Multi-objective Model Predictive Optimization using Computational Intelligence

Hirotaka Nakayama and Yeboon Yun

Abstract In many engineering design problems, the explicit function form of objectives/constraints can not be given in terms of design variables. Given the value of design variables, under this circumstance, the value of those functions is obtained by some simulation analysis or experiments, which are often expensive in practice. In order to make the number of analyses as few as possible, techniques for model predictive optimization (also referred to as sequential approximate optimization or metamodeling) which make optimization in parallel with model prediction have been developed. In this paper, we discuss several methods using computational intelligence for this purpose along with applications to multi-objective optimization under static/dynamic environment.

1 Brief Review of Model Predictive Methods

To begin with, we shall review several typical methods for model prediction. Response Surface Method (RSM) has been probably most widely applied to our aim [6]. The role of RSM is to predict the response y for the vector of design variables $\boldsymbol{x} = (x_1, \ldots, x_n)$ on the basis of the given sampled observations $(\tilde{\boldsymbol{x}}_i, \tilde{y}_i), i = 1, \ldots, \ell$.

Usually, Response Surface Method is a generic name, and it covers a wide range of methods. Above all, methods using design of experiments are famous. However, many of them use relatively low order (say, 1st or 2nd) polynomials on the basis of statistical analysis in design variable space. They may provide a good approximation

Hirotaka Nakayama
Department of Info. Sci. & Sys. Eng., Konan University, Kobe 658-8501, Japan, e-mail: nakayama@konan-u.ac.jp

Yeboon Yun
Department of Reliability-based Information Systems Engineering, Kagawa University,Takamatsu 761-0396, Japan, e-mail: yun@eng.kagawa-u.ac.jp

of black-box functions with a mild nonlinearity. It is clear, however, that in cases in which the black-box function is highly nonlinear, we can obtain better performance by methods using computational intelligence such as RBFN (Radial Basis Function Networks) or SVR (Support Vector Regression) taking into account not only the statistical property in design variable space but also that of range space of the black-box function (in other words, the shape of function).

In design of experiments, for example, D-optimality may be used for selecting a new additional sample to minimize the variance covariance matrix of the least squared error prediction. With the design matrix $\boldsymbol{X}$, this reduces to minimize the matrix $(\boldsymbol{X}^T\boldsymbol{X})^{-1}$ which is attained by maximizing det($\boldsymbol{X}^T\boldsymbol{X}$). This is the idea of D-optimality in design of experiments.

Other criteria are possible: to minimize the trace of $(\boldsymbol{X}^T\boldsymbol{X})^{-1}$ (A-optimality), to minimize the maximal value of the diagonal components of $(\boldsymbol{X}^T\boldsymbol{X})^{-1}$ (minimax criterion), to maximize the minimal eigen value of X^TX (E-optimality). In general, D-optimality criterion is widely used for many practical problems.

Jones *et al.* [5] suggested a method called EGO (Efficient Global Optimization) for black-box objective functions. They applied a stochastic process model for predictor and the expected improvement as a figure of merit for additional sample points. Regard y as a realized value of the stochastic variable Y, and let $f^p_{\min}$ be the minimal value of p-samples which are evaluated already. For minimization cases, the improvement at $\boldsymbol{x}$ is $I = \max\bigl(f^p_{\min} - Y,\ 0\bigr)$. Therefore, the expected improvement is given by

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

We select 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, we can select the best one among several candidates which are generated randomly in the design variable space. It has been observed through our experiences that this method is time consuming.

2 Using Computational Intelligence

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 [7], [8]. There, additional sample points are selected by considering both global and local information of the black-box function.

Support vector machine (SVM) has been recognized as a powerful machine learning technique. SVM was originally developed for pattern classification and later extended to regression ([1], [13]). 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.

This task is performed by solving some type of mathematical programming such as quadratic programming (QP) or linear programming (LP).

Linear classifiers on the basis of goal programming, on the other hand, were developed extensively in 1980's [3], [4]. The authors developed several varieties of SVM using multi-objective programming and goal programming (MOP/GP) techniques [10]. 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 $(i = 1, \dots, \ell)$ 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. [12]. Although many variants are possible, $\mu-\nu-$SVM considering the objectives i) and ii) is promising, because $\mu-\nu-$SVM for regression has been observed to provide a good sparse approximation [10].

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

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

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:

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

It has been observed through our experiences that $\mu-\nu-$SVR provides the least number of support vectors among existing SVRs. This implies that $\mu-\nu-$SVR can be effectively applied for selecting a new sample on the basis of information of support vector.

3 Using global and local information for adding new 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.

If the current optimal point is taken as such additional data, the estimated optimal point tends to converge to a local maximum (or minimum) point. This is due to lack of global information in predicting the objective function.

On the other hand, if additional data are taken far away from the existing data, it is difficult to obtain more detailed information near the optimal point. Therefore, it is hard to obtain a solution with a high precision. This is because of insufficient information near the optimal point.

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 [7]. Namely, two kinds of additional samples are taken simultaneously 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 converging to local maximum (or minimum) point.

4 Multi-objective Model Predictive Optimization: Static Cases

In multi-objective optimization, the so-called Pareto solution is introduced. 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 judgment. Among them, the aspiration level approach is now recognized to be effective in practice. As one of aspiration level approaches, one of authors proposed the satisficing trade-off method [9]. Suppose that we have objective functions $\boldsymbol{f}(\boldsymbol{x}) := (f_1(\boldsymbol{x}), \ldots, f_r(\boldsymbol{x}))$ 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:

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

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

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

Now, we propose a method combining the satisficing trade-off method for interactive multi-objective programming and the sequential approximate optimization using $\mu{-}\nu{-}$SVR. In the following, we explain the method along an example of the welded beam design problem [2] shown by Fig. 1. The problem is formulated as follows:

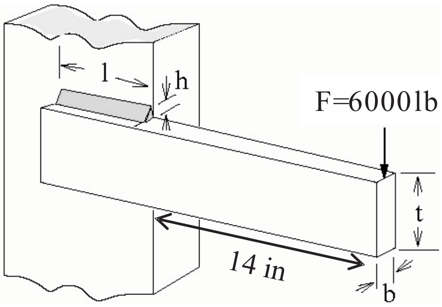

Fig. 1 Welded beam design problem

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

Here,

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

The ideal value and aspiration level are given s follows:

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

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

		h	l	t	b	f_1	f_2	# evaluation
asp. level 1	average	0.5697	1.7349	10	0.5804	5.0102	3.78E-03	249.9
	stdv	0.0409	0.1826	0	0.0072	0.0420	4.83E-05	69.6
	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. level 2	average	1.0834	0.8710	10.0000	1.7685	13.7068	1.25E-03	204.2
	stdv	0.3274	0.1662	5.11E-08	0.1828	1.3793	1.13E-04	30.1
	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. level 3	average	1.7345	0.4790	10	5	36.4212	4.39E-04	251.9
	stdv	0.0000	0.0000	0	0	0.0000	5.71E-20	146.2
	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. level 1	average	0.5223	1.9217	9.9934	0.5825	5.0344	3.78E-03
	stdv	0.0374	0.1656	0.0136	0.0011	0.0130	1.08E-05
	max	0.5832	2.2742	10	0.5845	5.0692	3.81E-03
	min	0.4520	1.6859	9.9558	0.5817	5.0224	3.77E-03
asp. level 2	average	0.8921	1.0398	9.9989	1.6809	13.0653	1.31E-03
	stdv	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
asp. level 3	average	2.2090	0.4486	10	5	36.6830	4.39E-04
	stdv	0.9355	0.2293	0	0	0.2695	5.71E-20
	max	3.7812	0.8734	10	5	37.1257	4.39E-04
	min	1.0391	0.1895	10	5	36.4212	4.39E-04

Table 1 shows the result by the simple satisficing trade-off method using SQP and a quasi-Newton method for randomly chosen starting points in 10 times. Table 2 shows the result by our proposed method combining the satisficing trade-off method and the model predictive optimization 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 for black box functions because we have to apply the numerical differentiation on the based on the incremental difference.

5 Multi-objective Model Predictive Optimization: Dynamic Cases

For dynamic optimization problems, the model predictive control has been developed along a similar idea to the above. Let $\boldsymbol{u}(t),\ \boldsymbol{x}(t)$ denote the control (input) vector and the state vector, respectively. Our problem is represented by

$$\text{Minimize} \qquad J = \phi[\boldsymbol{x}(T)] + \int_0^T F(\boldsymbol{x}, \boldsymbol{u}, t)dt \tag{3}$$

$$\text{subject to} \qquad \dot{\boldsymbol{x}} = f(\boldsymbol{x}(t), \boldsymbol{u}(t), t),\ \ \boldsymbol{x}(0) = \boldsymbol{x}_0 \tag{4}$$

If the function form in the above model is explicitly given, then we can apply some techniques in optimal control theory. However, we assume that some of function forms, in particular the dynamic system equation (4), can not explicitly be given. Under this circumstance, we predict some of future state $\boldsymbol{x}(t+1), \ldots, \boldsymbol{x}(t+k)$ for given $\boldsymbol{u}(t+1), \ldots, \boldsymbol{u}(t+p)$. The period $[t+1, t+k]$ is called the prediction period, and $[t+1, t+p]$ the control period. Our aim is to decide the optimal control sequence $\boldsymbol{u}(t)$ over $[0, T]$.

For predicting the future state, we apply a support vector regression technique, namely $\mu - \nu-$SVR which was stated in the previous section. In the following, we show a numerical result by using the satisficed trade-off method with model prediction. Our problem to be considered in this paper has multiple objectives

$$J = (J_1, \ldots, J_r).$$

For example, those objectives are the energy consumption, constraints of terminal state, the terminal time (T) itself and so on.

Step 1. Predict the model f based on $(x(k), u(k), x(k+1)),\ \ k = 0, 1, \ldots, t-1,\ x(0) = x_0$.

Step 2. Generate individuals of control sequence by GA algorithm

$$u_i(t),\ u_i(t+1), \ldots, u_i(T-1),\ \ i = 1, 2, \ldots, N_{population}.$$

- Predict the state resulting from each control sequence from the present time to the terminal time

$$x(k+1) = f(x(k), u(k)),\ k = t, t+1, \ldots, T-1,\ \ x(0) = x_0. \tag{5}$$

- Evaluate each individual in terms of auxiliary scalar objective function of satisficing trade-off method

$$F = \max_{1 \leqq i \leqq r} w_i \left(J_i(x) - \overline{J}_i^k \right) + \alpha \sum_{i=1}^{r} w_i \left(J_i(x) - \overline{J}_i^k \right)$$

$$J_1 = T,\ J_2 = \sum_{k=t}^{T-1} u^2(k).$$

- Select the best individual (control sequence) u^*. Calculate $x(t+1)$ by (5) using $x(t)$ and $u(t) = u^*(t)$.

Step 3. $t \leftarrow t + 1$ and go to Step 2.

The solutions for two different aspiration levels are depicted in Fig. 2. It may be seen that the proposed method provides reasonable solutions flexibly depending on the apiration levels.

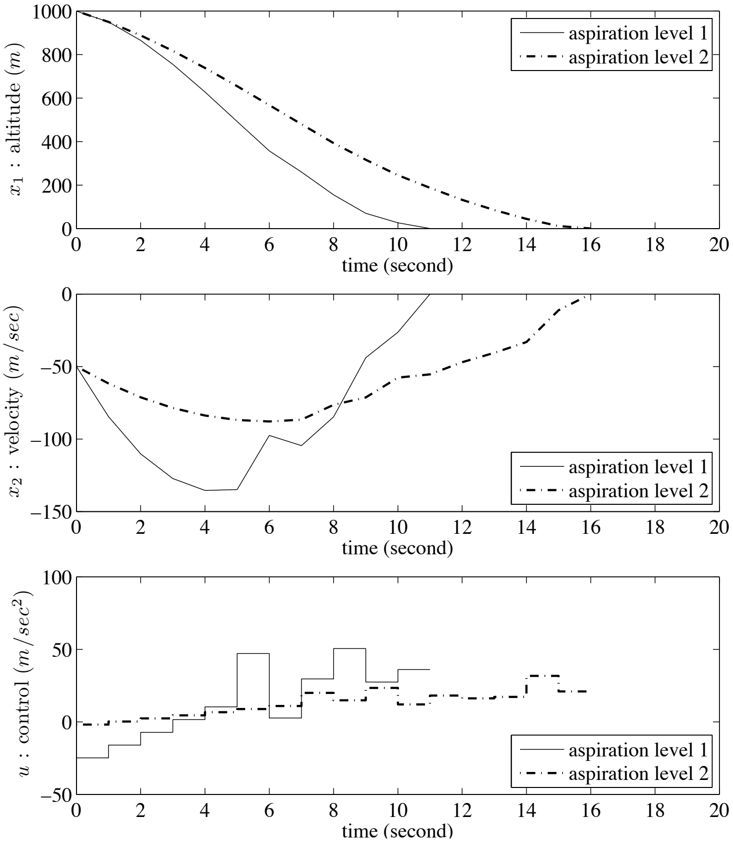

Fig. 2 Multi-objective model predictive control

6 Concluding Remarks

We discussed methods combining the satisficing trade-off method and model predictive optimization methods using computational intelligence under static and dynamic environment. The proposed method provides an approximate Pareto solution closest to the given aspiration level. It is promising in practical problems since it has been observed through several numerical experiments that the method reduces the number of function evaluation up to less than $1/100$ to $1/10$ of usual methods.

References

1. Cortes, C. and Vapnik, V., (1995) Support Vector Networks, *Machine Learning*, **20**, pp. 273–297
2. Deb, K. and Srinivasan, A., (2005) Innovization:Innovative Design Principles Through Optimization, KanGAL Report #2005007, Indian Institute of Technology
3. Erenguc, S.S. and Koehler, G.J., (1990) Survey of Mathematical Programming Models and Experimental Results for Linear Discriminant Analysis, *Managerial and Decision Economics*, **11**, 215-225
4. Freed, N. and Glover, F., (1981) Simple but Powerful Goal Programming Models for Discriminant Problems, *European J. of Operational Research*, **7**, 44-60
5. Jones, D.R., Schonlau, M. and Welch, W.J., (1998) Efficient Global Optimization of Expensive Black-Box Functions, *J. of Global Optimization*, **13** , 455-92
6. Myers, R.H. and Montgomery, D.C., (1995) *Response Surface Methodology: Process and Product Optimization using Designed Experiments*, Wiley
7. Nakayama, H., Arakawa, M. and Sasaki, R., (2002) Simulation based Optimization for Unknown Objective Functions, *Optimization and Engineering*, **3**, 201-214
8. 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
9. 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
10. Nakayama, H. and Yun, Y., (2006) Generating Support Vector Machines using Multiobjective Optimization and Goal Programming *Multi-objective Machine Learning*, Yaochu Jin(ed.), Springer Series on Studies in Computational Intelligence, pp.173-198.
11. Nakayama, H. and Yun, Y., (2006) Support Vector Regression Based on Goal Programming and Multi-objective Programming, IEEE World Congress on Computational Intelligence, CD-ROM (Paper ID : 1536)
12. Schölkopf, B. and Smola, A.J., (1998) New Support Vector Algorithms, *NeuroCOLT2 Technical report Series*, NC2-TR-1998-031
13. Vapnik, V.N., (1998) *Statistical Learning Theory*, John Wiley & Sons, New York
14. Yoon, M., Yun, Y.B. and Nakayama, H., (2003) A Role of Total Margin in Support Vector Machines, *Proc. IJCNN'03*, 2049-2053