchy Process. McGraw-Hill, New York.

Schaffer,J.D. (1985) Multiple objective optimization with vector evaluated genetic algorithm. **Proc. 1st Int. Conf. on Genetic Algorithm and Their Applications**, Lawrence Erlbaum Associates Publisher, Pittsburgh, USA, 93-100, 1985

Shimizu,Y. (1999a) Multiobjective analysis of mixed-integer linear programs through a hybrid use of genetic algorithm with mathematical programming. **Kagaku Kogaku Ronbunshu 25**, 66-72 *(in Japanese)*.

Shimizu,Y. (1999b) Multiobjective optimization for site location problems through hybrid genetic algorithm with neural networks. **J. Chem. Engng. Japan 32**, 51-58.

Tamaki, H., Mori, M. and Araki, M. (1995) Generation of a set of Pareto-optimal solutions by genetic algorithms. **Keisoku Jidoseigyo Gakkai Ronbunshu 31**, 1185-1192 *(In Japanese)*.

Venugopal,V. and Narendran, T.T. (1992) A genetic algorithm approach to the machine-component grouping problem with multiple objectives. **Computers Ind. Engng. 22**, 469-480.