Proceedings of the 2000 IEEE International Workshop
on Robot and Human Interactive Communication
Osaka, Japan - September 27-29 2000

Genetic Symbiosis Algorithm for Multiobjective Optimization Problem

Jiangming MAO, Kotaro HIRASAWA, Jinglu HU and Junichi MURATA

Department of Electrical and Electronic Systems Engineering, Kyushu University.
6-10-1 Hakozaki, Higashi-ku, Fukuoka 812-8581, Japan

Abstract

Evolutionary Algorithms are often well-suited for optimization problems. Since the mid-1980's, interest in multiobjective problems has been expanding rapidly. Various evolutionary algorithms have been developed which are capable of searching for multiple solutions concurrently in a single run. In this paper, we proposed a genetic symbiosis algorithm (GSA) for multi-object optimization problems (MOP) based on the symbiotic concept found widely in ecosystem. In the proposed GSA for MOP, a set of symbiotic parameters are introduced to modify the fitness of individuals used for reproduction so as to obtain a variety of Pareto solutions corresponding to user's demands. The symbiotic parameters are trained by minimizing a user defined criterion function. Several numerical simulations are carried out to demonstrate the effectiveness of proposed GSA.

1 Introduction

Many real-world problems involve simultaneous optimization of several incommensurable and often competing objectives. Often, there is no single optimal solution, but rather a set of alternative solutions. These solutions are optimal in the wider sense that no other solutions in the search space are superior to them when all objectives are considered. They are known as Pareto-optimal solutions. A tool exploring the design space for Pareto-optimal solutions is very useful to aid the designer in arriving at a final design.

Since the mid-1980's, there has been a growing interest in solving multicriteria optimization problem using evolutionary approaches. Several algorithms are available which are capable of searching for multiple Pareto-optimal solutions concurrently in a single run. For getting a variety of Pareto-optimal solutions, let us pay attention to ecosystems which hold a very wide diversity. Every species seeks their habitants called niche by adapting themselves to the ever changing environments. And in the niche they are interacted with one another by competing, exploiting and benefiting. These relations are generally called symbiosis. In this paper, we present a genetic algorithm (GA) for multiobjective optimization problems, in which a Genetic Symbiosis Algorithm (GSA) [1] and Pareto-based Ranking method [2] are adopted and it searches for a variety of Pareto-optimal solutions based on the symbiotic concept found in the ecosystems.

GSA has been developed to obtain many kind of solutions for optimization problems, which are optimal in one case or suboptimal in another case depending on the user's requests. In this paper, the GSA is extended to treat not only concentrative optimization problems but also multi-objective optimization problems, where many kinds of Pareto-optimal solutions can be obtained according to the user's requirements.

This paper is organized as follows. Next section introduces key concepts used in the field of evolutionary multicriteria optimization. Basic idea of using the concept of symbiosis to modify the fitness is described in section 3. Section 4 states symbiosis parameters, and section 5 is devoted to summarize the training method RasID [3] mainly used for training symbiosis parameters. Section 6 gives the simulation results of some general multiobjective problems. The last section offers concluding remarks and future perspectives.

2 Multiobjective Problem

2.1 Definitions

A general multi-objective problem is formulated by

$$\begin{aligned} \min\ \mathbf{f}(\mathbf{x}) &= (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_L(\mathbf{x})) \\ \textit{subject to}\quad \mathbf{x} &= (x_1, x_2, \ldots, x_M) \in X \end{aligned} \tag{1}$$

where $\mathbf{f}(\mathbf{x})$ is multi-objective function vector, $f_l(x)$ is lth objective function, $\mathbf{x}$ is input vector, L is the number of objective functions, M is the dimension of input space.

2.2 Pareto Optimality

The scalar concept of optimality does not apply directly in the multiobjective setting. A useful replacement is the notion of Pareto optimality. Essentially, a input vector $\mathbf{x}^* \in X$ is said to be Pareto optimal for (MOP) if all other vectors $\mathbf{x} \in X$ have a higher value for at least one of the objective functions $f_l(*)$, or else have the same value for all objectives. Formally speaking, we have the following definition:

A point $\mathbf{x}^* \in X$ is said to be (globally) Pareto optimal or a (globally) efficient solution or a non-dominated or

0-7803-6273-X/00/$10.00©2000 IEEE

a non-inferior point for (MOP) if and only if there is no $\mathbf{x} \in X$ such that $f_l(\mathbf{x}) \leq f_l(\mathbf{x}^*)$ for all $l \in \{1, 2, \ldots, L\}$, with at least one strict inequality.

Pareto optimal points are also known as efficient, non-dominated, or non-inferior points.

2.3 Multiobjective Ranking Methods

With a single objective problem, according to fitness, the genetic algorithm can allocate that the fitter members of the population have higher chance of producing more offspring than the less fit members. However, with multiobjective problems, every solution has a number of fitness values, one for each objective. This presents a problem in judging the overall fitness of the solutions.

Most researchers of genetic algorithm are favor to the problem of ranking populations. Pareto-based ranking was first proposed in [4]. The fitness of the separate objectives are treated independently and never combined, with only the value for the same objective in different solutions being directly compared. Solutions are ranked into 'non-dominated' order, in which the fittest is the solutions dominated the least by others.

However, various evolutionary algorithms can only be capable of searching for multiple solutions in a single run because of Genetic Drift. In the next subsection, we give some explanation about the effect of Genetic Drift in solving multicriteria optimization problems.

2.4 Genetic Drift

Genetic drift is a term borrowed from population genetics where it is used to explain changes in gene frequency through random sampling of the population. It is a phenomenon observed in GA due to the stochastic nature of the selection operator, and is one of the mechanisms by which the population converges to a single member.

For various evolutionary algorithms for MOP, we can analyse them as two steps. The first step is when not all points $\mathbf{x}$ are Pareto optimal points and the second step is when all points $\mathbf{x}$ are Pareto optimal points. Let us pay attention to the second step, all points are non-dominated or non-inferior, so each point has a same chance of producing offspring and we can consider the offsprings of them are random selected. If we decouple the effect of mutation and crossover, we can see the effect of Genetic Drift directly.

Alex Rogers and Adam Prügel-Bennett [5] have pointed out that the convergence of population is due to two factors. One factor is selection pressure producing multiple copies of fitter population members while the other factor is independent of population member fitness and is due to the stochastic nature of the selection operator-genetic drift. And the method of calculating the convergence rate of genetic drift in terms of changing population fitness variance has been developed in the paper [5]. Using this method, it is obviously shown that all points converge to a single run in the second step and the rate of convergence is determined by the selection schemes.

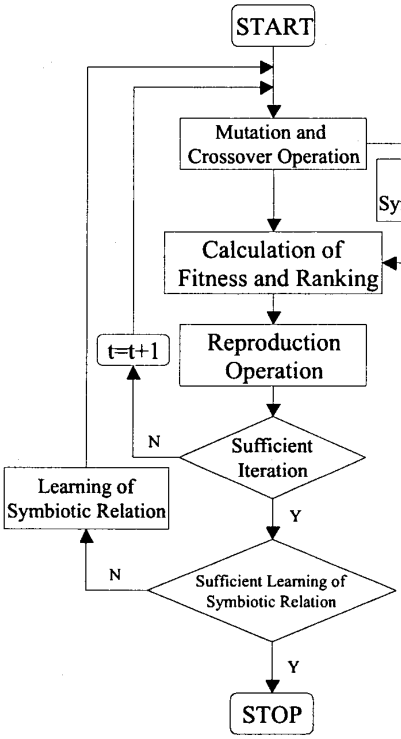
START
Mutation and Crossover Operation
Calculation of Symbiotic Parameter
Calculation of Fitness and Ranking
t=t+1
Reproduction Operation
N
Sufficient Iteration
Y
Learning of Symbiotic Relation
N
Sufficient Learning of Symbiotic Relation
Y
STOP

Fig. 1. Structure of GSA for MOP.

3 Basic Structure of GSA for MOP

As presented in the previous section, all Pareto solutions are non-dominated, so the selection pressure equals 0 in the second step. However paying attention to ecosystems, we can find that the selection pressure can be produced not only some objective conditions such as temperature, food, but also some interactions among species and some interactions among objective conditions. So in this paper, the symbiosis parameter θ_{lij} and θ_{ml} are introduced. The symbiosis parameter θ_{lij} represents the symbiosis relations which is developed in [1], such as competition, exploitation and benefiting between the individuals and the symbiosis parameter θ_{ml} describes the interactions among objective functions. These parameters can change the selection pressure and hold Pareto solutions in a wide diversity.

As described in Fig.1, GSA for MOP is structured in such a way that there is an outer loop for training symbiosis relations besides the main GA loop which includes the calculation of symbiosis parameters.

In the Main GA loop, we use symbiosis parameters θ_{lij} and θ_{ml} to modify the fitness and use the modified fitness to calculate rank of each individual. Reproduc-

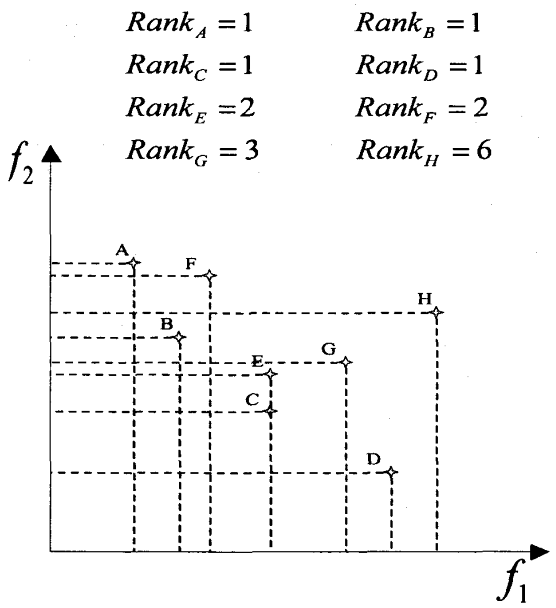

Fig. 2. Pareto-based ranking.

tion is done as in the ordinary GA after diversification processing such as mutation and crossover. In the outer loop, the symbiosis relations are trained by using a modified random search method called RasID [3]. In this paper, optimization of real coded functions is considered.

3.1 Ranking Population

The rank of each individual is calculated based on the modified fitness of individuals. The ranking method presented in [2] is used for this calculation; if individual i dominates n other individuals, then the rank of individual i is $1+n$. Figure 2 shows an example of the Pareto-based ranking.

3.2 Fitness Modification

We use symbiosis parameters θ_{lij} and θ_{ml} to modify the fitness $(1,\ldots,l,\ldots,L)$ of each individual as follows.

$$f_l^s(x_j) = [\sum_{m\in L} \theta_{ml}(x_j) + \sum_{i\in N} \theta_{lij}(x_j)] f_l(x_j) \tag{2}$$

where x_j is phenotype or genotype of individual j, $f_l(x_j)$ is fitness of individual j for function l before modification, $f_l^s(x_j)$ is fitness of individual j for function l after modification, $\theta_{ml}(x_j)$ is function symbiosis parameter from function m to function l, $\theta_{lij}(x_j)$ is genetic symbiosis parameter for function l from individual i to individual j, N is the Number of individuals, L is the number of functions.

4 Symbiosis Parameters

4.1 Symbiosis Parameter θ_{ml}

As stated in the previous section, symbiosis parameter θ_{ml} describes the interactions among objective functions. We can calculate it from distance $f_{ml}(x_j)$ of $f_l(x_j)$ and $f_m(x_j)$ for each individual j by using fuzzy inference as follows.

$$\text{If } \bar f_{ml}(x_j) \text{ is } F_q \text{ then } \theta_{ml} \text{ is } \Theta_q \quad m,l \in L \text{ and } q \in Q \tag{3}$$

Membership functions of fuzzy set F_q and Θ_q are as follows.

$$f_{fq}(\bar f_{ml}) = e^{-\frac{(\bar f_{ml}-\mu_{fq})^2}{2\sigma_{fq}^2}} \tag{4}$$

$$f_{\theta q}(\theta_{ml}) = e^{-\frac{(\theta_{ml}-\mu_{\theta q})^2}{2\sigma_{\theta q}^2}} \tag{5}$$

where

$$f_{ml} = f_m(x_j) - f_l(x_j)$$
$$\bar f_{ml} = \frac{f_{ml}}{|f_{\max} - f_{\min}|}$$
$$f_{\min} \le f_m(x_j), f_l(x_j) \le f_{\max}$$

Q is the number of fuzzy rules

In the fuzzy inference, Gaussian membership functions are adopted for *if* part and *then* part of the fuzzy rules, and θ_{ml} can be calculated by (3)-(5) using the following center-of-gravity method.

$$\theta_{ml} = \frac{\sum_{q\in Q} f_{fq}(\bar f_{ml})\, \mu_{\theta q}\, \sigma_{\theta q}}{\sum_{q\in Q} f_{fq}(\bar f_{ml})\, \sigma_{\theta q}} \tag{6}$$

4.2 Symbiosis Parameter θ_{lij}

Since symbiosis θ_{lij} describes the interactions among individuals, we can calculate it from distance r_{ij} of x_i and x_j and distance f_{lij} of $f_l(x_i)$ and $f_l(x_j)$ by using fuzzy inference as following rules.

$$\text{If } r_{ij} \text{ is } R_{lq} \text{ and } \bar f_{lij} \text{ is } F_{lq} \text{ then } \theta_{lij} \text{ is } \theta_{lq} \quad i,j \in N \text{ and } q \in Q \tag{7}$$

Membership functions of fuzzy set R_{lq}, F_{lq} and Θ_{lq} are as follows.

$$f_{rlq}(\bar r_{ij}) = e^{-\frac{(\bar r_{ij}-\mu_{rlq})^2}{2\sigma_{rlq}^2}} \tag{8}$$

$$f_{flq}(\bar f_{lij}) = e^{-\frac{(\bar f_{lij}-\mu_{flq})^2}{2\sigma_{flq}^2}} \tag{9}$$

$$f_{\theta lq}(\theta_{lij}) = e^{-\frac{(\theta_{lij}-\mu_{\theta lq})^2}{2\sigma_{\theta lq}^2}} \tag{10}$$

where

$$r_{ij} = \sqrt{\sum_{m \in M} (x_{im} - x_{jm})}$$
$$\bar{r}_{ij} = \frac{r_{ij}}{\sqrt{M(x_{\max} - x_{\min})}}$$
$$f_{lij} = f_l(x_i) - f_l(x_j)$$
$$\bar{f}_{lij} = \frac{f_{lij}}{|f_{\max} - f_{\min}|}$$
$$x_{\min} \leq x_{im}, x_{jm} \leq x_{\max}$$
$$f_{\min} \leq f_l(x_i), f_l(x_j) \leq f_{\max}$$

Q is the number of fuzzy rules

M is the dimension of input space

Symbiosis parameter θ_{lij} can be calculated by (7)-(10) using the following center-of-gravity method.

$$\theta_{lij} = \frac{\sum_{q \in Q} \{f_{rlq}(\bar{r}_{ij}) \cap f_{flq}(\bar{f}_{lij})\}\, \mu_{\theta lq}\, \sigma_{\theta lq}}{\sum_{q \in Q} \{f_{rlq}(\bar{r}_{ij}) \cap f_{flq}(\bar{f}_{lij})\}\, \sigma_{\theta lq}} \qquad (11)$$

5 Learning of Symbiosis Relations

From the previous section, it has become clear that symbiosis parameter can be calculated by the inference rules that have many membership function (MF) parameters like μ_{fq}, σ_{fq}; $\mu_{\theta q}$, $\sigma_{\theta q}$; μ_{rlq}, σ_{rlq}; μ_{flq}, σ_{flq}, $\mu_{\theta lq}$ and $\sigma_{\theta lq}$. Learning of symbiosis relations here means that MF parameters are trained to meet the designer's requirements on the distribution of the individuals.

5.1 Criterion Function

User's requirements can be described as criterion functions. For example, we can give a criterion function as follows.

$$\begin{aligned} L \;=\; & k_R \sum_{j \in N} R(x_j) \qquad (12) \\ & + k_{Ef} e^{-\alpha E_f} + k_{Vf} \sum_{j \in N} (D_{mf}(x_j) - E_f)^2 \\ & + k_{Er} e^{-\beta E_r} + k_{Vr} \sum_{j \in N} (D_{mr}(x_j) - E_r)^2 \end{aligned}$$

where

$$D_f(ij) = \sqrt{\sum_{l \in L} (f_l(x_i) - f_l(x_j))^2}$$
$$D_{mf}(x_j) = \min\{D_f(ij)\} \quad i \in N$$
$$E_f = \frac{1}{N} \sum_{j \in N} D_{mf}(x_j)$$
$$D_r(ij) = r_{ij}$$
$$D_{mr}(x_j) = \min\{D_r(ij)\} \quad i \in N$$
$$E_r = \frac{1}{N} \sum_{j \in N} D_{mr}(x_j)$$

where $R(x_j)$ is the rank of individual j. k_R, k_{Ef}, k_{Vf}, k_{Er}, k_{Vr}, α and β are the weighting factor. $D_f(ij)$ and $D_r(ij)$ are the distance in the gnome and the fitness space for individual j. $D_{mf}(x_j)$ and $D_{mr}(x_j)$ are the minimum distance for individual j.

In this criterion, we can consider the individual i which is the nearest to individual j in the genome or the fitness space as the neighbor of individual j in the gnome or the fitness space. The expectation of $D_{mf}(x_j)$ and $D_{mr}(x_j)$ is trained to maximum and the variance of $D_{mf}(x_j)$ and $D_{mr}(x_j)$ is trained to minimum. It can make the solutions well-distributed. On the other hand, minimizing the sum of rank for each individual can get Pareto solutions. Of course, it is the most important condition, so we should make the weighting factor k_R larger than others. The weighting factor k_{Ef}, k_{Vf}, k_{Er}, k_{Vr} can be changed by user's requirements. So this criterion function means searching a variety of Pareto solutions and make them well-distributed in Pareto space.

5.2 Learning Algorithm: RasID

The gradient-based method is often used to train parameters, but the gradient information cannot be available in the proposed model. In this paper, we train the MF parameters by using a modified random search method called RasID (Random Search with Intensification and Diversification)

RasID(Random Search with Intensification and Diversification) is a kind of random search optimization methods and executes intensified and diversified search in a unified manner using information on success and failure of the past searching. Basic idea of RasID is that it continues to iterate the searching in the following way. When there is quite a possibility of finding a better solution around the current one, intensified search is executed near the current solution, and when finding a better solution can not be expected because of falling into local minima, then diversified search is executed looking for a better solution. One of the distinguished features of RasID is that the probability density function for random searching is changed adaptably depending on the success and failure information of the past searching. So, by using RasID, gradient calculation is not needed, faster computation is expected and escaping from local minima becomes possible.

5.3 Learning Procedure

Summarizing the learning procedure, MF parameters are trained in order to minimize the criterion function L obtained by execution the main GA procedure, that is, the inner loop of GSA for MOP in Fig. 1. In other words, first the main GA loop procedure is carried out with MF parameters being fixed until the solution is obtained. Then, MF parameters μ_* and σ_* are trained to minimize the criterion function L. And, the above inner loop and outer loop calculations are done iteratively

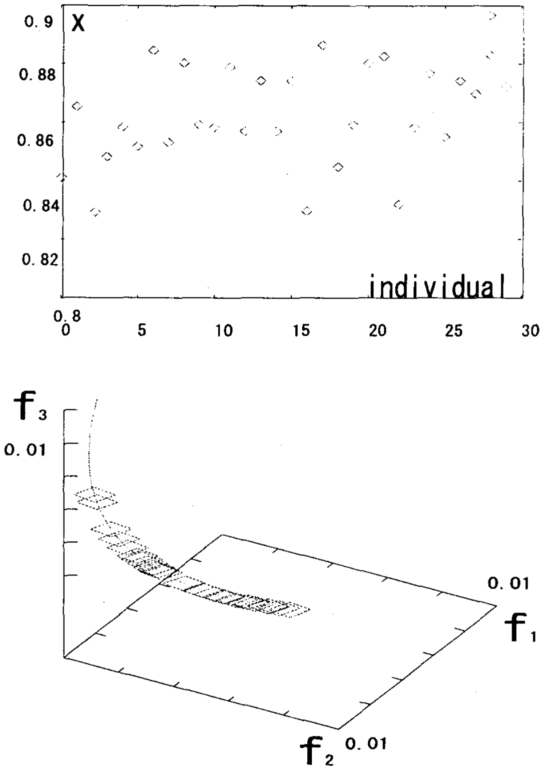

Fig. 3. GSA for MOP simulation 1.

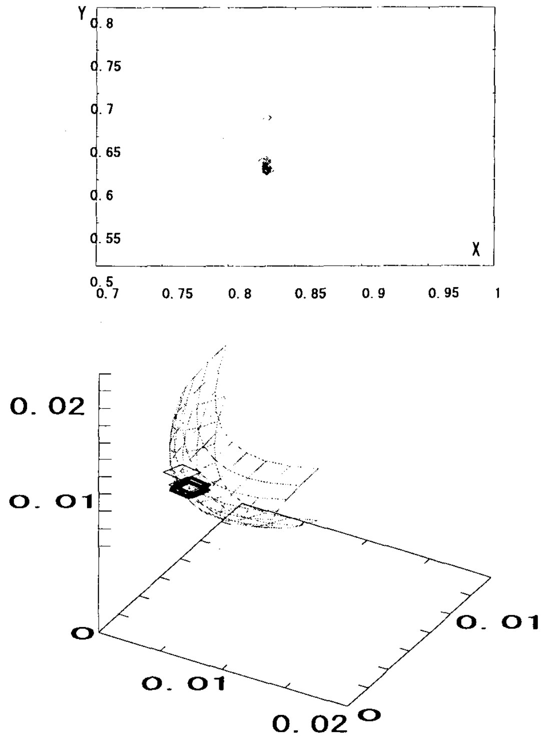

Fig. 4. GSA for MOP simulation 2.

until sufficient symbiosis relations are obtained.

6 Simulations

6.1 Simulation 1

In simulation 1, a very simple test function for multiobjective optimizers is defined as follows

$$\begin{aligned} \text{min} \quad & (f_1(x), f_2(x), f_3(x)) \\ \text{where} \quad & f_1(x) = (x - 0.8)^2 \\ & f_2(x) = (x - 0.85)^2 \\ & f_3(x) = (x - 0.9)^2 \\ & x \in [0, 1] \end{aligned}$$

Obviously, the Pareto-optimal points are located in the range $x \in [0.8, 0.9]$. In this simulation, the number of individuals N is 30, the number of generations is 500, and the number iterations for RasID training MF parameters is 1000. Mutation rate is 0.05, and as for crossover, Michalewicz crossover [6] is used because of the real value coding and $k_R = 10$ and $k_{Ef} = k_{Vf} = k_{Er} = k_{Vr} = 1$ and $\alpha = \beta = -50$.

The results are shown in Fig.3. It can be observed that GSA for MOP is able to well approximate the Pareto-optimal front.

6.2 Simulation 2

A two-dimension test function for multiobjective optimizers is defined as follows.

$$\begin{aligned} \text{min} \quad & (f_1(x,y), f_2(x,y), f_3(x,y)) \\ \text{where} \quad & f_1(x,y) = (x - 0.8)^2 + (y - 0.6)^2 \\ & f_2(x,y) = (x - 0.85)^2 + (y - 0.7)^2 \\ & f_3(x,y) = (x - 0.9)^2 + (y - 0.6)^2 \\ & x, y \in [0, 1] \end{aligned}$$

In simulation 2, the number of individuals N is 30, the number of generations is 500 and the number of iterations for RasID training MF parameters is 1000. Mutation rate is 0.05, and as for crossover, Michalewicz crossover [6] is used because of the real value coding and $k_R = 1$ and $k_{Ef} = k_{Vf} = k_{Er} = k_{Vr} = 0$. It means we don't consider the distribution of solutions.

The results are shown in Fig.4. It is clear that solutions are searched in a single run as the conventional evolutionary algorithms.

6.3 Simulation 3

Simulation 3 uses the same two-dimension test function and initial conditions with simulation 2 except $k_{Ef} = k_{Vf} = k_{Er} = k_{Vr} = 0.1$ and $\alpha = \beta = -50$. It means we consider the distribution of solutions.

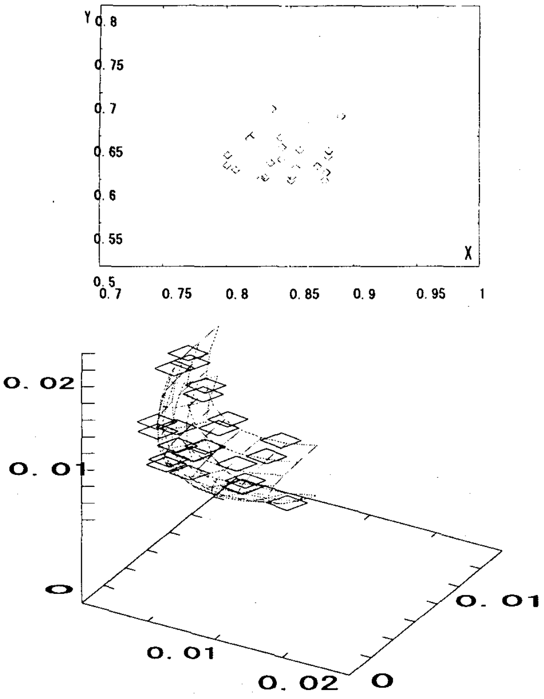

Fig. 5. GSA for MOP simulation 3.

The results are shown in Fig.5. It can be observed that GSA for MOP is able to well approximate the Pareto-optimal front according to the criterion function. In other words, the results are what we expect to get.

7 Conclusion

In this paper, GSA for MOP which is based on the concept of symbiosis found widely in ecosystems is presented. It can approximate the Pareto-optimal front to user's requirement.

Distinguished points of GSA for MOP are that (1) calculation of ranking is executed using the modified fitness $f_l^s(x_j)$ instead of the original fitness $f_l(x_j)$, (2) symbiosis parameter is calculated by fuzzy inference, and (3) fuzzy parameter can be trained by RasID learning to realized the required distribution in the gnome and the fitness space.

References

[1] K. Hirasawa, Y. Ishikawa, J. Hu, C. Jin, J. Murata, "Genetic Symbiosis Algorithm," *Proc. Congress on Evolutionary Computation*, (San Diego, USA), July 2000.

[2] 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.

[3] J. Hu, K. Hirasawa, J. Murata, "RasID—A Random Search Method for Neural Networks Training," *Journal of Advanced Computational Intelligence*, vol. 2, no. 4, pp.134-141, 1998

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

[5] Alex Rogers, Adam Prügel-Bennett, "Genetic Drift in Genetic Algorithm Selection Schemes," *IEEE Trans. Evol. Comput.*, vol. 3, no. 4. pp298-303, Nov 1999.

[6] Z. Michalewicz *Genetic Algorithm + Data Structures = Evolution Programs*, Second, extended edition, Springer 1994