Engineering

Industrial & Management Engineering fields

Okayama University *Year* 1995

GA-based decision support system for multicriteria optimization

Masahiro Tanaka
Okayama University

Hikaru Watanabe
Okayama University

Yasuyuki Furukawa
Okayama University

Tetsuzo Tanino
Okayama University

GA-Based Decision Support System for Multicriteria Optimization

Masahiro Tanaka, Hikaru Watanabe, Yasuyuki Furukawa and Tetsuzo Tanino
Department of Information Technology
Okayama University
Tsushima-naka 3-1-1, Okayama 700
Japan

ABSTRACT

In this paper, we propose a multicriteria decision making (MCDM) method by using a genetic algorithm (GA). The system consists of three phases. In the first phase, a rough set of Pareto optimal solutions is obtained using Kohonen's self organizing map (SOM). In the second phase, the decision maker (DM) selects his preferred solutions among the obtained set, where the mechanism of GA is used with the DM's preference assisted by radial basis function network (RBFN). In the third phase, the DM can explore the solution space further for the final decision.

1. INTRODUCTION

MCDM is often practically important in the real world. In these problems we can seldom expect the existence of the dominating solutions which optimize the several objectives simultaneously. Therefore Pareto optimal solutions are crucial.

For the MCDM problems, we have two important issues. They are

1. generating feasible Pareto optimal solutions,
2. making decision by DM's preference among Pareto optimal solutions.

In the traditional MCDM methods, the first problem was reduced to the mathematical programming problem by aggregating the vector objective function into a scalar by pre-determined manner or by some interactive manner mentioned in the second item above. This method intrinsically produces a single solution, and thus the searching point is usually only one. The primordial drawback of the traditional methods is that they all lack the parallel production of searching points. Considering the humans' decision process, we cannot be satisfied without comparing many alternatives simultaneously. Even if the shown alternative is "best one", we may have a difficulty in decision making and doubt whether there is no better ones for him/her.

Recently, there are studies to generate many Pareto optimal solutions by using GA[10],[7],[1]. This is certainly an attractive method, because the GA intrinsically treats multiple alternatives. But there is some kind of inefficiency in generating Pareto optimal solutions by GA-based methods that have been proposed so far, because the new alternatives are to be generated by using only feasible solutions and thus it is not likely to be easy to generate ones that are really near the frontier. Thus we propose here another method to generate Pareto optimal solutions. It is based on Kohonen's SOM.

We are interested not only in generating Pareto optimal solutions, but also in the DM's selection procedure.

The authors proposed an interactive method for MCDM[13]. In terms of GA, selection is based on ranking, where the rank levels are two. The rank is decided by the Pareto optimality and DM's preference. In this paper, we propose a revised version of the algorithm. The critical problem is the times of interactions between the DM and the DSS when the number of objective function is large. For the GA to work efficiently, a good amount of the evaluation points by the DM is necessary, which forces the DM to input evaluations of many alternatives. To circumvent this problem, the normalized RBFN is applied to form the utility function.

2. PROBLEM FORMULATION

The MCDM is given by

$$\text{(P1)} \begin{cases} \text{minimize } f(x) = (\phi_1(x), \cdots, \phi_p(x)) \\ \text{subject to } x \in X \subset \mathbf{R}^n \end{cases}$$

where X is the set of admissible solutions given *a priori* explicitly and/or implicitly by inequalities. The solutions of the problem (P1) are called Pareto optimal solutions, and the whole set of Pareto optimal solutions is denoted as $\tilde{X}(\subset X)$. Define also

$$\tilde{F} = f(\tilde{X}) = \{f(x) : x \in \tilde{X}\}$$

The DM is in charge of getting most preferable solution $x^* \in \tilde{X}$.

0-7803-2559-1/95 $4.00 © 1995 IEEE

3. REVIEW OF RELEVANT STUDIES

3.1 Traditional MCDM Methods

In traditional MCDM methods, the single objective function or the aggregated vector objective function is optimized by mathematical programming methods. The parameters for the aggregation are

- weight coefficients (weighting method, weighted minimax method)
- weight coefficients and the goal (goal programming, compromise programming)
- admissible objectives bound (ε-constraint method[3])
- reference point (reference point method[14])

and so on. These parameters are often input interactively, but they all have the common feature that they yield only one solution at a time basis. Figure 1 shows the concept of traditional MCDM flow.

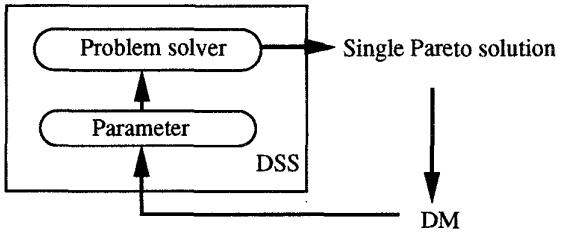

Fig. 1. Traditional MCDM

3.2 Generating Pareto Optimal Solutions by GA

In real world problems, DMs often need different alternatives in decision making[11]. Recently GAs have been attracting our attention as an efficient technique for generating Pareto optimal solutions.

GA is an optimization method by using multiple alternatives (or "individuals" in GA terminology), where "crossover", "mutation" and "selection" are the fundamental genetic operators[2]. GA is used for the optimization problems mainly because of its ability to produce globally (semi) optimal solution. But it has been modified so that it can yield multimodal problems, where techniques such as crowding and niching are created to avoid the individuals to converge and to let the individuals be more variety. Further, by using such techniques, GA has been applied to exhaustively generating Pareto optimal solutions.

Schaffer[10] developed a program called Vector Evaluated Genetic Algorithm (VEGA). In VEGA, selection is made based on each component's value interchangeably. In other words, let $x_i \in X$. For $j = 1, ..., p, 1, ..., p, ...$, selection is made based on the fitness $\phi_j(x)$ with GA's selection mechanism. This has the drawback to select only the element-wise extremals. Thus, two operations are supplementally used to avoid this phenomenon. One is "mate selection", which is to mate individuals generated based on different elements, and another is "non-dominated selection" that is to penalize non-Pareto solutions. That paper has been the milestone of the research on the use of GA in generating Pareto optimal solutions.

Ranking is used for the selection operation, so that Pareto optimal solutions are more likely to remain in the next generation. Individuals of the same rank should have the same selection pressure. To alleviate the bias of alternatives (i.e. to generate alternatives uniformly on the Pareto front), "niching" technique was adopted. Horn et al.[4] proposed similar technique. Srinivas and Deb[11] proposed also a similar technique, where the sharing method has some difference from that of Fonseca and Fleming.

3.3 Interaction for Higher-Level Decision Making

If the objective space is one or two-dimensional, the DM can easily decide his/her attitude by seeing the plot shown on the screen. But for high dimensional problems, this is not a trivial problem.

Fonseca and Fleming[1] used "goal attainment method" as the core methodology for this purpose. There are two kinds of parameters in that method: the goal vector and the weight coefficient for each objective. They used GA for optimizing by the goal vectors.

Another method is to use the collection of Pareto optimal solutions as a database, and the DM searches his/her preference solution among these finite number of given points. Next we introduce a GA-based technique for the higher-level decision making.

3.3.1 GA-Based Interaction. The authors have already proposed the original version of the algorithm[13]. In terms of GA, selection is based on ranking, where the rank levels are two. The rank is decided by the Pareto optimality and DM's preference. If an individual is both Pareto optimal and tentatively acceptable by the DM, its rank is marked high, otherwise it is marked low.

Next we show the algorithm in [13]. Each alternative is coded as a chromosome in binary numbers.

The algorithm can be described as follows.

Step A-1: $k := 0$. Generate an appropriate number of individuals $x_1, x_2, ...$ (values of the decision variable) and compute their objective function vectors $(f(x_1), f(x_2), ...)$. Remove dominated members, i.e., leave only Pareto optimal solutions. Repeat generating decision variables until the number of remaining members becomes appropriately large. The surviving members constitute the initial population $\tilde{X}(0)$ for the subsequent search.

Step A-2: $k := k + 1$. Create new individuals $\tilde{X}(k)$ by mating current ones; apply mutation and recombination as the parent individuals mate.

Step A-3: Evaluate the obtained individuals and let

$$\tilde{F}(k) = \{f(x) : x \in \tilde{X}(k)\}$$

Remove dominated (non-Pareto optimal) solutions from $(\tilde{X}(k), \tilde{F}(k))$.

Step A-4: If Steps A-2 and A-3 have been repeated several times, go to Step A-5. If not, reproduce the remaining individuals and return to Step A-2.

Step A-5: Show $\{\tilde{X}(k), \tilde{F}(k)\}$ to the DM. If he accepts one of them, then stop. If he is not satisfied with any solution, the DM provides his preference attitude based on the solutions shown to him, and return to Step A-2. He may provide the information on his preference attitude in three different ways.

1) He chooses satisfactory solutions and unsatisfactory solutions. Reproduce the satisfactory solutions and remove the unsatisfactory solutions.

2) He inputs the aspiration levels of the objective functions, which constitute a desirable point for him. Compute the distances between the aspiration levels and the shown solutions. Delete (reproduce) the solutions for which the distances are long (short).

3) He inputs the level of each objective function which should be satisfied at the worst. If every objective value of a solution is better than the above level, reproduce it. Otherwise, delete it.

We have found some drawbacks of the above algorithm. They are

premature convergence This occurs because only the Pareto optimal solutions remain.

interaction times In the interaction method 1), the number of evaluations of individuals tend to be large.

The items 2) and 3) above are the methods used in the traditional MCDM methods. They can be effectively used for scalarization, but it is doubtful if the required parameters can be easily determined. Moreover, they are not oriented for treating various preferable sets of alternatives. Fonseca and Fleming[1] used goal programming for GA whose approach is like 2).

4. NEW MCDM METHOD

The algorithm consists of three phases:

1. Generating Pareto optimal solutions
2. Selecting among the prepared alternatives
3. Exploring new alternatives

The basic flow is denoted in Figure 4. In the following, these phases are explained in more detail.

4.1 Phase 1

Step 1 (Generation of Pareto optimal solutions)
$k := 0$, and generate Pareto optimal solutions. Here, we propose a new method based on Kohonen's SOM, which is the mapping from input data to 2-

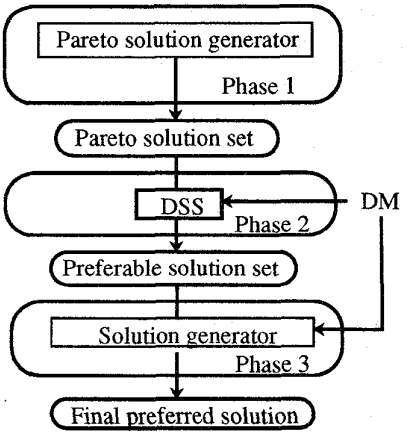

Fig. 2. Flow of the algorithm

dimensional grid, on each grid point a vector of the size of input vector is prepared[5], [6]. It has two convenient features for this applications:

1. The topological relation is well preserved.
2. The distribution of input data reflects to the number of grid points on the map.

The first feature can be used for detecting Pareto optimal points. Suppose a grid point p is mapped from different category of data $x_1 \in X$ and $x_2 \in \bar{X}$, i.e. feasible and non-feasible solutions. This means x_1, x_2 are near the boundary. If $x_1 \to p_1$, $x_2 \to p_2$ and (p_1, p_2) are neighbors, (x_1, x_2) are also understood as located near boundaries. Pareto optimal is judged among the vectors that appeared above.

The second feature can be used for generating solution alternatives near the boundary. The point that should be further explored is where

- in the map, different category data are mapped nearby
- the code vector on SOM is not very similar.

In this algorithm, pick up the feasible solutions found there, and judge whether they are Pareto optimal or not by comparing them to other vectors. Suppose K Pareto optimal solutions were found. Then create new vectors around them so that they are not dominated until the total number amounts to the original number of vectors.

For the problems where finite number of alternatives X are given *a priori*, the algorithm can be used beginning with the second phase using $\tilde{X}$: the Pareto optimal ones in X. Because the input-output relation is already found and there is no need to generate new points in the first place, it is sufficient to use $\tilde{F}$ for GA operations in the next phase.

4.2 Phase 2

In this phase, DM selects his preferred solutions among the prepared $(\tilde{X}, \tilde{F})$, and find most preferable ones in

the iterating interaction.

Step 2: Here the DM chooses samples from $\tilde{F}(k)$. Suppose that the sample size is N.

Step 2-1: Pick up a point $f \in \tilde{F}$ randomly and $\Psi(k) = \{f\}$.

Step 2-2: Randomly pick up two samples f_1, f_2 from $\tilde{F}$. Select the one $\hat{f}$ whose smallest distance from the already selected ones is larger than the other, i.e.

$$\|\hat{f} - p\| \geq \|f_i - q\|, \quad \forall p, q \in \Psi(k); i = 1, 2.$$

and add this point to $\Psi(k)$:

$$\Psi(k) := \Psi(k) \cup \{\hat{f}\}$$

Step 3: Show estimates of utility function

Except the first generation, DSS shows the estimate of the alternatives in $\Psi(k)$. The histogram of their utility value is also shown to DM.

The estimates are computed by using normalized RBFN. Suppose the DSS has collected M evaluations $y(f_1)$, $y(f_2)$, $\cdots$, $y(f_M)$ by the DM's preference. Based on these evaluations, we form the normalized RBFN as

$$\hat{y}(f) = \sum_{i=1}^{M} w_i \eta(\|f - f_i\|) / \sum_{j=1}^{M} \eta(\|f - f_j\|)$$

where the suffix is the index, η is the RBF which is here defined as a Gaussian function

$$\eta(r) = \exp(-r^2/\sigma^2)$$

where σ is a predetermined parameter. This network smoothly connects the evaluated value, and keeps the value outside the evaluated values. To decide $\{w_i\}$ from the given data, we can get the coefficients $w_1, \cdots, w_M$ by

$$\begin{bmatrix} w_1 \\ \vdots \\ w_M \end{bmatrix} = \begin{bmatrix} p_{11} & \cdots & p_{1M} \\ \vdots & & \vdots \\ p_{M1} & \cdots & p_{MM} \end{bmatrix}^{-1} \begin{bmatrix} y(f_1) \\ \vdots \\ y(f_M) \end{bmatrix}$$

where

$$p_{ij} = \eta(\|f_i - f_j\|) / \sum_{l=1}^{M} \eta(\|f_i - f_l\|)$$

Input preference When the shown preference values are not appropriate, the DM inputs his preference values $y(f_i)$ to the sample data $f_i \in \Psi(k); i = 1, 2, \cdots$.

Re-calculate RBFN Based on the evaluates $\{y(f_i) : f_i \in \Psi(k); i = 1, 2, \cdots\}$, DSS forms an RBFN. This step exploits the past interaction to release the DM's task, but the DM has to remember his attitude he made in the past. Here, we took an approach to add the outputs of RBFNs that were obtained up to the current iteration.

Figure 3 shows the given data, and Figure 4 shows the function surface generated by normalized RBFN using data in Figure 3, where $\sigma = 1$.

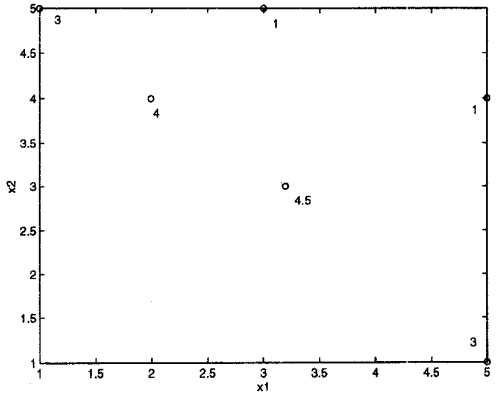

Fig. 3. Given data points (circle is the position, value is the utility value)

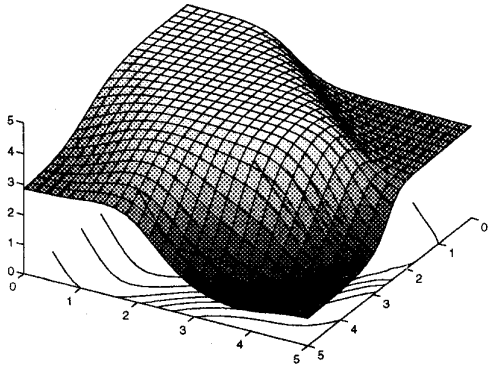

Fig. 4. Function surface by normalized RBFN

Step 4: Crossover and mutation operations are done in the function space. The nearest point in $\tilde{F}$ to the newly generated point is selected as a new member of $\Psi(k)$ if it is not a member yet.

Step 5: $k := k + 1$ and go to Step 2.

The GA operations are based on the methods by Michalewicz[8]. In the ranking method, the reproductive number is based on the ranking,

$$r = -aq + b$$

where r is the reproductive number, q is the ranking, with $b = 2$ or 3 and $a > 0$ is decided appropriately.

Table 1 GA used here

code	real value code
crossover	arithmetical crossover, simple crossover[8]
mutation	non-uniform mutation $p_m = 0.1$
selection	ranking selection

4.3 Phase 3

In the last phase, DM has selected his preferred solution(s) among the prepared alternatives. Here, he searches and makes the final decision.

Step 6: Explore new points of DM's preference.

DSS shows the final alternatives to DM. DM selects several points that he likes among them. Due to GA, new points are generated, and this interaction is repeated until he is satisfied.

5. NUMERICAL EXAMPLE

Here we demonstrate a numerical example with two-dimensional input. The problem is given by

$$(\mathrm{P2})\begin{cases} \text{minimize } f(x) = (x_1, x_2) \\ \text{s. t.} \quad x_1^2 + x_2^2 \geq 1 + 0.1\cos\left(16\arctan\frac{x_2}{x_1}\right) \\ \qquad \left(x_1 - \frac{1}{2}\right)^2 + \left(x_2 - \frac{1}{2}\right)^2 \leq \frac{1}{2} \\ \qquad 0 < x_1, x_2 \leq \pi \end{cases}$$

Figure 5 shows the feasible solutions. Note that such an exhaustive solutions are not shown on a plane nor generated for high dimensional system. This figure is only for demonstration.

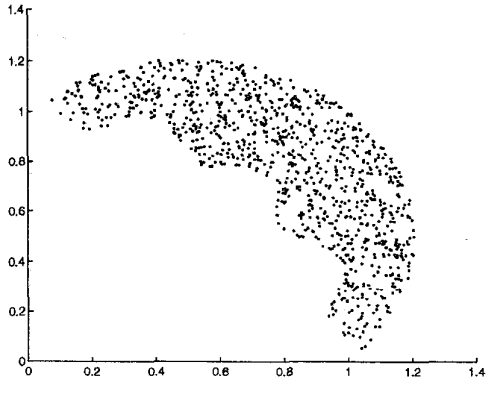

Fig. 5. Feasible solutions

The used SOM is 10×10 size. It was learned by presenting the 30 times of 500 learning vectors interchangeably.

Figure 6 shows the SOM at the second generation. We can see that the vectors are more concentrated at the Pareto optimal frontier.

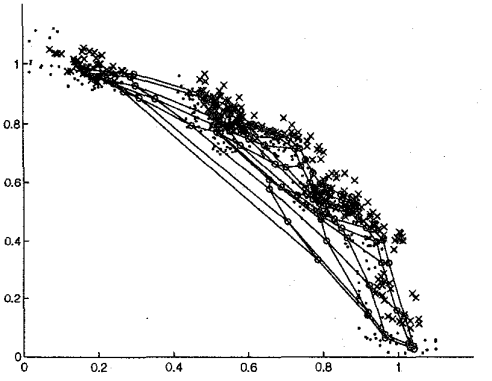

Fig. 6. SOM at second generation

Figure 7 shows the set of solutions that were finally generated (10th generation) in phase 1, and these are shown to DM. In high dimensional systems, this kind of map cannot be shown. The vector values (or some visually shown figure in certain sophisticated manner) of the points denoted by the circles are shown to DM.

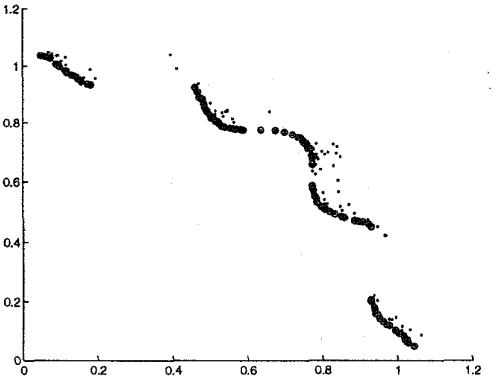

Fig. 7. Alternatives finally generated in phase 1

Now phase 2 begins. Table 2 shows the samples at 1st run. DM inputs evaluations for 10 alternatives. The size of population is between 100 and 150 whose number is variate on the selection condition. In normalized RBFN, $\sigma = 0.1$

Table 3 shows the samples at the second run. The third row shows the estimated evaluations by using the RBFN. The 4th row shows the values entered by the DM.

Table 4 shows the 5th run. Suppose that the DM satisfies

Table 2 Entered preferences at the 1st turn

x_1	0.05	0.12	0.48	0.51	0.56
x_2	1.04	0.98	0.87	0.80	0.78
input	0	0	1	3	5
x_1	0.75	0.78	0.78	0.83	1.04
x_2	0.73	0.57	0.54	0.49	0.05
input	3	1	0	0	0

Table 3 Entered preferences at the 2nd turn

x_1	0.47	0.51	0.59	0.62	0.65
x_2	0.91	0.81	0.77	0.77	0.77
RBFN	1.74	3.03	3.92	4.03	3.93
input	1	2	5	4	3
x_1	0.76	0.77	0.81	0.85	0.86
x_2	0.71	0.57	0.51	0.48	0.48
RBFN	2.64	0.51	0.28	0.16	0.15
input	2	1	0	0	0

with this evaluations.

Table 4 Result at the 5nd turn

x_1	0.565	0.568	0.568	0.572	0.579
x_2	0.775	0.775	0.775	0.775	0.774
RBFN	10.33	10.38	10.39	10.45	10.54
x_1	0.585	0.595	0.596	0.601	0.626
x_2	0.774	0.773	0.773	0.773	0.773
RBFN	10.625	10.703	10.707	10.723	10.586

Table 5 shows the best alternatives obtained at the final run. In this case, the DM can easily decide his best alternative among these ones. So, phase 3 is omitted.

6. CONCLUSIONS

We proposed an interactive decision making method for multi-criteria optimization problems. Various techniques are included: to generate many Pareto optimal solutions by SOM, and GA-based interaction for selecting preferred solutions, and GA-based generation of new alternatives. The algorithm is particularly useful in the problems of which objective functions are not given explicitly and/or not differentiable. A numerical example shows the basic interaction manner. RBFN was used for the selection, but there still remains to develop a sophisticated selection method.

REFERENCES

[1]C. M. Fonseca and P. J. Fleming, "Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization", *Proc. of 5th Int. Conf. on Genetic Algorithms*, pp. 416-423, 1993.

[2]D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley, 1989.

[3]Y. Y. Haimes and W. A. Hall, "Multiobjectives in Water Resources Systems Analysis: The Surrogate Worth Trade-Off Method", *Water Resources Res.*, 10, pp. 614-624, 1974.

[4]J. Horn and N. Nafpliotis and D. E. Goldberg, "A Niched Pareto Genetic Algorithm for Multiobjective Optimization", *Proc. 1st IEEE Conf. on Evolutionary Computation*, Vol. 1, pp. 82-87, 1994.

[5]T. Kohonen, *The Self-Organization and Associative Memory*, Springer-Verlag, New York, 1989.

[6]T. Kohonen, *Self-Organizing Maps*, Springer-Verlag, Berlin, 1995.

[7]F. Kursawe, "Evolution Strategies for Vector Optimization", *Proc. 10th Int. Conf. on Multiple Criteria Decision Making*, Vol. 3, pp. 187-193, 1992.

[8]Z. Michalewicz, *Genetic Algorithms + Data Structure = Evolution Programs*, Springer, 1994.

[9]Y. Sawaragi, H. Nakayama and T. Tanino, *Theory of Multiobjective Optimization*, Academic Press,1985.

[10]J. D. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms", *Proc. Int. Conf. on Genetic Algorithms and Their Applications*, pp. 93-100, 1985.

[11]N. Srinivas and K. Deb, "Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithm", *Evolutionary Computation*, Vol. 2, No. 3, pp. 221-248, 1995.

[12]R. E. Steuer, *Multiple Criteria Optimization: Theory, Computation and Application*, Wiley, 1986.

[13]T. Tanino, M. Tanaka and C. Hojo, "An Interactive Multi-Criteria Decision Making Method by Using a Genetic Algorithm", *The 2nd Int. Conf. on Systems Science and Systems Engineering*, pp. 381-386, 1993.

[14]A. P. Wierzbicki, "The Use of Reference Objectives in Multiobjective Optimization", in *G. Fandel and T. Gal (eds.), Multiple Criteria Decision Making: Theory and Application*, pp. 468-486, 1980.

Table 5 Best alternatives at the 5nd turn

Best	1	2	3	4	5
x_1	0.6005	0.5958	0.5953	0.5951	0.6147
x_2	0.7731	0.7732	0.7734	0.7734	0.7731
No.	6	7	8	9	10
x_1	0.6149	0.5931	0.5294	0.6170	0.6214
x_2	0.7731	0.7731	0.7734	0.7731	0.7731