Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Multicriteria Decision Making (MCDM 2007)

Combining Aspiration Level Methods in Multi-objective Programming and Sequential Approximate Optimization using Computational Intelligence

Hirotaka Nakayama, *Member, IEEE*, and Yeboon Yun, *Member, IEEE*

***Abstract*— Since Pareto optimal solutions in multi-objective optimization are not unique but makes a set, decision maker (DM) needs to select one of them as a final decision. In this event, DM tries to find a solution making a well balance among multiple objectives. Aspiration level methods support DM to do this in an interactive way, and are very simple, easy and intuitive for DMs. Their effectiveness has been observed through various fields of practical problems. One of authors proposed the satisficing trade-off method early in '80s, and applied it to several kinds of practical problems.**

On the other hand, in many engineering design problems, the explicit form of objective function can not be given in terms of design variables. Given the value of design variables, under this circumstance, the value of objective function is obtained by some simulation analysis or experiments. Usually, these analyses are computationary expensive. In order to make the number of analyses as few as possible, several methods for sequential approximate optimization which make optimization in parallel with model prediction has been proposed.

In this paper, we form a coalition between aspiration level methods and sequential approximate optimization methods in order to get a final solution for multi-objective engineering problems in a reasonable number of analyses. In particular, we apply $\mu - \nu-$SVM which was developed by the authors recently on the basis of goal programming. The effectiveness of the proposed method will be shown through some numerical experiments.

I. Introduction

In multi-objective optimization, no solution optimizing all objective functions simultaneously exists in general. Instead, the notion of Pareto optimal solutions, which are "efficient" in terms of all objective functions, are introduced. Usually the Pareto optimal solution is not unique, but makes a set. As a consequence, how to decide a final solution among Pareto optimal solutions becomes our issue. In many practical problems, we need to select one Pareto solution taking into account the balance among those multiple objective functions, which is called "trade-off analysis". It is no exaggeration to say that the most important task in multi-objective optimization is trade-off analysis.

In cases with two or three objective functions, the set of Pareto optimal solutions in the objective function space (i.e., Pareto frontier) can be depicted relatively easily. Seeing Pareto frontiers, we can grasp the trade-off relation among objectives totally. Therefore, it would be the best way to depict Pareto frontiers in cases with two or three objectives. (It might be difficult to read the trade-off relation among objectives with three dimension, though). In cases with more than three objectives, however, it is impossible to depict Pareto frontier. Under this circumstance, interactive methods can help us to make local trade-off analysis showing a "certain" Pareto optimal solution. In particular, the aspiration level methods have been observed to be effective in many practical problems because the aspiration level is very simple, easy and intuitive for DMs. One of authors proposed the satisficing trade-off method early in '80s, and applied it to several kinds of practical problems.

Unfortunately, however, in many engineering design problems, the explicit form of objective function can not be given in terms of design variables. Given the value of design variables, under this circumstance, the value of objective function is obtained by some analysis such as structural analysis, fluidmechanic analysis, thermodynamic analysis, and so on. Usually, these analyses are computationary expensive. In order to make the number of analyses as few as possible, the authors proposed a sequential approximate optimization method using computational intelligence such as radial basis function networks (RBFN) and support vector machines (SVM). In this approach, optimization is performed in parallel with predicting the form of objective function.

In the following, we form a coalition between aspiration level methods and sequential approximate optimization methods in order to get a final solution for multi-objective engineering problems in a reasonable number of analyses. In particular, we apply $\mu - \nu-$SVM which was developed by the authors recently on the basis of goal programming. Finally, the effectiveness of the proposed method will be shown through some numerical experiments.

II. Aspiration Level Methods for Interactive Multi-objective Programming

Since there may be many Pareto solutions in practice, the final decision should be made among them taking the total balance over all criteria into account. This is a problem of value judgment of DM. The totally balancing over criteria is usually called *trade-off*. Interactive multi-objective programming searches a solution in an interactive way with DM while making trade-off analysis on the basis of DM's value

Hirotaka Nakayama is with the Department of Information Science and Systems Engineering, Konan University, Kobe 658-8501, Japan (phone: +81-78-435-2534; fax: 81-78-435-2540; email: nakayama@konan-u.ac.jp).

Yeboon Yun is with the Department of Reliability-based Information Systems Engineering, Kagawa University, Takamatsu 761-0396, Japan (email: yun@eng.kagawa-u.ac.jp).

This paper is to be presented at IEEE Symposium on Computational Intelligence in Multicriteria Decision Making, Hawaii, 2007.

1-4244-0702-8/07/$20.00 ©2007 IEEE

judgment. Among them, the aspiration level approach is now recognized to be effective in practice, because

(i) it does not require any consistency of DM's judgment,
(ii) aspiration levels reflect the wish of DM very well,
(iii) aspiration levels play the role of probe better than the weight for objective functions.

As one of aspiration level approaches, one of authors proposed the satisficing trade-off method [11]. Suppose that we have objective functions $\boldsymbol{f}(\boldsymbol{x}) := (f_1(\boldsymbol{x}), \dots, f_r(\boldsymbol{x}))$ to be minimized over $\boldsymbol{x} \in X \subset R^n$. In the satisficing trade-off method, the aspiration level at the k-th iteration $\overline{\boldsymbol{f}}^k$ is modified as follows:

$$\overline{\boldsymbol{f}}^{k+1} = T \circ P(\overline{\boldsymbol{f}}^k).$$

Here, the operator P selects the Pareto solution nearest in some sense to the given aspiration level $\overline{\boldsymbol{f}}^k$. The operator T is the trade-off operator which changes the k-th aspiration level $\overline{\boldsymbol{f}}^k$ if DM does not compromise with the shown solution $P(\overline{\boldsymbol{f}}^k)$. Of course, since $P(\overline{\boldsymbol{f}}^k)$ is a Pareto solution, there exists no feasible solution which makes all criteria better than $P(\overline{\boldsymbol{f}}^k)$, and thus DM has to trade-off among criteria if he wants to improve some of criteria. Based on this trade-off, a new aspiration level is decided as $T \circ P(\overline{\boldsymbol{f}}^k)$. Similar process is continued until DM obtains an agreeable solution.

The operation which gives a Pareto solution $P(\overline{\boldsymbol{f}}^k)$ nearest to $\overline{\boldsymbol{f}}^k$ is performed by some auxiliary scalar optimization. It has been shown in Sawaragi-Nakayama-Tanino (1985) that the only one scalarization technique, which provides any Pareto solution regardless of the structure of problem, is of the Tchebyshev norm type. However, the scalarization function of Tchebyshev norm type yields not only a Pareto solution but also a weak Pareto solution. Since weak Pareto solutions have a possibility that there may be another solution which improves a criterion while others being fixed, they are not necessarily "*efficient*" as a solution in decision making. In order to exclude weak Pareto solutions, the following scalarization function of the augmented Tchebyshev type can be used:

$$\max_{1 \leqq i \leqq r} \omega_i \left(f_i(\boldsymbol{x}) - \overline{f}_i \right) + \alpha \sum_{i=1}^{r} \omega_i f_i(\boldsymbol{x}), \tag{1}$$

where α is usually set a sufficiently small positive number, say 10^{-6}.

The weight ω_i is usually given as follows: Let f_i^* be an ideal value which is usually given in such a way that $f_i^* < \min \{f_i(\boldsymbol{x}) \mid \boldsymbol{x} \in X\}$. For this circumstance, we set

$$\omega_i^k = \frac{1}{\overline{f}_i^k - f_i^*} \tag{2}$$

In cases that DM is not satisfied with the solution for $P(\overline{\boldsymbol{f}}^k)$, he/she is requested to answer his/her new aspiration level $\overline{\boldsymbol{f}}^{k+1}$. Let $\boldsymbol{x}^k$ denote the Pareto solution obtained by projection $P(\overline{\boldsymbol{f}}^k)$, and classify the objective functions into the following three groups:

(i) the class of criteria which are to be improved more,

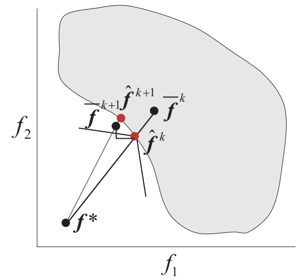


Fig. 1. Satisficing Trade-off Method

(ii) the class of criteria which may be relaxed,
(iii) the class of criteria which are acceptable as they are.

Let the index set of each class be denoted by I_I^k, I_R^k, I_A^k, respectively. Clearly, $\overline{f}_i^{k+1} < f_i(x^k)$ for all $i \in I_I^k$. Usually, for $i \in I_A^k$, we set $\overline{f}_i^{k+1} = f_i(\boldsymbol{x}^k)$. For $i \in I_R^k$, DM has to agree to increase the value of $\overline{f}_i^{k+1}$. It should be noted that an appropriate sacrifice of f_j for $j \in I_R^k$ is needed for attaining the improvement of f_i for $i \in I_I^k$.

In cases with a large number of objective functions, it is laborious for DM to answer his/her new aspiration level for all of objective functions. To overcome this difficulty, one of authors developed the automatic trade-off analysis using sensitivity analysis in usual mathematical programming [13], and in addition the exact trade-off method for linear or quadratic programming using parametric optimization techniques [12].

III. Sequential Approximate Optimization using Computational Intelligence

In many engineering design problems, objective functions are "black-box", whose forms are not explicitly known in terms of design variables, but whose values are given by sampled real/computational experiments. Usually, these real/computational experiments are considerably expensive. Therefore, if these optimization problems are solved by existing methods, it takes an unrealistic order of time to obtain a solution. For this situation, the number of necessary sampled experiments should be as few as possible.

Recently, the authors proposed to apply machine learning techniques such as RBF (Radial Basis Function) networks and Support Vector Machines (SVM) for approximating the black-box function [9], [10]. There, how to select additional sample points is a main issue in order to make a good approximation with as few sample points as possible.

A. Selection of Additional Samples

If the current solution is not satisfactory, namely if our stopping condition is not satisfied, we need some additional samples in order to improve the approximation of the black-box objective function. Now, how to select such additional samples becomes an important issue. to this end, there have been developed several methods such as design of experiments, kriging method, maximal entropy, active learning and so on (see, for example, Jin [8]).

It is important to get well balanced samples providing both global information and local information on black-box objective functions. The author and his coresearchers suggested a method which gives both global information for predicting the objective function and local information near the optimal point at the same time [9]. Namely, two kinds of additional samples are taken at the same time for relearning the form of the objective function. One of them is selected inside a neighborhood of the current optimal point in order to add local information near the (estimated) optimal point. The size of this neighborhood is controlled during the convergence process. The other one is selected far away from the current optimal value in order to give a better prediction of the form of the objective function. The former additional sample gives more detailed information near the current optimal point. The latter sample prevents converging to local maximum (or minimum) points.

The algorithm is summarized as follows:

Step 1: Predict the form of the objective function by a certain computational intelligence technique, e.g., RBFN or SVM on the basis of the given training data.

Step 2: Estimate an optimal point for the predicted objective function by some optimization techniques, e.g., GA.

Step 3: Terminate the iteration, if the optimal value for the predicted objective function, $\hat{\boldsymbol{f}}^{*k}$, and the optimal value among the learning samples so far, $\tilde{\boldsymbol{f}}^{*k}$, enjoy the following convergence condition with a given $\epsilon > 0$

$$\left\| \hat{\boldsymbol{f}}^{*k} - \tilde{\boldsymbol{f}}^{*k} \right\| < \epsilon.$$

Otherwise go to the next step.

Step 5: Select an additional sample near the current optimal value, i.e., inside S.

Step 6: Select another additional sample outside S in a place in which the density of the training data is low as stated above.

Step 7: Go to Step.1.

B. Application of $\mu-\nu-$SVM

SVM was originally developed for pattern classification and later extended to regression (Vapnik *et al.* [3], [19], Cristianini and Shawe-Taylor [4], Schölkopf-Smola [18]). In pattern classification problems with two class sets, it generalizes linear classifiers into high dimensional feature spaces through nonlinear mappings defined implicitly by kernels in the Hilbert space so that it may produce nonlinear classifiers in the original data space. Linear classifiers then are optimized to give the maximal margin separation between the classes. Linear classifiers on the basis of goal programming were developed extensively in 1980's [7]. The authors developed several varieties of SVM using multi-objective programming and goal programming (MOP/GP) techniques [14].

In the goal programming approach to linear classifiers, we consider two kinds of deviations: One is the exterior deviation ξ_i which is a deviation from the hyperplane of a point $\boldsymbol{x}_i$ improperly classified; The other one is the interior deviation η_i which is a deviation from the hyperplane of a point $\boldsymbol{x}_i$ properly classified. Several kinds of objective functions are possible in this approach as follows:

i) minimize the maximum exterior deviation (decrease errors as much as possible),

ii) maximize the minimum interior deviation (i.e., maximize the margin),

iii) maximize the weighted sum of interior deviation,

iv) minimize the weighted sum of exterior deviation.

Introducing the objective iv) above leads to the soft margin SVM with slack variables (or, exterior deviations) ξ_i $(i = 1, \ldots, \ell)$ which allow classification errors to some extent.

Taking into account the objectives (ii) and (iv), we can have the same formulation of ν-support vector algorithm developed by Schölkopf et al. [17].

Other variants of SVM considering both slack variables for misclassified data points (i.e., exterior deviations) and surplus variables for correctly classified data points (i.e., interior deviations) are possible: Total Margin Algorithm considering iii) and v), $\mu-$SVM considering the objectives i) and iii), $\mu-\nu-$SVM considering the objectives i) and ii) [14]. The authors extended those SVMs to regression [15]. Among them, $\mu-\nu-$SVR has been observed to provide a good sparse approximation. In this paper, therefore, we apply $\mu-\nu-$SVR to the sequential approximate optimization.

The primal formulation of $\mu-\nu-$SVR is given by

$$\begin{aligned}
\underset{\boldsymbol{w},b,\varepsilon,\xi,\acute{\xi}}{\text{minimize}} \quad & \frac{1}{2}\|\boldsymbol{w}\|_2^2 + \nu\varepsilon + \mu(\xi + \acute{\xi}) \\
\text{subject to} \quad & \left(\boldsymbol{w}^T \boldsymbol{z}_i + b\right) - y_i \leqq \varepsilon + \xi, \;\; i = 1, \ldots, \ell, \\
& y_i - \left(\boldsymbol{w}^T \boldsymbol{z}_i + b\right) \leqq \varepsilon + \acute{\xi}, \;\; i = 1, \ldots, \ell, \\
& \varepsilon, \;\; \xi, \;\; \acute{\xi} \geqq 0,
\end{aligned}$$

where ν and μ are trade-off parameters between the norm of $\boldsymbol{w}$ and ε and ξ $(\acute{\xi})$.

Applying the Lagrange duality theory, we obtain the following dual formulation of $\mu-\nu-$SVR:

$$\begin{aligned}
\underset{\alpha_j,\acute{\alpha}_i}{\text{maximize}} \quad & -\frac{1}{2}\sum_{i,j=1}^{\ell} \left(\acute{\alpha}_i - \alpha_i\right)\left(\acute{\alpha}_j - \alpha_j\right) K\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) \\
& + \sum_{i=1}^{\ell} \left(\acute{\alpha}_i - \alpha_i\right) y_i \\
\text{subject to} \quad & \sum_{i=1}^{\ell} \left(\acute{\alpha}_i - \alpha_i\right) = 0, \\
& \sum_{i=1}^{\ell} \acute{\alpha}_i \leqq \mu, \;\; \sum_{i=1}^{\ell} \alpha_i \leqq \mu, \\
& \sum_{i=1}^{\ell} \left(\acute{\alpha}_i + \alpha_i\right) \leqq \nu, \\
& \acute{\alpha}_i \geqq 0, \;\; \alpha_i \geqq 0, \;\; i = 1, \ldots, \ell.
\end{aligned}$$

IV. Combining Satisficing trade-off Method and Sequential Approximate Optimization

Now, we propose a method combining the satisficing trade-off method for interactive multi-objective programming and the sequential approximate optimization using μ−ν−SVR. The procedure is summarized as follows:

Step 1. (Real Evaluation)
Evaluate actually the values of objective functions $\boldsymbol{f}(\boldsymbol{x}_1), \boldsymbol{f}(\boldsymbol{x}_2), \ldots, \boldsymbol{f}(\boldsymbol{x}_\ell)$ for sampled data $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_\ell$ through computational simulation analysis or experiments.

Step 2. (Approximation)
Approximate each objective function $\hat{f}_1(\boldsymbol{x}), \ldots, \hat{f}_m(\boldsymbol{x})$ by the learning of $\mu - \nu -$SVR on the basis of real sample data set.

Step 3. (Find a Pareto Solution Nearest to the Aspiration Level and Generate Pareto Frontier)
Find a Pareto optimal solution nearest to the given aspiration level for the approximated objective functions $\hat{\boldsymbol{f}}(\boldsymbol{x}) := (\hat{f}_1(\boldsymbol{x}), \ldots, \hat{f}_m(\boldsymbol{x}))$. This is performed by using GA for minimizing the augmented Tchebyshev scalarization function (1). In addition, generate Pareto frontier by MOGA for accumulated individuals during the procedure for optimizing the augmented Tchebyshev scalarization function.

Step 4. (Choice of Additional Learning Data)
Choose the additional ℓ_0-data from the set of obtained Pareto optimal solutions. Setting $\ell \leftarrow \ell + \ell_0$, go to Step 1.

how to choose the additional data

Stage 0. First, add the point with highest achievement for the scalarized objective function (1) obtained in Step 3.

Stage 1. Evaluate the ranks for the real sampled data of Step 1 by the ranking method [6].

Stage 2. Approximate the rank function associated with the ranks calculated in the Stage 1 by $\mu - \nu -$SVR.

Stage 3. Calculate the fitness for Pareto optimal solutions obtained in Step 3 on the basis of the obtained approximate rank function.

Stage 4. Add the point with highest fitness to the set of sample points.

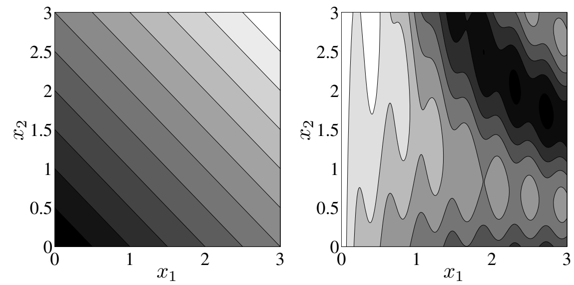
V. Numerical Examples

A. Example 1

First, we consider the following illustrative example (Ex-1):

$$
\begin{aligned}
\text{minimize} \quad & f_1 := x_1 + x_2 \\
\text{minimize} \quad & f_2 := 20\cos(15x_1) + (x_1 - 4)^4 + 100\sin(x_1 x_2) \\
\text{subject to} \quad & 0 \leqq x_1,\ x_2 \leqq 3.
\end{aligned}
$$

The true function of each objective function f_1 and f_2 in the problem (Ex-1) are shown in Fig. 2.



(a) f_1 (b) f_2

Fig. 2. The true contours to the problem

In our simulation, the ideal point and the aspiration level is respectively given by

$$
\begin{aligned}
\left(f_1^*,\ f_2^*\right) &= (0,\ -120), \\
\left(\overline{f}_1,\ \overline{f}_2\right) &= (3,\ 200),
\end{aligned}
$$

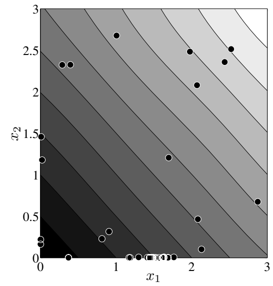
and the closest Pareto solution to the above aspiration level is as follows:

$$
\begin{aligned}
\text{exact optimal solution } \left(\hat{x}_1,\ \hat{x}_2\right) &= (1.41321,\ 0) \\
\text{exact optimal value } \left(\hat{f}_1,\ \hat{f}_2\right) &= (1.41321,\ 30.74221)
\end{aligned}
$$

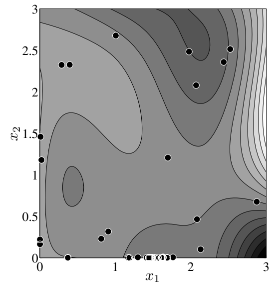
Starting with initial data 10 points randomly, we obtained the following approximate solution by proposed method after 50 real evaluations:

$$
\begin{aligned}
\text{approximate solution } \left(x_1,\ x_2\right) &= (1.45748,\ 0) \\
\text{approximate objective value } \left(f_1,\ f_2\right) &= (1.45748,\ 35.34059)
\end{aligned}
$$

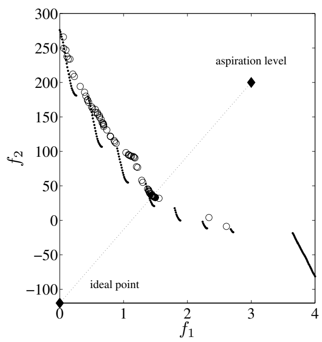
The final result is shown in Fig. 3. It can be seen that we can obtain a reasonable solution minimizing (1) approximately even with rough approximation for each objective function. At this stage, DMs can change the aspiration level according to their wish. If DMs want to improve some of objectives, they have to agree with some sacrifice for other objectives. This is trade-off analysis. For making trade-off analysis, it is more convenient if DMs can know some information on Pareto frontier around the present solution. To this end, approximation of Pareto frontier obtained in Step 3 can be available. One may see the convergence process in Fig. 4. In addition, Table I shows the results for 10 times numerical experiments.



(a) contour of f_1



(b) contour of f_2



(c) Pareto solution nearest to the aspiration level and approximation of Pareto frontier

Fig. 3. Result for 50 samples

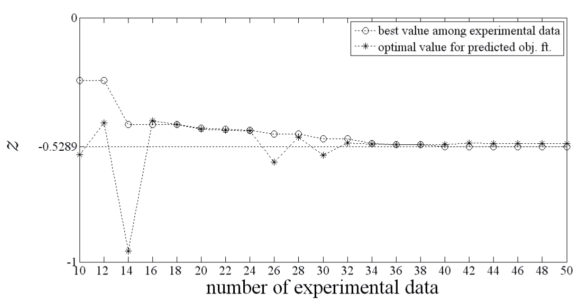


Fig. 4. Convergence process

TABLE I
NUMERICAL RESULTS FOR EXAMPLE 1

	best value among experimental data			
	x_1	x_2	f_1	f_2
best	1.4135	0.0000	1.4135	30.6627
worst	1.3695	0.0000	1.3695	45.4425
average	1.4109	0.0003	1.4111	31.7150
σ	0.0143	0.0009	0.0144	4.6869

	optimal value for predicted obj. ft.			
	x_1	x_2	f_1	f_2
best	1.4252	0.0000	1.4252	32.1449
worst	1.5630	0.0000	1.5630	47.0042
average	1.4519	0.0000	1.4519	34.8111
σ	0.0393	0.0000	0.0393	4.3464

B. Example 2

Next, we consider a welded beam problem shown in Deb [5] (Fig. 5).

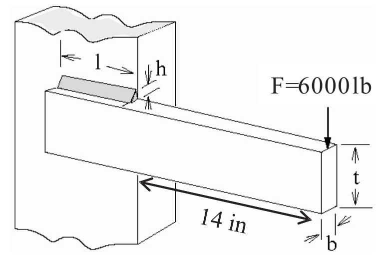


Fig. 5. A welded beam

The mathematical formulation of the problem is given by

$$\begin{aligned}
\min_{h,l,t,b} \quad & f_1 := 1.10471h^2l + 0.04811tb(14+l) \\
\min_{h,l,t,b} \quad & f_2 := \frac{2.1952}{t^3b} \\
\text{s.t.} \quad & g_1 := \tau \leqq 13600 \\
& g_2 := \sigma \leqq 30000 \\
& g_3 := h - b \leqq 0 \\
& g_4 := P_c \geqq 6000 \\
& 0.125 \leqq h,\ b \leqq 5.0,\ \ 0.1 \leqq l,\ t \leqq 10.0
\end{aligned}$$

Here,

$$\begin{aligned}
\tau &= \sqrt{(\tau')^2 + (\tau'')^2 + \frac{l\tau'\tau''}{\sqrt{0.25(l^2+(h+t)^2)}}} \\
\tau' &= \frac{6000}{\sqrt{2}hl} \\
\tau'' &= \frac{6000(14+0.5l)\sqrt{0.25(l^2+(h+t)^2)}}{\sqrt{2}hl\left(\frac{l^2}{12}+0.25(h+t)^2\right)} \\
\sigma &= \frac{504000}{t^2b},\ \ P_c = 64746.022(1-0.0282346t)tb^3
\end{aligned}$$

Set the ideal point and aspiration level as follows:

$$\begin{aligned}
\text{ideal point} &:= (f_1^*, f_2^*) = (0,\ 0) \\
\text{aspiration level} &:= (\overline{f}_1, \overline{f}_2) = (20,\ 0.002)
\end{aligned}$$

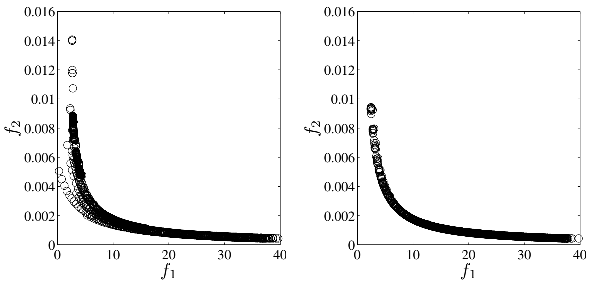
Table II shows the result by using a conventional optimization method (nonlinear constrained optimization in a

TABLE II
RESULT BY CONVENTIONAL OPTIMIZATION METHOD (EX.2)

	h	l	t	b	f_1	f_2
average	1.0834	0.8710	10.0000	1.7685	13.7068	1.25E-03
σ	0.3274	0.1662	5.11E-08	0.1828	1.3793	1.13E-04
max	2.0132	0.9896	10	2.1263	16.3832	1.31E-03
min	0.9221	0.4026	10.0000	1.6818	13.0527	1.03E-03

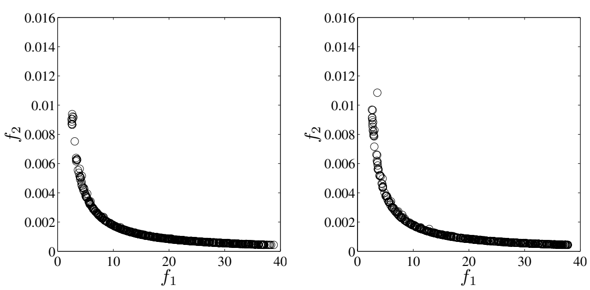
TABLE III
RESULT BY THE PROPOSED METHOD (EX.2)

	h	l	t	b	f_1	f_2
average	0.8921	1.0398	9.9989	1.6809	13.0653	1.31E-03
σ	0.0898	0.1106	0.0012	0.0012	0.0081	7.79E-07
max	1.0787	1.1895	10	1.6824	13.0781	1.31E-03
min	0.7849	0.8273	9.9964	1.6789	13.0531	1.31E-03



(a) 50 samples (b) 100 samples

Fig. 6. Pareto frontier by our proposed method



(a) SPEA2 (b) NSGAII

Fig. 7. Pareto frontier by conventional MOGA (5000 samples)

toolbox of Matlab. It needs 200-300 functional calls. On the other hand, Table III shows the result by our proposed method with 100 samples starting from 50 samples. The numerical experiments were carried out 10 times. Fig. 6 shows an approximate Pareto frontier obtained in Step 3. For comparison, Pareto frontiers generated by conventional MOGA (e.g., SPEA2, NSGAII) with 100 individuals and 50 generations are shown in Fig. 7.

The solution by our proposed method is in a reasonable precision with 100 samples (function calls) in comparison to a conventional optimization method with 200-300 functional calls.

VI. CONCLUDING REMARKS

We proposed a method combining the satisficing trade-off method and a sequential approximate optimization method using computational intelligence for supporting DM to get a final solution. The proposed method provides a Pareto solution closest to the given aspiration level as well as the local trade-off information by an approximate Pareto frontier in the neighborhood of the solution with a reasonable number of simulation analyses. It is promising in practical problems since it has been observed that the method reduces the number of function evaluations up to less than $1/100$ to $1/10$ of usual methods such as MOGAs and usual aspiration level methods through several numerical experiments.

ACKNOWLEDGEMENT

This research was partially supported by JSPS.KAKENHI16510130.

REFERENCES

[1] Coello Coello, C.A., Van Veldhuizen, D.A. and Lamont, G.B. (2002), *Evolutionary Algorithms for Solving Multi-Objective Problems*, Kluwer Academic Publishers.

[2] Deb, K. (2001), *Multi-Objective Optimization using Evolutionary Algorithms*, John & Wiley Sons, Ltd..

[3] Cortes, C. and Vapnik, V., (1995) Support Vector Networks, *Machine Learning*, **20**, pp. 273–297

[4] Cristianini, N. and Shawe-Taylor, J., (2000) *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*, Cambridge University Press

[5] Deb, K. and Srinivasan, A. , (2005) Innovization: Innovative Design Principles Through Optimization, KanGAL Report #2005007

[6] Fonseca, C.M. and Fleming, P.J., (1993) Genetic Algorithms for Multi-objective Optimization: Formulation, Discussion and Generalization, *Proceedings of the Fifth International Conference on Genetic Algorithms*, 416–426

[7] Freed, N. and Glover, F., (1981) Simple but Powerful Goal Programming Models for Discriminant Problems, *European J. of Operational Research*, **7**, 44-60

[8] Jin, Y. (2005), A comprehensive survey of Fitness Approximation in Evolutionary Computation, *Soft Computing*, **9(1)**, 3-12

[9] Nakayama, H., Arakawa, M. and Sasaki, R. (2002), Simulation based Optimization for Unknown Objective Functions, *Optimization and Engineering*, **3**, 201-214

[10] Nakayama, H., Arakawa, M. and Washino, K., (2003), Optimization for Black-box Objective Functions, *Optimization and Optimal Control*, (eds.) P.M. Pardalos, I. Tseveendorj and R. Enkhbat, World Scientific, 185-210

[11] Nakayama, H. and Sawaragi, Y., (1984), Satisficing Trade-off Method for Multi- objective Programming, in M. Grauer and A. Wierzbicki (eds.): *Interactive Decision Analysis*, Springer, 113-122

[12] Nakayama, H. (1992), Trade-off Analysis using Parametric Optimization Techniques, *European J. of Operational Research*, 60, 87-98

[13] Nakayama, H. (1995), Aspiration Level Approach to Interactive Multi-objective Programming and its Applications, P.M. Pardalos, Y. Siskos and C. Zopounidis (eds.), *Advances in Multicriteria Analysis*, Kluwer Academic Publishers, 147-174.

[14] Nakayama, H. and Yun, Y., (2006) Generating Support Vector Machines using Multiobjective Optimization and Goal Programming *Multi-objective Mahine Learning*, Yaochu Jin(ed.), Springer Series on Studies in Computational Intelligence, pp.173-198.

[15] Nakayama, H. and Yun, Y., (2006) Gupport Vector Regression Based on Goal Programming and Multi-objective Programming, IEEE World Congress on Computational Intelligence, CD-ROM (Paper ID : 1536)

[16] Sawaragi, Y., Nakayama, H. and Tanino, T. (1985). *Theory of Multi-objective Optimization*, Academic Press

[17] Schölkopf, B. and Smola, A.J., (1998) New Support Vector Algorithms, *NeuroCOLT2 Technical report Series*, NC2-TR-1998-031

[18] B.Schölkopf, and A.J.Smola, (2002) *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond-*, MIT Press

[19] Vapnik, V.N., (1998) *Statistical Learning Theory*, John Wiley & Sons, New York

[20] Yoon, M., Yun, Y.B. and Nakayama, H., (2003) A Role of Total Margin in Support Vector Machines, *Proc. IJCNN'03*, 2049-2053