Optim Eng (2009) 10: 167–181
DOI 10.1007/s11081-008-9063-1

Multi-objective optimization based on meta-modeling by using support vector regression

Yeboon Yun · Min Yoon · Hirotaka Nakayama

Received: 26 September 2008 / Accepted: 20 October 2008 / Published online: 6 November 2008

Abstract Practical engineering design problems have a black-box objective function whose forms are not explicitly known in terms of design variables. In those problems, it is very important to make the number of function evaluations as few as possible in finding an optimal solution. So, in this paper, we propose a multi-objective optimization method based on meta-modeling predicting a form of each objective function by using support vector regression. In addition, we discuss a way how to select additional experimental data for sequentially revising a form of objective function. Finally, we illustrate the effectiveness of the proposed method through some numerical examples.

Keywords Multi-objective optimization · Pareto frontier · Support vector regression · Sequential approximation method · Evolutionary multi-objective optimization

1 Introduction

Many decision making problems are formulated as multi-objective optimization problems so as to satisfy diverse demands of the decision maker. One of main issues in multi-objective optimization is how to find a Pareto optimal solution which meets decision maker's demands. To the end, interactive optimization methods, for

Y. Yun (✉)
Faculty of Engineering, Kagawa University, 2217-20 Hayashicho, Takamatsu, Kagawa 761-0396, Japan
e-mail: yun@eng.kagawa-u.ac.jp

M. Yoon
Konkuk University, 134 Shinchon-Dong, Seodaemun-Gu, Seoul 120-749, Republic of Korea

H. Nakayama
Konan University, 8-9-1 Okamoto, Higashinada, Kobe 658-8501, Japan

example aspiration level methods (Nakayama and Sawaragi 1984), have been developed. And, several methods using evolutionary algorithms have been researched for generating the whole set of Pareto optimal solutions (Coello Coello et al. 2002; Deb 2001; Fonseca and Fleming 1993; Yun et al. 2001; Zitzler et al. 2001). However, those conventional methods have some problems when applying them to real practical problems considering the number of function evaluations. In many engineering design problems, generally, there are black-box objective functions whose forms are not explicitly known in terms of design variables. Under this circumstance, given design variables, the values of objective function can be obtained by sampled real/computational experiments such as structural analysis, fluid-mechanical analysis, thermodynamic analysis, and so on. These analyses are expensive and time consuming, and it is an important issue in real application problems how to make the number of necessary analyses as few as possible. For that purpose, many researches on fitness approximation, as one of attempts, in evolutionary algorithm has been performed (Jin 2005). On the other hand, in design of experiments, several statistical methods for choosing experimental points have been proposed, for example orthogonal arrays, central composite designs and A, D, E-optimality criteria (Kiefer 1959). Generally, D-optimality criterion is most widely applied in design of experiments in which relatively low order (say, 1st or 2nd) polynomial models are used. The method called efficient global optimization (Jones et al. 1998) has been suggested for black-box functions. In this method, a new sample point is selected by maximizing the expected improvement. Although they proposed a method for maximizing the expected improvement by using the branch and bound method, it is not easy to find a point which maximizes the expected improvement. Recently, the authors have suggested to apply machine learning techniques such as radial basis function neural networks and support vector machines for approximating the black-box function (Emmerich et al. 2006; Chafekar et al. 2005; Nakayama et al. 2001, 2002, 2003; Yun et al. 2007). In addition, it was proposed that the sample points are chosen by considering both global and local information of the black-box function for a single-objective optimization.

In this paper, combining aspiration level method and meta-modeling, we suggest a multi-objective optimization method for predicting a form of each objective function by using support vector regression (SVR) and to find a set of Pareto optimal solutions by using genetic algorithm. In addition, by using the sensitivity information of SVR at each sample point, we proposed how to select additional sample points for sequentially revising a form of objective functions. Finally, we illustrate the effectiveness of the proposed method through numerical examples.

2 Support vector machines

In this section, to begin with, we introduce support vector regression (SVR), which is a kind of Support vector machine (SVM) for function approximation.

SVM has been recognized as a powerful machine learning technique. SVM was originally developed for pattern classification and later extended to regression (Cristianini and Shawe-Taylor 2000; Cortes and Vapnik 1995; Schölkopf and Smola 2002; Vapnik 1998). Therefore, we review briefly SVM for classification.

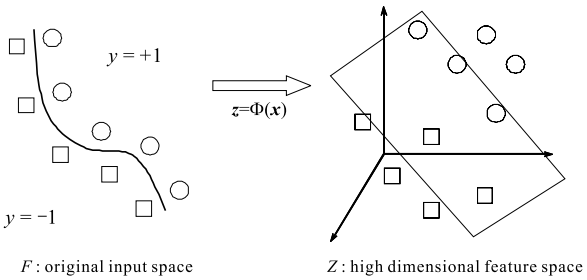


Fig. 1 Geometric Interpretation of SVM

Let $\mathcal{F}$ be a space of conditional attributes. For binary classification problems, the value of $+1$ or -1 is assigned to each pattern $\boldsymbol{x}_i \in \mathcal{F}$ according to its class $\mathcal{A}$ or $\mathcal{B}$. The aim of machine learning is to predict which class newly observed patterns belong to on the basis of the given training data set $(\boldsymbol{x}_i, y_i)$ $(i = 1, \ldots, \ell)$, where $y_i = +1$ or -1. Linear discriminant functions can be expressed by a form

$$f(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x} + b$$

with the property $\boldsymbol{w}^T \boldsymbol{x} + b \geqq 0$ for $\boldsymbol{x} \in \mathcal{A}$ and $\boldsymbol{w}^T \boldsymbol{x} + b < 0$ for $\boldsymbol{x} \in \mathcal{B}$.

In cases where training data set $\mathcal{F}$ is not linearly separable, we map the original data set $\mathcal{F}$ to a feature space Z by some nonlinear map ϕ. Increasing the dimension of the feature space, it is expected that the mapped data set becomes linearly separable. We try to find linear classifiers with maximal margin in the feature space (see Fig. 1).

Letting $\boldsymbol{z}_i = \phi(\boldsymbol{x}_i)$, the separating hyperplane with maximal margin can be given by solving the following problem with the normalization $\boldsymbol{w}^T \boldsymbol{z} + b = \pm 1$ at points with the minimum distance:

$$\begin{aligned} &\underset{\boldsymbol{w}, b}{\text{minimize}} \ \frac{1}{2} ||\boldsymbol{w}||_2^2 && \text{(hard}-\text{SVM)}_P \\ &\text{subject to } y_i \left(\boldsymbol{w}^T \boldsymbol{z}_i + b \right) \geqq 1, \quad i = 1, \ldots, \ell. \end{aligned}$$

Using the kernel function $K(\boldsymbol{x}, \boldsymbol{x}') = \phi(\boldsymbol{x})^T \phi(\boldsymbol{x}')$, the dual problem for (hard-SVM)$_P$ can be obtained as follows:

$$\begin{aligned} &\underset{\boldsymbol{\alpha}}{\text{maximize}} \ \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{\ell} \alpha_i \alpha_j y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) && \text{(hard}-\text{SVM)} \\ &\text{subject to } \sum_{i=1}^{\ell} \alpha_i y_i = 0, \\ &\qquad \alpha_i \geqq 0, \quad i = 1, \ldots, \ell. \end{aligned}$$

Although several kinds of kernel functions have been suggested, typical kernel functions are Gaussian kernel function and pth-order polynomial kernel function,

and so on (Schölkopf and Smola 2002):

$$K(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|_2^2}{2\sigma^2}\right),$$
$$K(\boldsymbol{x}, \boldsymbol{x}') = \left(\langle \boldsymbol{x} \cdot \boldsymbol{x}' \rangle + 1\right)^p.$$

The above hard−SVM is the most basic model, and various SVM models have been suggested. For the details, see the references (Cristianini and Shawe-Taylor 2000; Cortes and Vapnik 1995; Schölkopf and Smola 1998). Later, SVM has been extended to regression by introducing the $\varepsilon-$insensitive loss function (Vapnik 1995), for example, soft−SVR (Vapnik 1995) and $\nu-$SVR (Schölkopf and Smola 1998).

Using multi-objective programming and goal programming techniques (Nakayama and Yun 2006a; Yoon et al. 2004), the authors have developed several variants of SVM and extended the family of SVM for classification to regression (Nakayama and Yun 2006b) including the primal problem

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

and its dual problem

$$\underset{\boldsymbol{\alpha},\acute{\boldsymbol{\alpha}}}{\text{maximize}} \ -\frac{1}{2}\sum_{i,j=1}^{\ell}\left(\acute{\alpha}_i - \alpha_i\right)\left(\acute{\alpha}_j - \alpha_j\right) K\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) + \sum_{i=1}^{\ell}\left(\acute{\alpha}_i - \alpha_i\right) y_i \qquad (\mu\text{–}\nu\text{–SVR})$$

$$\begin{aligned} \text{subject to} \ & \sum_{i=1}^{\ell}\left(\acute{\alpha}_i - \alpha_i\right) = 0, \\ & \sum_{i=1}^{\ell}\acute{\alpha}_i \leqq \mu, \qquad \sum_{i=1}^{\ell}\alpha_i \leqq \mu, \\ & \sum_{i=1}^{\ell}\left(\acute{\alpha}_i + \alpha_i\right) \leqq \nu, \\ & \acute{\alpha}_i \geqq 0, \qquad \alpha_i \geqq 0, \quad i = 1, \ldots, \ell, \end{aligned}$$

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

$$f(\boldsymbol{x}) = \sum_{i=1}^{\ell} \beta_i K(\boldsymbol{x}, \boldsymbol{x}_i), \quad \beta_i = \acute{\alpha_i} - \alpha_i.$$

One of the most prominent features in μ–ν–SVR is that it provides relatively less support vectors, which means the computation is less expensive. The fact that μ–ν–SVR yields good function approximation with reasonable accuracy and with less support vectors, is important in practice in engineering design. For some kinds of engineering design problems, approximation functions should be realized on the basis of as few data points as possible. Such practical engineering design problems are considering as the application, and therefore, we adopt μ–ν–SVR as a useful tool of function approximation in this paper.

3 Multi-objective optimization problem

Next, we explain the concept of solution in multi-objective optimization. Consider a multi-objective optimization problem with m-objective functions and n-dimensional decision variable as follows:

$$\begin{aligned} &\underset{\boldsymbol{x}}{\text{minimize}}\ \boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x}))^T \\ &\text{subject to } \boldsymbol{x} \in X = \left\{ \boldsymbol{x} \in \mathbb{R}^n \mid g_j(\boldsymbol{x}) \leqq 0,\ j = 1, \ldots, l \right\}, \end{aligned} \tag{MP}$$

where X is the set of all feasible solution set in design variable space.

The objective function space is partially ordered, thus for convenience, the following notations are used for two vectors $\boldsymbol{y}^1 = (y_1^1, \ldots, y_m^1)^T$ and $\boldsymbol{y}^2 = (y_1^2, \ldots, y_m^2)^T$ in $\mathbb{R}^m$.

$$\begin{aligned} \boldsymbol{y}^1 > \boldsymbol{y}^2 \quad &\Longleftrightarrow \quad y_i^1 > y_i^2, \quad i = 1, \ldots, m, \\ \boldsymbol{y}^1 \geqq \boldsymbol{y}^2 \quad &\Longleftrightarrow \quad y_i^1 \geqq y_i^2, \quad i = 1, \ldots, m, \\ \boldsymbol{y}^1 \geq \boldsymbol{y}^2 \quad &\Longleftrightarrow \quad y_i^1 \geqq y_i^2, \quad i = 1, \ldots, m \text{ but } \boldsymbol{y}^1 \neq \boldsymbol{y}^2. \end{aligned}$$

Generally, unlike traditional optimization problem with a single objective function, there seldom exists an optimal solution that minimizes all objective functions $f_i(\boldsymbol{x}),\ i = 1, \ldots, m$, simultaneously in the problem (MP). Using on the above partial order relation, hence Pareto optimal solution is defined as follows (Sawaragi et al. 1985):

Definition 1 (Pareto optimal solution) A point $\hat{\boldsymbol{x}} \in X$ is said to be a *Pareto optimal solution* if there exists no $\boldsymbol{x} \in X$ such that $\boldsymbol{f}(\boldsymbol{x}) \leq \boldsymbol{f}(\hat{\boldsymbol{x}})$. Also, Pareto optimal solution $\hat{\boldsymbol{x}} \in X$ in the objective function space is called as *Pareto optimal value*, and the set of them is *Pareto frontier* (see Fig. 2).

Fig. 2 Pareto frontier

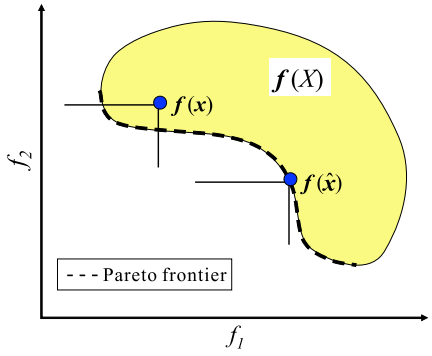
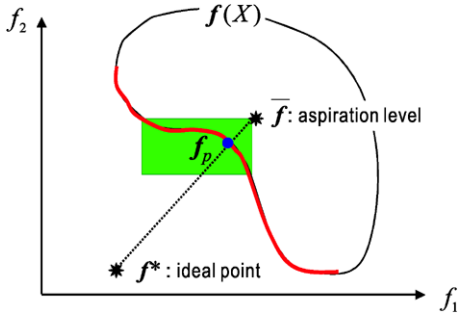


Fig. 3 Aspiration level method



Usually, Pareto optimal solutions are considered as the candidates of a final solution to the problem (MP). Therefore, main issues in multi-objective optimization are how to obtain Pareto optimal solutions and how to choose one solution from the set of Pareto optimal solutions. To this end, many interactive optimization methods, for example, aspiration level approach (Nakayama and Sawaragi 1984) have been developed. In the aspiration level approach, for a given aspiration level $\overline{\boldsymbol{f}} := \left(\overline{f}_1, \ldots, \overline{f}_m\right)^T$, we consider the following problem with the augmented Tchebyshev scalarization function:

$$\begin{aligned} \underset{\boldsymbol{x},\delta}{\text{minimize}}\ & \delta + \lambda \sum_{i=1}^{m} \omega_i \left(f_i(\boldsymbol{x}) - \overline{f}_i\right) \\ \text{subject to}\ & \omega_i \left(f_i(\boldsymbol{x}) - \overline{f}_i\right) \leqq \delta, \quad i = 1, \ldots, m, \\ & \boldsymbol{x} \in X, \end{aligned} \tag{AP}$$

where $\omega_i = \frac{1}{\overline{f}_i - f_i^*}, i = 1, \ldots, m$, $\boldsymbol{f}^* := \left(f_1^*, \ldots, f_m^*\right)^T$ is an ideal point, and λ is sufficiently small number, for example, 10^{-7}.

By solving the problem (AP), as shown in Fig. 3, we can obtain the closest Pareto optimal value $\boldsymbol{f}_p$ where is an intercept of Pareto frontier and the line passing the aspiration level point $\overline{\boldsymbol{f}}$ and the ideal point $\boldsymbol{f}^*$.

On the other hand, recently GA has been developed for solving a multi-objective optimization problem. GA, which is a kind of multi-point searching methods, can generate a set of Pareto optimal solutions. As a result, GA has been shown to be very effective method in multi-objective optimization, and several evolutionary multi-objective optimization (EMO) have been researched (Coello Coello et al. 2002; Deb 2001). Main efforts in EMO are made for the diversity and the accuracy of solutions. Consequently, when applying EMO to actual problems, so many function evaluations (= the number of population × generation) are needed. However, in the problems such as engineering design problems, because the value of objective function cannot be obtained so easily, it is very important to reduce the number of function evaluations as few as possible in finding an explicit/implicit optimal solution to the problem. In order to reduce the number of function evaluations, we have proposed several algorithms (Yun et al. 2001, 2004, 2007).

4 Multi-objective optimization based on meta-modeling

In this paper, we suggest a multi-objective optimization method based on meta-modeling that

(i) predict a form of objective function on the basis of some sample data by using SVR,
(ii) find a set of Pareto optimal solutions to the predicted objective functions by using EMO, and
(iii) choose an additional data point for re-predicting a form of objective function.

The proposed method can be summarized concretely as follows:

Step 1. (Real Calculation) Calculate actually the real value of objective function $\boldsymbol{f}(\boldsymbol{x}_i)$ for sampled data point $\boldsymbol{x}_i,\ i = 1, \ldots, \ell$.

Step 2. (Prediction) Predict each objective function $\hat{f}_1(\boldsymbol{x}), \ldots, \hat{f}_m(\boldsymbol{x})$ by using μ–ν–SVR on the basis of training data set $\{(\boldsymbol{x}_i, \boldsymbol{f}(\boldsymbol{x}_i)),\ i = 1, \ldots, \ell\}$.

Step 3. (Generation of Pareto optimal solutions) Generate the whole set of Pareto optimal solutions $\{\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^k\}$ to the predicted objective functions $\hat{\boldsymbol{f}}(\boldsymbol{x}) = (\hat{f}_1(\boldsymbol{x}), \ldots, \hat{f}_m(\boldsymbol{x}))$ by using EMO.

Step 4. (Choice of additional data) Choose additional points for relearning from Pareto optimal solutions, and go to Step 1 if not satisfied with the stop condition depending on the number of function evaluations and the convergence situation of the solution. Here, we propose how to choose the additional points.

(i) In order to approximate well a neighborhood of the Pareto optimal solution closest to the aspiration level, that is, to make easily a trade-off analysis among objective functions, first one additional point $\boldsymbol{x}^a$ among the set of Pareto optimal solutions $\{\boldsymbol{x}^1, \ldots, \boldsymbol{x}^k\}$ obtained at Step 3 is selected by the equation

$$\boldsymbol{x}^a = \arg \min_{j=1,\ldots,k} \left\{ \max_{i=1,\ldots,m} \left\{ \omega_i \left(\hat{f}_i(\boldsymbol{x}^j) - \overline{f}_i \right) \right\} + \lambda \sum_{i=1}^{m} \omega_i \left(\hat{f}_i(\boldsymbol{x}^j) - \overline{f}_i \right) \right\}.$$

The point $\boldsymbol{x}^a$ is the same as the solution to the problem (AP) of aspiration level method.

(ii) For predicting more accurately objective function, we utilize the value of Lagrangian multiplier $\boldsymbol{\beta}$ in SVR as a sensitivity information. If $\beta_i \neq 0$ then i-th sample point is a support vector which may be on the boundary or outside the band of $\pm\varepsilon$ around the predicted objective function $f(\boldsymbol{x})$. The support vectors are considered to be important, since the regression function is represented only in terms of support vectors.

- calculate $\|\boldsymbol{\beta}_i\| = \|(\beta_{i1}^*, \ldots, \beta_{im}^*)\|$ for each sample point $\boldsymbol{x}_i,\ i = 1, \ldots, \ell$.
- $\boldsymbol{x}^c = \arg\max_{i=1,\ldots,\ell} \|\boldsymbol{\beta}_i\|$, where $\|\cdot\|$ is given by, for example, $\|\boldsymbol{\beta}_i\|_2^2$ or $\max\{|\beta_{i1}^*|, \ldots, |\beta_{im}^*|\}$.
- choose the second additional sample point $\boldsymbol{x}^b$ among the set of Pareto optimal solutions $\{\boldsymbol{x}^1, \ldots, \boldsymbol{x}^k\}$ obtained at Step 3 by the following equation

$$\boldsymbol{x}^b = \arg\min_{j=1,\ldots,k} \|\boldsymbol{x}^j - \boldsymbol{x}^c\|.$$

Generally, the predicted regression function may be unstable at the sample point with a high value of Lagrangian multiplier $\boldsymbol{\beta}_i$ in SVR . Therefore, by adding some points around those sample points, more accurate function prediction can be expected, and this yields that it can make it to reduce the number of function evaluations.

5 Numerical example

For illustrating the proposed method, consider a simple problem with two design variables and two objective functions, and the true contours of objective functions are shown in Fig. 4.

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

The three aspiration level are given by

$$\begin{aligned} &\text{aspiration level 1} \quad (\overline{f}_1, \overline{f}_2) := (0.2, 0.9),\\ &\text{aspiration level 2} \quad (\overline{f}_1, \overline{f}_2) := (0.5, 0.6),\\ &\text{aspiration level 3} \quad (\overline{f}_1, \overline{f}_2) := (0.9, 0.2). \end{aligned}$$

In our simulation, we start with randomly generated 10 points, and the results after each 5 additional learning are shown in Figs. 5–13. We adopt SPEA2 (Zitzler et al. 2001) in generating the Pareto optimal solutions, and the parameters of SPEA2 are set as follows.

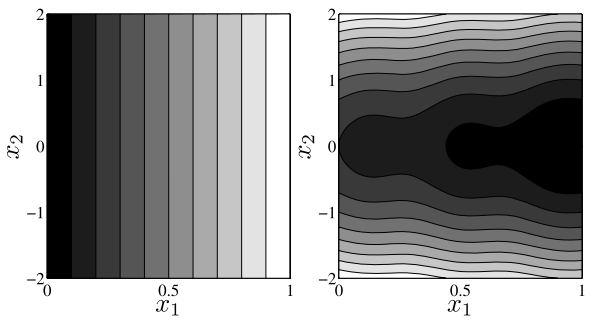


Fig. 4 Real contours of example (Ex-1)

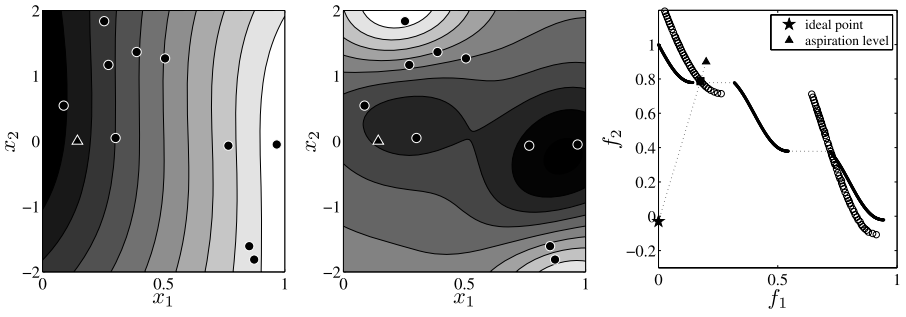


Fig. 5 # experimental data = 10 points initial learning

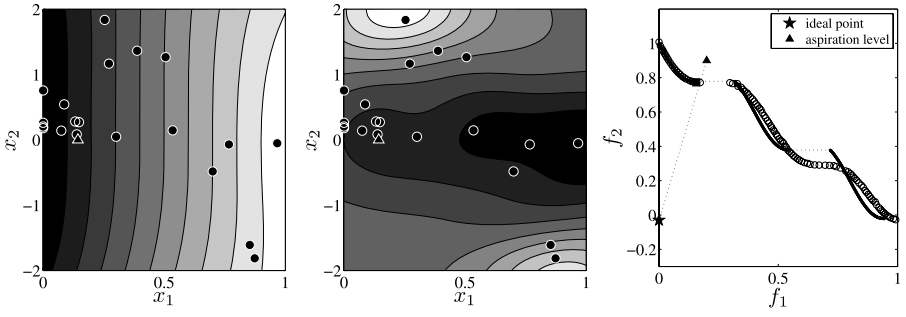


Fig. 6 # experimental data = 20 points after 5 additional learning

1. case of aspiration level 1.

(i) the number of generations: 100, the size of population: 100
(ii) the representation of chromosome: 20-bit
(iii) the probability of crossover: 1, the probability of mutation: 0.05
(iv) the point of crossover: 1- or 2-point

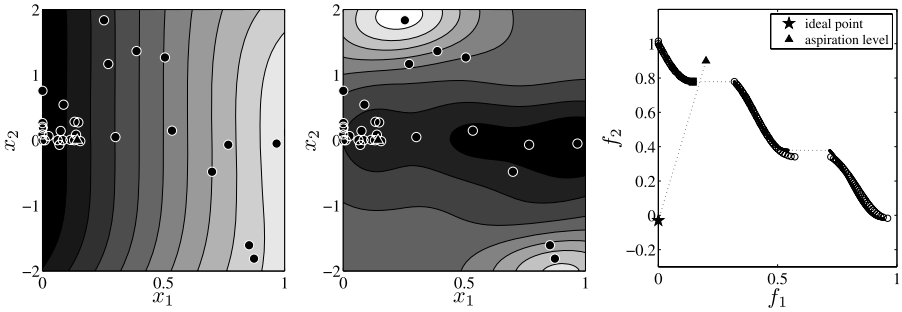


Fig. 7 # experimental data = 30 points after 10 additional learning

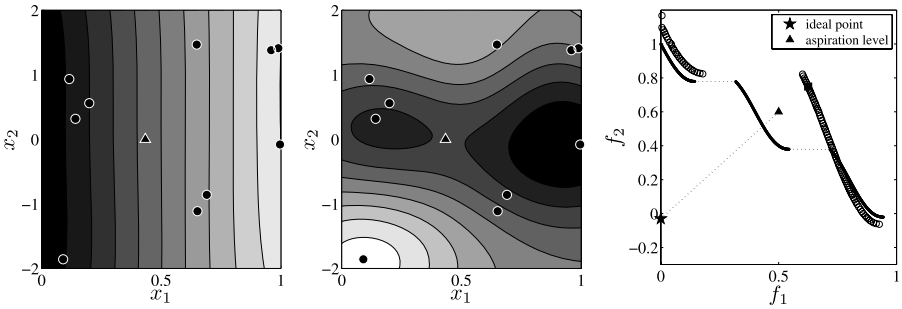


Fig. 8 # experimental data = 10 points initial learning

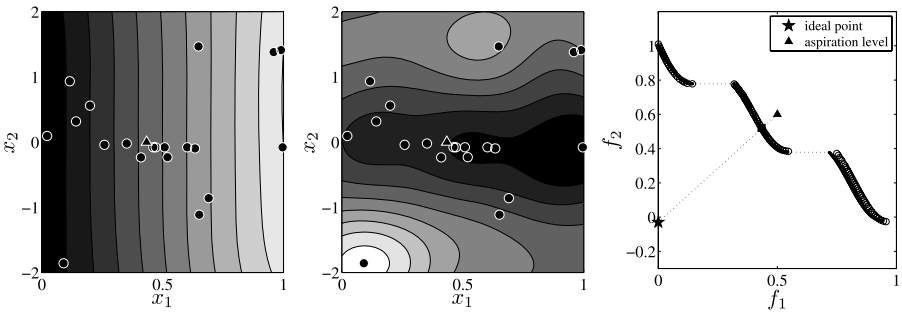
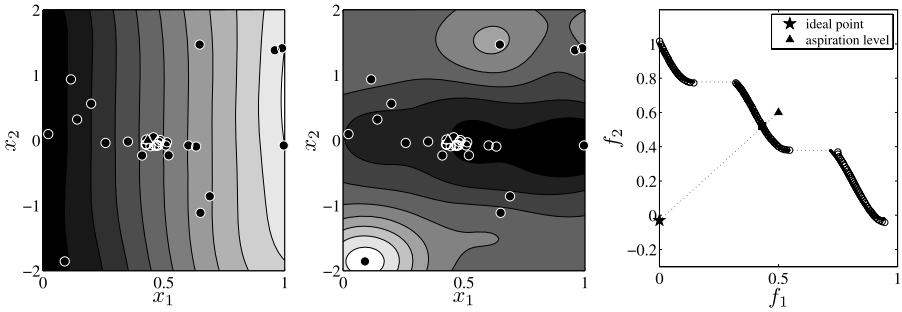


Fig. 9 # experimental data = 20 points after 5 additional learning

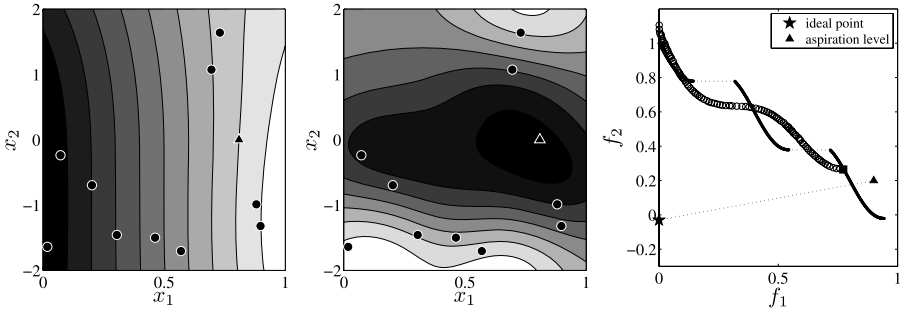
2. case of aspiration level 2.

As can be seen from Fig. 5, Fig. 8 and Fig. 11 of initial learning, since the sample points are few, the predicted objective function are different from the original objective functions. As an additional learning goes by, the accuracy for prediction has been improved gradually as shown in the figures.



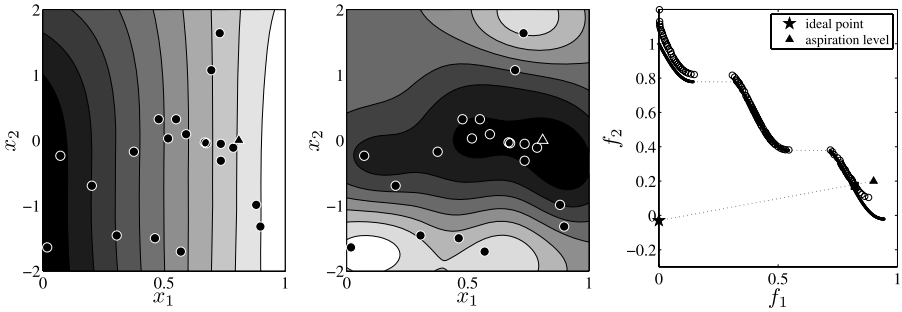
(a) f_1 (b) f_2 (c) Pareto optimal values

Fig. 10 # experimental data = 30 points after 10 additional learning



(a) f_1 (b) f_2 (c) Pareto optimal values

Fig. 11 # experimental data = 10 points initial learning

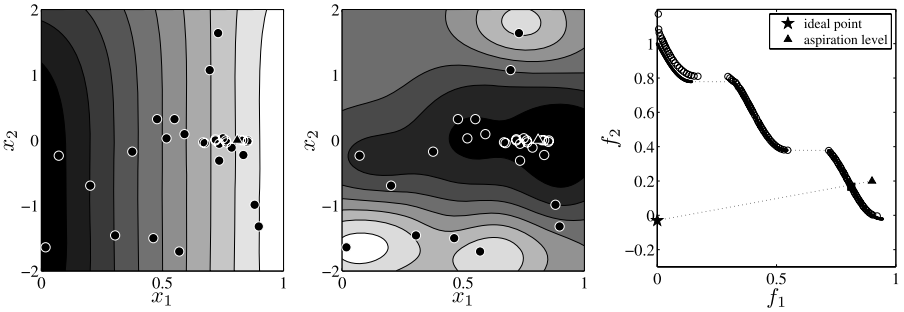


(a) f_1 (b) f_2 (c) Pareto optimal values

Fig. 12 # experimental data = 20 points after 5 additional learning

3. case of aspiration level 3.

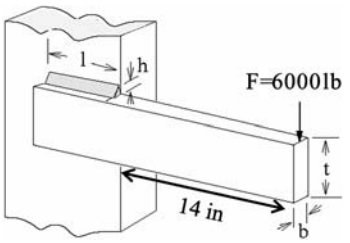
Next, we consider the welded beam design problem (Deb and Srinivasan 2005) shown in Fig. 14.



(a) f_1 (b) f_2 (c) Pareto optimal values

Fig. 13 # experimental data = 30 points after 10 additional learning

Fig. 14 Welded beam design problem



The problem is formulated as follows:

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

Here,

$$\tau = \sqrt{(\tau')^2 + (\tau'')^2 + \frac{l\tau'\tau''}{\sqrt{0.25(l^2+(h+t)^2)}}},$$

$$\tau' = \frac{6000}{\sqrt{2}hl},$$

$$\tau'' = \frac{6000(14+0.5l)\sqrt{0.25(l^2+(h+t)^2)}}{\sqrt{2}hl(\frac{l^2}{12}+0.25(h+t)^2)},$$

$$\sigma = \frac{504000}{t^2 b}, \qquad P_c = 64746.022(1 - 0.0282346t)tb^3.$$

The ideal value and aspiration level are given s follows:

$$\text{ideal value} := (f_1^*, f_2^*) = (0,\ 0),$$
$$\text{aspiration level 1} := (\overline{f}_1^1, \overline{f}_2^1) = (4,\ 0.003),$$
$$\text{aspiration level 2} := (\overline{f}_1^2, \overline{f}_2^2) = (20,\ 0.002),$$
$$\text{aspiration level 3} := (\overline{f}_1^3, \overline{f}_2^3) = (40,\ 0.0002).$$

Table 1 shows the result by the simple satisficing trade-off method using SQP and a quasi-Newton method (using MatLab Optimization toolbox) for randomly generated starting points in 10 times. Table 2 shows the result by our proposed method combining the satisficing trade-off method and optimization based on meta-modeling using μ–ν–SVR with 100 sample points. Since we used the usual gradient based optimization method for the simple satisficing trade-off method, the number of function evaluation would be almost 4 times because we have to apply the numerical differentiation on the based on the incremental difference for black box functions. From the viewpoint of the accuracy for obtained solutions, depending on the initial points, the proposed method provided worse solutions sometimes than the method by SQP and a quasi-Newton method without model prediction. Compared with Table 1, it seems that the solutions in Table 1 by the proposed method may not be so bad. Increasing the number of function evaluation, the proposed method can improve the accuracy of solution. However, as was mentioned in the introduction, our aim is how to make the number of function evaluations needed for finding a solution as few as possible. It can be considered that the proposed method can find sufficiently accurate solutions with a small number of function evaluations.

Table 1 Result by SQP using a quasi-Newton method without model prediction

		h	l	t	b	f_1	f_2	# evaluation
asp.	ave	0.5697	1.7349	10	0.5804	5.0102	3.78E–03	249.9
level	stdv	0.0409	0.1826	0	0.0072	0.0420	4.83E–05	69.6
1	max	0.5826	2.2546	10	0.5826	5.0235	3.92E–03	369.0
	min	0.4533	1.6772	10	0.5599	4.8905	3.77E–03	164.0
asp.	ave	1.0834	0.8710	10.0000	1.7685	13.7068	1.25E–03	204.2
level	stdv	0.3274	0.1662	5.11E–08	0.1828	1.3793	1.13E–04	30.1
2	max	2.0132	0.9896	10	2.1263	16.3832	1.31E–03	263.0
	min	0.9221	0.4026	10.0000	1.6818	13.0527	1.03E–03	172.0
asp.	ave	1.7345	0.4790	10	5	36.4212	4.39E–04	251.9
level	stdv	0.0000	0.0000	0	0	0.0000	5.71E–20	146.2
3	max	1.7345	0.4790	10	5	36.4212	4.39E–04	594.0
	min	1.7345	0.4790	10	5	36.4212	4.39E–04	112.0

Table 2 Result by the proposed method with 100 evaluations of function

		h	l	t	b	f_1	f_2
asp.	ave	0.6268	1.6066	9.9986	0.5874	5.0986	3.74E–03
level	stdv	0.0530	0.2000	0.0017	0.0141	0.0997	8.74E–05
1	max	0.7329	1.9674	9.9999	0.6175	5.2856	3.85E–03
	min	0.5491	1.2888	9.9949	0.5709	4.9743	3.56E–03
asp.	ave	0.7772	1.2457	9.9996	1.6756	13.0986	1.31E–03
level	stdv	0.1277	0.1989	0.0005	0.0125	0.1217	9.87E–06
2	max	1.0564	1.5370	10.0000	1.6882	13.2759	1.33E–03
	min	0.6284	0.8795	9.9983	1.6526	12.8691	1.30E–03
asp.	ave	0.7622	1.3313	9.9999	4.9992	37.6771	4.39E–04
level	stdv	0.1771	0.3150	0.0002	0.0021	0.6044	2.06E–07
3	max	1.1405	1.9566	10.0000	5.0000	38.9594	4.40E–04
	min	0.5177	0.7936	9.9994	4.9936	36.7259	4.39E–04

6 Conclusions

The most prominent feature in our proposed method is that it provides local and global information of Pareto frontier. Combining the aspiration level method and EMO, it is possible to find the most interesting part for the decision maker as well as to grasp the configuration of Pareto frontier. Furthermore, we have suggested how to choose additional sample points for sequentially predicting a form of objective functions. As a result, it is promising in practical problems since it has been observed through several numerical experiments that the method reduces the number of function evaluations up to less than $1/100$ to $1/10$ of usual methods. Usually, we do not know when to stop the computation in advance, and the computation is terminated relatively early by the limitation of time and cost in practical problem. In practical engineering design problems, it is desirable that the number of function evaluations is less than 1000 (100 if possible). Therefore, we can conclude that the proposed method is capable of wide application.

References

Chafekar D, Shi L, Rasheed K, Xuan J (2005) Multi-objective GA optimization using reduced models. IEEE Trans Syst Man Cybern Part C Rev Appl 9(2):261–265

Coello Coello CA, Van Veldhuizen DA, Lamont GB (2002) Evolutionary algorithms for solving multi-objective problems. Kluwer Academic, Dordrecht

Cristianini N, Shawe-Taylor J (2000) An introduction to support vector machines and other kernel-based learning methods. Cambridge University Press, Cambridge

Cortes C, Vapnik VN (1995) Support vector networks. Mach Learn 20:273–297

Deb K (2001) Multi-objective optimization using evolutionary algorithms. Wiley, New York

Deb K, Srinivasan A (2005) Innovization: innovative design principles through optimization. KanGAL Report #2005007, Indian Institute of Technology

Emmerich M, Giannakoglou K, Naujoks B (2006) Single and multi-objective evolutionary optimization assisted by Gaussian random field meta-models. IEEE Trans Evol Comput 10(4):421–439

Fonseca CM, Fleming PJ (1993) Genetic algorithms for multi-objective optimization: formulation, discussion and generalization. In: Proceedings of the fifth international conference on genetic algorithms, pp 416–426

Jin Y (2005) A comprehensive survey of fitness approximation in evolutionary computation. Soft-Comput 9(1):3–12

Jones DR, Schonlau M, Welch WJ (1998) Efficient global optimization of expensive black-box functions. J Glob Optim 13:455–492

Kiefer J (1959) Optimimum experimental designs. J R Stat Soc B 21(2):272–304

Nakayama H, Sawaragi Y (1984) Satisficing trade-off method for multi-objective programming. In: Interactive decision analysis. Springer, Berlin, pp 113–122

Nakayama H, Yun YB (2006a) Generating support vector machines using multiobjective optimization and goal programming. In: Multi-objective machine learning. Series on studies in computational intelligence. Springer, Berlin, pp 173–198

Nakayama H, Yun YB (2006b) Support vector regression based on goal programming and multiobjective programming. In: Proceedings of world congress on computational intelligence

Nakayama H, Arakawa M, Sasaki R (2001) A computational intelligence approach to optimization with unknown objective functions. In: Dorffner G, Bischof H, Hornik K (eds) Artificial neural networks (ICANN2001). Springer, Berlin, pp 73–80

Nakayama H, Arakawa M, Sasaki R (2002) Simulation based optimization for unknown objective functions. Optim Eng 3:201–214

Nakayama H, Arakawa M, Washino K (2003) Optimization for black-box objective functions. In: Pardalos PM, Tseveendorj I, Enkhbat R (eds) Optimization and optimal control. World Scientific, Singapore, pp 185–210

Sawaragi Y, Nakayama H, Tanino T (1985) Theory of multiobjective optimization. Academic Press, New York

Schölkopf B, Smola AJ (1998) New support vector algorithms. NeuroCOLT2 Technical report Series, NC2-TR-1998-031

Schölkopf B, Smola AJ (2002) Learning with kernels. MIT Press, Cambridge

Vapnik VN (1995) The nature of statistical learning theory. Springer, Berlin

Vapnik VN (1998) Statistical learning theory. Wiley, New York

Yun YB, Nakayama H, Tanino T, Arakawa M (2001) Generation of efficient frontiers in multi-objective optimization problems by generalized data envelopment analysis. Eur J Oper Res 129(3):586–595

Yun YB, Nakayama H, Arakawa M (2004) Multiple criteria decision making with generalized DEA and an aspiration level method. Eur J Oper Res 158(1):697–706

Yun YB, Nakayama H, Yoon M (2007) Sequential approximation method in multi-objective optimization by using computational intelligence. SICE Trans Soc Instrum Control Eng 43(8):672–678

Yoon M, Yun YB, Nakayama H (2004) Total margin algorithms in support vector machines. IEICE Trans Inf Syst E87(5):1223–1230

Zitzler E, Laumanns M, Thiele L (2001) SPEA2: improving the strength Pareto evolutionary algorithm. Technical Report 103, Computer Engineering and Networks Laboratory (TIK), Department of Electrical Engineering, Swiss Federal Institute of Technology (ETH)