The Ninth International Symposium on Operations Research and Its Applications (ISORA'10)
Chengdu-Jiuzhaigou, China, August 19–23, 2010

Sequential Approximate Multiobjective Optimization using Computational Intelligence

Hirotaka Nakayama

Department of Intelligence and Informatics, Konan University, Kobe 658-8501

Abstract In many practical problems, in particular in engineering design, the function form of criteria is not given explicitly in terms of design variables. Given the value of design variables, under this circumstance, the value of objective functions is obtained by real/computational experiments such as structural analysis, fluidmechanic analysis, thermodynamic analysis, and so on. Usually, these experiments are time consuming and expensive. One of recent trends in optimization is how to treat these expensive criteria. In order to make the number of these experiments as few as possible, optimization is performed in parallel with predicting the form of objective functions. This is called sequential approximate optimization with meta-modeling. It has been observed that techniques of computational intelligence can be effectively applied for this purpose. This talk will discuss several issues in sequential approximate multiobjective optimization using computational intelligence.

Keywords Sequential Approximate Optimization; Multiobjective Optimization; Metamodeling; Global Optimizaiton

1 Introduction

In order to solve optimization problems with expensive objective/constraint functions, we can apply effectively sequential approximate optimization technniques in which optimization is performed in parallel with predicting those expensive functions. Sequential approximate optimization consists of two phases: 1) construction of a metamodel, and 2) optimization for the metamodel. For the phase 1), there have been developed many kinds of methods which are called *metamodeling* (in the sense of making "model of the model"), *surrogate modeling*, *response surface methods*, etc. depending on research communities [8, 11, 21]. For the phase 2), all kinds of optimization techniques may be available in general. In many practical enginering design problems, however, due to complex nonlinearity and muli-modality metaheuristic methods such as genetic algorithms, particle swarm optimization methods, ant colony optimization methods, and differential evolution methods may be effectively applied. In the following, we discuss metamodeling in more detail.

2 Metamodeling

The identification of function forms of objective functions and constraints functions is referred to as "modeling" in practical problems. For the sake of simplicity, we consider

throughout this section the following simple problem:

$$\text{minimize} \qquad f(\boldsymbol{x}) \qquad \text{over} \qquad \boldsymbol{x} \in X \subset \mathbb{R}^n.$$

Then, the function f is also considered as a "*model*" of objectives. When it is difficult to identify the function form of f, but when we can observe the value of $f(\boldsymbol{x})$ for sampled point $\boldsymbol{x}$, we try to get an approximate function (or model) $\hat{f}$ for f on the basis of observations $(\boldsymbol{x}_i, y_i)$ where $y_i = f(\boldsymbol{x}_i)$ and $i = 1, \ldots, \ell$. This approximate function $\hat{f}$ is called a *metamodel*.

Now, our aim is to construct a good metamodel in the sense that

i) we can obtain an approximate optimal solution through the metamodel with the property
$$|\hat{f}(\hat{\boldsymbol{x}}^*) - f(\boldsymbol{x}^*)| \leqq \varepsilon_1,$$
where $\hat{\boldsymbol{x}}^*$ and $\boldsymbol{x}^*$ minimize $\hat{f}$ and f, respectively, and ε_1 is a given small positive number,
ii) the total number of observations is as small as possible,
iii) the metamodel $\hat{f}$ approximates well f entirely, if possible. Namely
$$||\hat{f} - f|| \leqq \varepsilon_2,$$
where ε_2 is a given small positive number.

If our aim is merely to find the optimal solution minimizing $f(\boldsymbol{x})$, then the metamodel $\hat{f}$ does not necessarily approximate well f entirely, but sufficiently well at least in a neighborhood of the optimal solution $\boldsymbol{x}^*$. Depending on practical problems, however, one may want to see the global behavior of the model f. Therefore, the priority of iii) above is behind the criteria i) and ii) which are crucial in general.

Metamodels can be constructed by plynomial regression, logistic regression, thin plate splines, radial basis function networks, support vector regression (SVR) and so on. In this paper, we review SVR briefly. Support Vector Machine (SVM) proposed by Vapnik *et al.* in the middle of 90's [1] is now recognized as an effective tool for machine learning [3, 17, 18]. SVR is a version of SVM applied to regression problems, in which the ε insensitive loss function is introduced [18]. It should be noted that the sparseness of support vectors results from this ε insensitive loss function.

Denote the given sample data by $(\boldsymbol{x}_i, y_i)$, $i = 1, \ldots, \ell$. Define a metamodel $\hat{f}$ on the feature space Z mapped from the data space by some nonlinear map Φ as follows:

$$\hat{f}(\boldsymbol{z}) = \boldsymbol{w}^T \boldsymbol{z} + b,$$

and the linear ε insensitive loss function is defined by

$$L^{\varepsilon}(\boldsymbol{z}, y, \hat{f}) = |y - \hat{f}(\boldsymbol{z})|_{\varepsilon} = \max(0, |y - \hat{f}(\boldsymbol{z})| - \varepsilon).$$

Taking into account this linear ε insensitive loss function, C-SVR allowing an error ξ_i ($\acute{\xi}_i$) for each data to some extent is defined as follows: For a given insensitivity parameter

ε,

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

where C is a trade-off parameter between the norm of $\boldsymbol{w}$ and ξ_i ($\acute{\xi}_i$).

On the other hand, the authors proposed another version of SVR, called μ−SVR, minimizing the maximum of error ξ_i as follows [15]: For a given insensitivity parameter ε,

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

where μ is a trade-off parameter between the norm of $\boldsymbol{w}$ and ξ ($\acute{\xi}$).

The dual formulation to the problem $(\mu\text{−SVR})_P$ is given by

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

where $K(\boldsymbol{x}_i,\boldsymbol{x}_j)$ is a kernel satisfying $K(\boldsymbol{x}_i,\boldsymbol{x}_j) =< \Phi(\boldsymbol{x}_i),\Phi(\boldsymbol{x}_j) >$. The optimal formula of metamodel is given by

$$\hat{f}(\boldsymbol{x}) = \sum_{i=1}^{\ell}\left(\acute{\alpha}_i^*-\alpha_i^*\right)K(\boldsymbol{x},\boldsymbol{x}_i)+b^*$$

where $\acute{\alpha}_i^*$, α_i^* are the solutions to $(\mu\text{−SVR})$ and b^* is the Lagrange multiplier to the equality constraint in $(\mu\text{−SVR})$. It has been observed that μ−SVR provides a good performance robust against outliers with less support vectors [15].

3 Incremental Design of Experiments

In sequential approximate optimization, one of most major issues is how to obtain an approximate optimal solution by as less number of experiments as possible. To this aim, we usually start with a relatively small number of experiments and add experiments sequentially, if necessary. For design of incremental experiments, several optimality criteria in the thoery of experimental design may be applied. However, it has been observed that those optimality criteria such as the so-called alphabetical optimality (D-optimality, E-optimality and etc.) depend on model formula. For example. if we adopt linear polynomial models and D-optimality, then candidate samples to be added tend to be concentrated in the area of edgepoints.

On the other hand, Jones *et al.* proposed to decide additional samples on the basis of expected improvement in the *efficient global optimization* (EGO) [7]. The EGO is a kind of global optimization methods using Bayesian approach (*Bayesian global optimization*) [10, 23], and uses the Kriging model as a prediction method for unknown function.

3.1 Expected Improvement

The expected improvement is evaluated on the basis of the Kriging model as follows: Consider the response $y(\boldsymbol{x})$ as a realization of a random function, $Y(\boldsymbol{x})$ such that

$$Y(\boldsymbol{x}) = \mu(\boldsymbol{x}) + Z(\boldsymbol{x}). \tag{1}$$

Here, $\mu(\boldsymbol{x})$ is a global model and $Z(\boldsymbol{x})$ reflecting a deviation from the global model is a random function with zero mean and nonzero covariance given by

$$cov[Z(\boldsymbol{x}), Z(\boldsymbol{x}')] = \sigma^2 R(\boldsymbol{x}, \boldsymbol{x}') \tag{2}$$

where R is the correlation between $Z(\boldsymbol{x})$ and $Z(\boldsymbol{x}')$. Usually, the stochastic process is supposed to be stationary, which implies that the correlation $R(\boldsymbol{x}, \boldsymbol{x}')$ depends only on $\boldsymbol{x} - \boldsymbol{x}'$, namely

$$R(\boldsymbol{x}, \boldsymbol{x}') = R(\boldsymbol{x} - \boldsymbol{x}'). \tag{3}$$

A commonly used example of such correlation functions is

$$R(\boldsymbol{x}, \boldsymbol{x}') = \exp[-\sum_{i=1}^{n} \theta_i |x_i - x_i'|^2], \tag{4}$$

where x_i and x_i' are i-th component of $\boldsymbol{x}$ and $\boldsymbol{x}'$, respectively.

Although a linear regressrion model $\sum_{j=1}^{k} \mu_j f_j(\boldsymbol{x})$ can be applied as a global model in (1) (*universal Kriging*), $\mu(\boldsymbol{x}) = \mu$ in which μ is unknown but constant is commonly used in many cases (*ordinary Kriging*). In the ordinary Kriging, the best linear unbiased predictor of y at an untried x can be given by

$$\hat{y}(\boldsymbol{x}) = \hat{\mu} + \boldsymbol{r}^T(\boldsymbol{x}) \boldsymbol{R}^{-1} (\boldsymbol{y} - \mathbf{1}\hat{\mu}), \tag{5}$$

where $\hat{\mu} = (\mathbf{1}^T \boldsymbol{R}^{-1} \mathbf{1})^{-1} \mathbf{1}^T \boldsymbol{R}^{-1} \boldsymbol{y}$ is the generalized least squares estimator of μ, $\boldsymbol{r}(\boldsymbol{x})$ is the $n \times 1$ vector of correlations $R(\boldsymbol{x}, \boldsymbol{x}_i)$ between Z at $\boldsymbol{x}$ and sampled points $\boldsymbol{x}_i$ $(i = 1, \ldots, p)$, $\boldsymbol{R}$ is an $n \times n$ correlation matrix with (i, j)-element defined by $R(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and $\mathbf{1}$ is a unity vector whose components are all 1.

The estimated value of the mean of the stochastic process, $\hat{\mu}$, is given by

$$\hat{\mu} = \frac{\mathbf{1}^T \boldsymbol{R}^{-1} \boldsymbol{y}}{\mathbf{1}^T \boldsymbol{R}^{-1} \mathbf{1}}. \tag{6}$$

In this event, the variation σ^2 is estimated by

$$\hat{\sigma}^2 = \frac{(\boldsymbol{y} - \mathbf{1}\hat{\mu})^T \boldsymbol{R}^{-1} (\boldsymbol{y} - \mathbf{1}\hat{\mu})}{n}. \tag{7}$$

The mean squared error of the predictor is estimated by

$$s^2(\boldsymbol{x}) = \sigma^2 [1 - \boldsymbol{r}^T \boldsymbol{R}^{-1} \boldsymbol{r} + \frac{(1 - \mathbf{1}^T \boldsymbol{R}^{-1} \boldsymbol{r})^2}{\mathbf{1}^T \boldsymbol{R}^{-1} \mathbf{1}}]. \tag{8}$$

In the following $s = \sqrt{s^2(\boldsymbol{x})}$ is called a standard error.

Using the above predictor on the basis of stochastic process model, Jones *et al.* applied the expected improvemnet for adding a new sample point. Let $f^p_{\min} = \min\{y_1, \ldots, y_p\}$ be the current best function value. They model the uncertainty at $y(\boldsymbol{x})$ by treating it as the realization of a normally distributed random variable Y with mean and standard deviation given by the above predictor and its standard error.

For minimization cases, the improvement at $\boldsymbol{x}$ is $I = [\max(f^p_{\min} - Y,\ 0)$. Therefore, the expected improvement is given by

$$\mathrm{E}[I(\boldsymbol{x})] = \mathrm{E}[\max(f^p_{\min} - Y,\ 0)].$$

It has been shown that the above formula can be expanded as follows:

$$E(I) = \begin{cases} (f^p_{\min} - \hat{y})\Phi(\frac{f^p_{\min} - \hat{y}}{s}) + s\phi(\frac{f^p_{\min} - \hat{y}}{s}) & \text{if} \quad s < 0 \\ 0 & \text{if} \quad s = 0, \end{cases} \tag{9}$$

where ϕ is the standard normal density and Φ is the distribution function.

We can add a new sample point which maximizes the expected improvement. Although Jones *et al.* proposed a method for maximizing the expected improvement by using the branch and bound method, it is possible to select the best one among several candidates which are generated randomly in the design variable space.

3.2 Distance-based Local and Global Information

One of most important tasks in optimization is to balance between exploration and exploitation. The exploration is to search an optimal solution from a global viewpoint, while the exploitation is to search an optimal solution in a good precision (namely, from a local viewpoint). Therefore, it is important to design additional exepriments taking into account global information and local information on the metamodel. It should be noted that EGO using EI decides an additional sample by controling the weight on global information and local information depnding on the situation.

Nakayama *et al.* [13] have suggested the method which gives both global information for predicting the entire objective function and local information near the optimal point

at the same time. In this method, two kinds of additional data are taken for relearning the form of the objective function. One of them is selected from 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 data gives more detailed information near the current optimal point. The latter data prevents from converging to local optimum point.

The neighborhood of the current optimal point is given by a square S, whose center is the current optimal point, with the length of a side l. Let S_0 be a square, whose center is the current optimal point, with the fixed length of a side l_0. The square S is shrinked according to the number C_x of optimal points appeared continuously in S_0 in the past. That is,

$$l = l_0 \times \frac{1}{C_x + 1}. \tag{10}$$

The first additional data is selected inside the square S randomly. The second additional data is selected in an area, in which the existing learning data are sparse, outside the square S. An area with sparsely existing data may be found as follows:

i) First, a certain number (N_{rand}) of data are generated randomly outside the square S.
ii) Denote d_{ij} the distance from this random data p_i $(i = 1, \ldots, N_{rand})$ to the existing learning data q_j $(j = 1, \ldots, N)$. Select the shortest k distances $\tilde{d}_{ij}$ $(j = 1, \ldots, k)$ for each p_i, and sum up these k distances, i.e., $D_i = \sum_{j=1}^{k} \tilde{d}_{ij}$.
iii) Take p_t which maximizes $\{D_i\}_{(i=1,\ldots,N_{rand})}$ as an additional data outside S.

The algorithm using distance-based local and global information by [13] can be summarized as follows:

Step 1. Predict the form of the objective function by some regression method on the basis of the given training data.
Step 2. Estimate an optimal point for the predicted objective function by some optimization method.
Step 3. Count the number of optimal points appeared continuously in the past in S_0. This number is represented by C_x.
Step 4. Terminate the iteration,
- if C_x is larger than or equal to the given C_x^0 a priori, or
- if the best value of the objective function obtained so far is identical during the last certain number (C_f^0) of iterations.

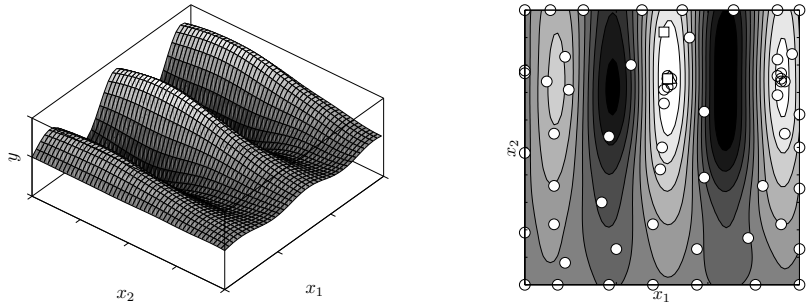
Otherwise calculate l by the equation (10), and go to the next step.
Step 5. Select an additional data near the current optimal value, i.e., inside S.
Step 6. Select another additional data outside S in a place in which the density of the training data is low as stated above.
Step 7. Go to Step 1.

Example 1.

Consider a simple example given by

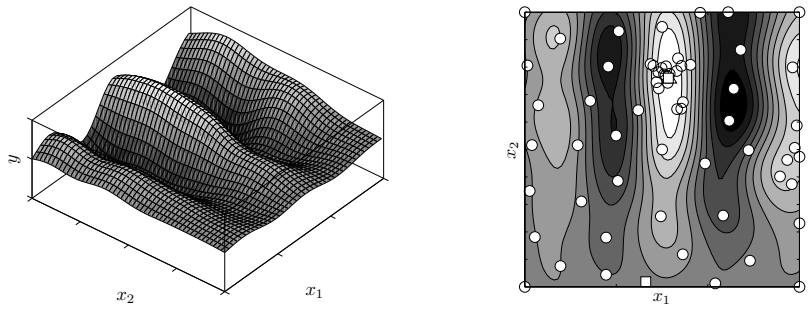
$$\begin{array}{ll} \underset{x_1,x_2}{\text{maximize}} & f(\boldsymbol{x}) = 10\exp\left(-\dfrac{(x_1-10)^2+(x_2-15)^2}{100}\right)\sin x_1 \\ \text{subject to} & 0 \le x_1 \le 15,\ \ 0 \le x_2 \le 20. \end{array} \tag{11}$$

This problem has an optimal (maximal) value $f^* = 9.5585$ at $x_1^* = 7.8960$ and $x_2^* = 15$. Fig.1 shows the result at the 63 samples by EGO using EI. On the other hand, Fig.2 shows the result at the 61 samples by RBF using distance-based global and local information. One may see that both results are almost the same: although EI seems slightly better than by distance-based global and local information, the calculation of EI takes more time.



$\hat{x}_1^* = 7.8964,\ \hat{x}_2^* = 14.9819,\ \hat{f}^* = 9.5587$

Figure 1: #data = 63 (final iteration) by EGO using EI



$\hat{x}_1^* = 7.8947,\ \hat{x}_2^* = 15.1424,\ \hat{f}^* = 9.2880$

Figure 2: #data = 61 (final iteration) by distance-based global and local information

4 Sequential Approximate Multiobjective Optimization

In multi-objective optimization, it is one of main issues how to obtain Pareto optimal solutions, and how to choose one solution from many Pareto optimal solutions. To this

end, the *interactive optimization method* [5, 9, 22], for example, *aspiration level method* [16, 12], have been developed. Aspiration level method searches a solution by processing the following two stages repeatedly: 1) solving auxiliary optimization problem to obtain the closest Pareto optimal solution to a given aspiration level of decision maker, and 2) revising her/his aspiration level by making the trade-off analysis. Conventional interactive optimization methods are useful in particular in cases with many objective functions, in which it is difficult to visualize Pareto frontier, and also to depict the trade-off among many objective functions. In cases with two or three objective functions, on the other hand, it may be the best way to depict Pareto frontier, because visualizing Pareto frontier helps to grasp trade-off among objective functions. For that purpose, evolutionary methods such as genetic algorithm (GA) have been effectively applied for solving a multi-objective optimization problem: so called *evolutionary multi-objective optimization* (EMO) methods have been proposed for generating Pareto optimal solutions [4, 2]. However, EMO has some problems: i) it is difficult to visualize Pareto frontiers in cases with many objective functions, ii) many function evaluations are usually needed for generating the whole Pareto frontier. Considering the number of function evaluations, it would be rather reasonable to generate not the whole Pareto frontier, but a necessary part of it in which the decision maker may be interested. To this aim, we introduce some methods combining aspiration level approach and computational intelligence method in this section.

A multi-objective optimization problem (MOP) can be formulated as follows:

$$\begin{array}{ll} \underset{\boldsymbol{x}}{\text{minimize}} & \boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \ldots, f_r(\boldsymbol{x}))^T \\ \text{subject to} & \boldsymbol{x} \in X = \left\{ \boldsymbol{x} \in \mathbb{R}^n \mid g_j(\boldsymbol{x}) \leqq 0,\ j = 1, \ldots, m \right\}, \end{array} \tag{MOP}$$

where X denotes the set of all feasible solutions in the design variable space.

To begin with, we summarize the method for *sequential approximate multi-objective optimization* (shortly, SAMO) using satisficing trade-off method proposed by [16] as follows:

Step 1. Calculate the real values of objective functions $\boldsymbol{f}(\boldsymbol{x}_1), \ldots, \boldsymbol{f}(\boldsymbol{x}_\ell)$ for given sample points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_\ell$.

Step 2. Approximate each objective function $f_k(\boldsymbol{x}),\ k = 1, \ldots, r$, by using some regression method on the basis of training data set $(\boldsymbol{x}_i, f_k(\boldsymbol{x}_i)),\ i = 1, \ldots, \ell$. An optimal solution/value to approximate objective function $\hat{\boldsymbol{f}}(\boldsymbol{x})$ is called an *approximate optimal solution/value*.

Step 3. Find an approximate optimal solution $\boldsymbol{x}^a$ closest to the given aspiration level $\overline{\boldsymbol{f}}$ by solving the following problem (AP) of satisficing trade-off method:

$$\begin{array}{ll} \underset{\boldsymbol{x}, z}{\text{minimize}} & z + \alpha \displaystyle\sum_{i=1}^{r} w_i \hat{f}_i(\boldsymbol{x}) \\ \text{subject to} & w_i \left(\hat{f}_i(\boldsymbol{x}) - \overline{f}_i \right) \leqq z,\ i = 1, \ldots, r, \\ & \boldsymbol{x} \in X, \end{array} \tag{AP}$$

where α is a sufficiently small positive number, for example 10^{-6}, $w_i = 1/(\overline{f}_i - f_i^*)$ and f_i^* is an ideal value.

Furthermore, generate approximate Pareto optimal solutions $\boldsymbol{x}^1,\ldots,\boldsymbol{x}^p$ to $\hat{\boldsymbol{f}}$ by using EMO for approximating the whole set of Pareto solutions and for deciding the second additional samples which will be stated in Step 5 below.

Step 4. Stop the iteration if a certain stop condition is satisfied. Otherwise, go to the next step. The stop condition is given by, for example, the limitation of the number of sample points, the count of no-changing approximate solution obtained in the Step 3, and so on.

Step 5. Choose additional sample points for relearning, and go to Step 1.

It is important to improve the prediction ability for function approximation in order to find an approximate solution closer to the exact one with as small number of sample data as possible. To this aim, starting with relatively few initial samples, we add new samples step by step, if necessary. Here, we introduce one of the ways how to choose additional sample points proposed by Yun-Yoon-Nakayama [20].

(i) First, one additional sample point is added as the solution $\boldsymbol{x}^a$ closest to the given aspiration level which is found in step 3. This is for approximating well a neighborhood of Pareto optimal solution closest to the aspiration level, which enables to make easily the trade-off analysis among objective functions. Here, the additional point $\boldsymbol{x}^a$ is considered as a *local information of Pareto frontier*, because $\boldsymbol{x}^a$ can provide the information around the closest Pareto optimal solution to the aspiration level.

(ii) Another additional sample point is for depicting the configuration of Pareto frontier. This is for giving a rough information of the whole Pareto frontier, and we call this a *global information of Pareto frontier* in contrast with the above local information.

Stage 1. Evaluate the rank R_i for each sample point $\boldsymbol{x}_i,\ \ i=1,\ldots,\ell$ by the ranking method of Goldberg [6].

Stage 2. Approximate a ranking function $\hat{R}(\boldsymbol{x})$ on the basis of training data set $(\boldsymbol{x}_i,R_i)$, $i=1,\ldots,\ell$ by some regression method.

Stage 3. Calculate the values of ranking function $\hat{R}(\boldsymbol{x})$ for the approximate Pareto optimal solutions $\boldsymbol{x}^j,\ j=1,\ldots,p$ generated in step 3.

Stage 4. Among them, select a point with the best rank

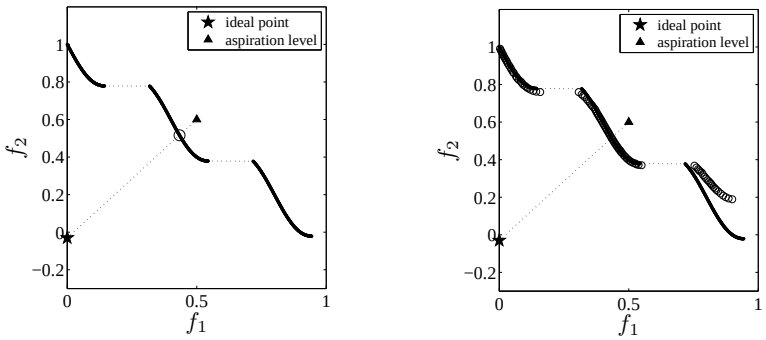
$$\boldsymbol{x}^b = \arg\min_{j=1,\ldots,p} \hat{R}(\boldsymbol{x}^j).$$

Example 2. (Case 1)

Consider the following problem with two design variables and two objective functions:

$$\begin{array}{lll} \underset{x_1,x_2}{\text{minimize}} & f_1(\boldsymbol{x}) = x_1 & \text{(Ex-1)} \\ & f_2(\boldsymbol{x}) = 1+x_2^2-x_1-0.1\sin(5\pi x_1) & \\ \text{subject to} & 0 \leqq x_1 \leqq 1,\ \ -2 \leqq x_2 \leqq 2. & \end{array}$$

Starting with initial sample 10 points generated randomly, we stop the iteration after 15 additional learning. We approximate each objective function by using $\mu-$SVR with gauss kernel function. Fig. 3 shows the results with 40 sample points after 15 additional learning, in which one may see that the obtained approximate Pareto optimal solutions are almost the same as the exact ones.



(a) approximate value closest to aspiration level

(b) approximate Pareto frontier

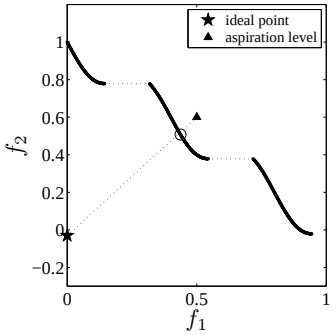
Figure 3: # data = 40 sample pts (after 15 additional learning) (Ex-1) : case 1

Example 2. (Case 2)

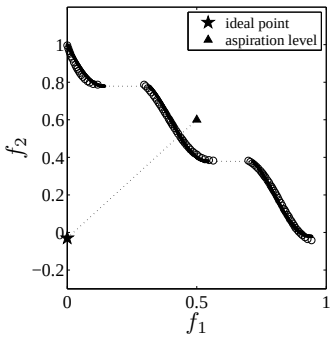
Note that the second additional sample at Step 5 seems to provide a local information in the sense that it is for generating a better approximation of the whole Pareto frontier. Indeed, the additional samples in the space of design variables $\boldsymbol{x}$ are concentrated in the neighborhood of Pareto optimal solutions. From a viewpoint of generating a better approximation of each objective function, further sample points for global information may be needed. As the third additional sample point, we recommend to add a point in a sparse area of existing samples. This is performed by a similar way as the distance based local and global information method. Fig. 4 shows the result obtained by this method. One may see that each objective function is approximated well enough and that a well approximated Pareto frontier is generated.

5 Concluding Remarks

The most prominent feature in SAMO described above is that combining the aspiration level method and EMO, it is possible to find the most interesting part of Pareto frontier for the decision maker as well as to grasp the configuration of the whole Pareto frontier. In particular in cases with many objective functions, since it is difficult and expensive to visualize the whole Pareto frontier, it becomes effective to restrict the search region to the most interesting part of Pareto frontier. An example of this approach can be seen in [15]. With further devices peculiar to given problems, the effectiveness of SAMO has been observed in a wide range of engineering problems [15].



(c) approximate value closest to aspiration level



(d) approximate Pareto frontier

Figure 4: # data = 40 (after 10 additional learning) : case 2

References

[1] Cortes, C. and Vapnik, V., Support vector networks, *Machine Learning*, vol. 20, pp. 273–297, 1995

[2] Coello, C. A. C., Van Veldhuizen, D. A. and Lamont, G. B., *Evolutionary Algorithms for Solving Multi-Objective Problems*, Kluwer Academic Publishers, 2001

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

[4] Deb, K., *Multi-Objective Optimization using Evolutionary Algorithms*, Wiley, 2001.

[5] Gal, T., Stewart, T. J. and Hanne, T., *JMulticriteria Decision Making – Advances in MCDM Models, Algorithms, Theory, and Applications*, Kluwer Academic Publishers, 1999.

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

[7] Jones, D. R., Schonlau, M. and Welch, W. J., Efficient Global Optimization of Expensive Black-Box Functions, *Journal of Global Optimization*, 455–492, 1998.

[8] Knowles, J. and Nakayama, H., Meta-Modeling in Multi-Objective Optimization, in Branke, J., Deb, K., Miettinen, K. and Slowinski, R. (eds.), *Multiobjective Optimization Interactive and Evolutionary Approaches*, Springer, 2008.

[9] Miettinen, K. M., Nonlinear Multiobjective Optimization, Kluwer Academic Publishers, 1999.

[10] Mockus, J., Application of Bayesian Approach to Numerical Methods of Global and Stochastic Optimization, *Journal of Global Optimization*, 347–365, 1994.

[11] Myers, R. and Montogomery, D., *Response Surface Methodology*, Wiley, 1995.

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

[13] Nakayama, H., Arakawa, M. and Sasaki, R., Simulation-based Optimization using Computational Intelligence, *Optimization and Engineering*, 201-214, 2002.

[14] Nakayama, H., Yun, Y., Generating Support Vector Machines using Multiobjective Optimization and Goal Programming, in Yaochu Jin (ed.), *Multi-Objective Machine Learning*, Springer, 2006.

[15] Nakayama, H., Yun, Y. and Yoon, M., *Sequential Approximate Multiobjective optimization Using Computational Intelligence*, Springer, 2009.

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

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

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

[19] Yun, Y. B., Nakayama, H. and Arakawa, M., Multiple Criteria Decision Making with Generalized DEA and an Aspiration Level Method, *European Journal of Operational Research*, 697–706, 2004.

[20] Yun, Y. B., Yoon, M. and Nakayama, H., Multi-objective Optimization based on Metamodeling by using Support Vector Regression, *Optimization and Engineering*, 167–181, 2009.

[21] Wang, G. G. and Shan, S., Review of Metamodeling Techniques in Support of Engineering Design Optimization, *ASME Transactions, Journal of Mechanical Design*, 370–380, 2006.

[22] Wierzbicki, A. P., Makowski, M. and Wessels, J., *Model-based Decision Support Methodology with Environmental Applications*, Kluwer Academic Publishers, 2000.

[23] Žilinskas, A., One-step Bayesian Method of the Search for Extremum of an One-dimensional Function, *Cybernetics*, 139-144, 1975.