Int J Adv Manuf Technol (2009) 41:440–451
DOI 10.1007/s00170-008-1491-8

ORIGINAL ARTICLE

Optimal cutting condition determination for desired surface roughness in end milling

Chakguy Prakasvudhisarn · Siwaporn Kunnapapdeelert · Pisal Yenradee

Received: 8 March 2007 / Accepted: 25 March 2008 / Published online: 16 May 2008

Abstract CNC end milling is a widely used cutting operation to produce surfaces with various profiles. The manufactured parts' quality not only depends on their geometries but also on their surface texture, such as roughness. To meet the roughness specification, the selection of values for cutting conditions, such as feed rate, spindle speed, and depth of cut, is traditionally conducted by trial and error, experience, and machining handbooks. Such empirical processing is time consuming and laborious. Therefore, a combined approach for determining optimal cutting conditions for the desired surface roughness in end milling is clearly needed. The proposed methodology consists of two parts: roughness modeling and optimal cutting parameters selection. First, a machine learning technique called support vector machines (SVMs) is proposed for the first time to capture characteristics of roughness and its factors. This is possible due to the superior properties of well generalization and global optimum of SVMs. Next, they are incorporated in an optimization problem so that a relatively new, effective, and efficient optimization algorithm, particle swarm optimization (PSO), can be applied to find optimum process parameters. The cooperation between both techniques can achieve the desired surface roughness and also maximize productivity simultaneously.

Keywords Metrology · Surface roughness · Milling operation · Support vector machines · Particle swarm optimization

1 Introduction

Fiercely global competition has forced manufacturers to find ways to increase productivity as well as to improve quality. Automated manufacturing systems are widely employed in machining along with computer numerical control (CNC) machines to achieve such goals. Among several CNC machining processes, end milling is one of the most conventional and commonly used cutting operations for producing surfaces with various profiles. The quality of the manufactured parts not only depends on their geometries but also their surfaces textures such as roughness and waviness. Surface roughness affects friction, wear, fatigue, corrosion, and electrical and thermal conductivity [1]. Therefore, the desired surface finish is usually specified in advance and the proper process with various settings is repeatedly attempted until the required quality is obtained.

Several factors such as feed rate, spindle speed, depth of cut, tool geometry, tool wear, chip loads and chip formations, coolant, and material properties of both tool and workpiece influence surface roughness. Only some of these factors can be controlled or set up in advance. In addition, the selection of cutting parameters' values is traditionally conducted by trial and error, experience, and machining handbooks. Subsequently, various sets of cutting conditions are repeated to achieve a desired roughness. This results in cutting condition which may compromise quality

C. Prakasvudhisarn (✉)
School of Technology, Shinawatra University,
Shinawatra Tower III,
15th floor, 1010 Viphavadi Rangsit Rd., Chatuchak,
Bangkok 10900, Thailand
e-mail: chakguy@shinawatra.ac.th

S. Kunnapapdeelert · P. Yenradee
School of Manufacturing Systems and Mechanical Engineering,
Sirindhorn International Institute of Technology,
Thammasat University,
Pathum Thani, Thailand

of the obtained products and/or efficiency of the end milling operation. Moreover, such empirical processing is very time consuming and laborious. Generally, the formation of surface roughness is very complicated and process dependent. Thus, an analytical approach for its modeling is very difficult due to its nonlinearity. Various theoretical models that have been proposed so far are not accurate enough for wide range of cutting conditions. Thus, there is a need for a tool that can accurately predict the surface roughness of a product and also select optimum machining parameters while minimizing cutting time before performing the actual milling operation. Hence, improvement of parts quality and productivity can be obtained. Furthermore, this tool should be easy to use, fast, and highly reliable.

Many researchers have studied various relationships between surface roughness and its relevant factors, such as spindle speed, feed rate, depth of cut, tool diameter, workpiece hardness, deflection and chatter of the workpiece-tool system for the end milling process. Techniques such as the closed-form relationship model, multiple regression, neural network, neural-fuzzy system have been investigated for determination of cutting conditions with good predictive outcomes. They can be roughly categorized into two groups, analytical and numerical methods. The former consists of closed-form prediction models of several components of a cutting operation such as cutting force system, cutter deflection, and workpiece deflection. Such a study was investigated by Kline et al. [2] with good prediction results of surface errors. The latter can also be further classified into two subgroups, traditional statistics-based approach and computational intelligence-based approach. The statistically based methods have been shown to perform very well for roughness prediction. Lou et al. [3] determined correlation between roughness and relevant parameters, spindle speed, feed rate, and depth of cut by using multiple regression. Mansour and Abdalla [4] developed the correlation model from cutting speed, feed rate, and depth of cut via design of experiment and response surface methodology, which is similar to the work by Arbizu and Perez [5]. In addition, Feng and Wang [6] discussed a model for turning based on workpiece hardness, feed rate, tool point angle, depth of cut, spindle speed, and cutting time. Taylor tool life equation was first transformed into simpler forms. Then, multiple regression analysis was used to establish the prediction model. The reasons why statistically based methods are quite widely used are that they are relatively simple, robust, objective, and consistent. However, their major drawbacks are the requirement of data's distribution assumption and the use of trial and error procedure for unknown relationships of data. These limit the usefulness of this approach. For these reasons, computational intelligence-based approach has recently gained more acceptance. Several methods from artificial intelligence, evolutionary computation, and fuzzy systems have been introduced to model metal cutting processes. A hybrid method such as a combined neural network and fuzzy system was applied by Lou and Chen [7] with spindle speed, feed rate, depth of cut, and vibration as parameters. The fuzzy rules bank was built by learning from various cutting conditions. A similar work was also conducted in Lou and Chen [8]. In addition, Lo [9] investigated the effectiveness of an adaptive-network based on fuzzy inference system with common cutting conditions such as spindle speed, feed rate, and depth of cut. Feedforward neural networks were also modeled to predict both surface roughness and tool flank wear for various cutting conditions in finish dry hard turning [10]. El-Mounayri et al. [11] used swarm intelligence to find proper values of coefficients in a widely accepted model representing a relationship between roughness and spindle speed, feed rate, and depth of cut. In those works, their results confirm the effectiveness of computational intelligence-based methods. A major shortcoming of using optimization-based methods is that models describing relationships between roughness and cutting parameters must be known beforehand, which is normally true for merely a few parameters. In addition, they cannot be easily extended to cover other cutting parameters for more comprehensive and realistic models. By the same token, fuzzy logic-based methods are usually difficult to build the rule set and to choose the proper membership function and its ranges. With high competition in the manufactured part market such as automotive industry and the increasing need for more accurate and practical evaluation systems, techniques in artificial intelligence, particularly artificial neural networks (ANNs), receive more attention in the industry and academia because they can be used to learn relationships of roughness and its parameters. However, a number of parameters such as network topology, learning rate, and training methods have to be fine tuned before they can be deployed successfully. Furthermore, drawbacks like local optima, overfitting, and long learning time tend to occur.

Theoretically, the aforementioned shortcomings of ANNs have been countered by the development of support vector machines (SVMs). Unlike ANNs which minimize empirical risk, SVMs are designed to minimize the structural risk by minimizing an upper bound of the generalization error rather than the training error. Therefore, the overfitting problem in machine learning can be solved successfully. Another outstanding property of SVMs is that the task of training SVMs can be mapped to a uniquely solvable linearly constrained quadratic programming problem. This produces a solution that is always unique and globally optimal. They have been extended to solve regression problems as well.

In this paper, support vector regression (SVR), which is based on support vector machines (SVMs), is investigated as an alternative technique for roughness prediction. It has shown very good results in various fields, such as optical character recognition in machine vision [12] and function approximation of quantitative structure-activity relationships (QSAR) in medicinal chemistry [13]. It was also modified to estimate minimum zone tolerances in manufacturing [14]. The SVR retains much of the elegance of the SVMs such as good generalization and global optimum properties and no normal distribution assumption requirement. The linear approximation is a fundamental concept of SVR. Its extension to nonlinear case can be achieved by using the mechanism of inner-product kernel to avoid the curse of dimensionality. To speed up its regression, the use of proper kernel can be calculated in advance. Even though this kernel computation requires large memory space, various relief efforts have already been proposed [15, 16].

So far, no study has yet to focus on surface roughness modeling and prediction by using SVR. Moreover, in practice, efficient operation of CNC end milling requires various goals simultaneously, including desired roughness of the machined surface and minimum machining time. Thus, the effects of process conditions on the resultant surface quality and the cutting speed of the end milling operation can be formulated as an optimization problem for determination of optimum machining parameters. Tanden et al. [17] presented a similar approach by attempting to optimize production cost for end milling with the combined feedforward neural network and the particle swarm optimization (PSO). The neural network was used to predict cutting forces which in turn were used in the formulated optimization model as a constraint. The PSO also showed good performance and was suitable for use with the found model where no explicit relation between inputs and outputs was available. With attractive properties of no requirement on gradient information, consistent results, and fast convergence, and successfully machining applications in [11, 17], the PSO is then selected as an optimizer in this work.

Therefore, the purpose of this study is to develop a procedure that can determine optimal cutting condition in CNC end milling for desired surface roughness and minimizing machining time simultaneously. There are two main steps in this procedure. Firstly, a surface roughness predictive model based on the SVR is developed and evaluated by using experimental data from published references. Secondly, optimum process parameters are determined based on the need of the user on a required surface roughness and the need to maximize productivity by integrating the SVR-based roughness model in an optimization problem to be solved by the PSO.

2 Methodology

Section 2.1 discusses on some selected techniques namely support vector regression, neural network, and particle swarm optimization for roughness modeling. Next, Sect. 2.2 describes how cutting conditions for desired surface roughness can be determined while maximizing productivity at the same time.

2.1 Prediction of surface roughness

2.1.1 Support vector machine-based roughness model

Support vector machines (SVMs) represent a relatively new type of learning machine. They are an approximate implementation of the method of structural risk minimization which attempts to minimize the generalization error occurred when the machines are tested with unseen data. The generalization error rate is bounded by the sum of a pair of competing terms, the training error rate and the confidence interval which depends on the Vapnik–Chervonenkis (VC) dimension. Hence, the *VC* dimension and the training error (empirical risk) are both minimized at the same time. To realize this in SVMs, a structure is imposed on the set of hyperplanes by trying to obtain the weight vector **w** having the minimum Euclidean norm. Coupled with dual transformations, the optimization model yields a global optimum. These key properties really separate the SVMs from other learning machine algorithms. The initial applications of SVMs are optical character recognition [12], regression, and time series predictions [18, 19].

In regression problems, the problem of approximating the following set of data $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_1, y_1)\} \subset \Re^n \times \Re$ with a linear function $f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b$, where $\mathbf{w} \in \Re^n$, $b \in \Re$, and $\langle ., . \rangle$ represents dot product, is taken into consideration. The $\mathbf{x}_i$ is the set of cutting conditions and y_i is the output which is the roughness value. The ε-insensitive loss function proposed by Vapnik [20] is commonly incorporated with SVMs (ε-SVR) to create sparseness in the support vectors and to embed robustness of Huber's loss function. This means that $f(\mathbf{x})$ is allowed to vary at most ε deviation from the target and is as flat as possible simultaneously. If the deviations are larger than a priori ε specified, this implies a bad fit and this function is proportionally penalized with a priori constant C. This constant C determines the trade off between the training errors and model complexity. The flattest of $f(\mathbf{x})$ is accomplished by searching the smallest **w**. Hence, a formulation of ε-SVR can be described by:

$$\min \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{l}\left(\xi_i + \xi_i^*\right) \tag{1}$$

subject to

$$y_i - \langle \mathbf{w}, \mathbf{x}_i \rangle - b \leq \varepsilon + \xi_i$$
$$\langle \mathbf{w}, \mathbf{x} \rangle + b - y_i \leq \varepsilon + \xi_i^*$$
$$\xi_i, \xi_i^* \geq 0$$

Everything above ε is captured in slack variables ξ_i and everything below -ε is captured in slack variables ξ_i^*. This ε-insensitive loss function, $|\xi|_\varepsilon$, is depicted in Fig. 1 and is defined as

$$|\xi|_\varepsilon = \begin{cases} 0 & ;\text{if}|\mathrm{f}(\mathbf{x}) - y| < \varepsilon \\ |f(\mathbf{x}) - y| - \varepsilon & ;\text{otherwise} \end{cases}. \tag{2}$$

Using the Lagrangian multipliers and the Karush–Kuhn–Tucker (KKT) conditions, one obtains the following dual problem:

$$\max_\alpha -\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\left(\alpha_i - \alpha_i^*\right)\left(\alpha_j - \alpha_j^*\right)\langle \mathbf{x}_i, \mathbf{x}_j \rangle -\varepsilon\sum_{i=1}^{l}\left(\alpha_i + \alpha_i^*\right) + \sum_{i=1}^{l} y_i\left(\alpha_i - \alpha_i^*\right) \tag{3}$$

subject to

$$\sum_{i=1}^{l}\left(\alpha_i - \alpha_i^*\right) = 0$$

and

$$\alpha_i, \alpha_i^* \in [0, C] \quad , i = 1, \ldots, l.$$

Transforming into dual form yields a quadratic programming problem with linear constraints and positive definite Hessian matrix. This leads to a global optimum. A nonlinear form is usually required to adequately model surface roughness data. Hence, a nonlinear mapping, φ, can be used to map data from input space into a higher dimensional intermediate (or feature) space where linear regression can be performed. Consequently, major hurdles result in the complexity of φ and the curse of dimensionality in Eq. (3). To alleviate these difficulties, the inner-product kernel is then introduced as follows:

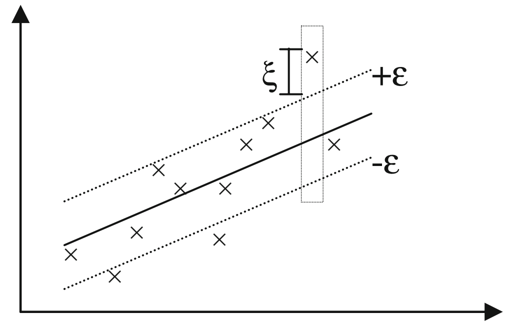

Fig. 1 The soft margin tube

$$K\left(\mathbf{x}_i, \mathbf{x}_j\right) = \left\langle \varphi\left(\mathbf{x}_i\right), \varphi\left(\mathbf{x}_j\right) \right\rangle.$$

The dimensionality of the feature space is thus hidden from the remaining computations. Some of the most widely used kernels such as linear, polynomial, and Gaussian radial basis functions were tested in this study. The kernel function can be employed in the optimization models above by replacing $\langle .,. \rangle$ with $K(.,.)$. This adds the capability to approximate both linear and nonlinear functions.

In summary, the main advantages of SVR are implicit mapping by using kernels in handling nonlinear data, convexity of quadratic optimization, and generalization properties. In addition, distribution of the data used is not necessarily assumed in advance, which makes it very promising for the real-world problems.

2.1.2 Neural networks-based roughness model

Most of the aforementioned advantages of the SVR are very similar to those of neural networks. Hence, a feed-forward backpropagation neural network (NN) was attempted to predict surface roughness with the same datasets. A neural network normally has two elementary components, processing elements (or processing nodes) and connection weights. A feedforward architecture specifies that the network has no loops as opposed to feedback architecture. A classical learning algorithm, backpropagation, was used to train and update the weights on each link of a neural network with training examples. These weights capture the relationship pattern of multivariable function through learning. In other words, they were used to capture the relationships between various cutting conditions and surface roughness. Weight adjustment between processing nodes in backpropagation is carried out according to the difference between the target value and the output value of the neural network. The difference of the error is measured by the mean square error shown below:

$$E = \sum_{p=1}^{P}\sum_{k=1}^{K}\left(d_{pk} - o_{pk}\right)^2 \tag{4}$$

where d_{pk} is the k^{th} desired value of the p^{th} data and o_{pk} is the actual output.

The weights ($\mathbf{W}$) are adjusted toward the gradient direction that produces a smaller approximation error as follows:

$$\mathbf{W} = \mathbf{W} + \eta\delta y \tag{5}$$

where η is a positive constant called learning rate, δ is the gradient of the difference between the desired and actual neuron's responses, and $\mathbf{y}$ is the input vector.

2.1.3 Particle swarm optimization-based roughness model

The widely accepted analytical model below was suggested by [11] to predict surface roughness:

$$Ra = \frac{10aR^bF^c}{S^d} \tag{6}$$

where Ra is surface roughness; R is the radial depth of cut; F is the feed factor; and S is the spindle speed; coefficients a, b, c, and d are the unknowns to be determined by the PSO algorithm. The PSO was introduced by Kennedy and Eberhart [21] to imitate social behavior of animals such as bird flocking in searching for food. Each particle flies in hyperspace searching for the best solution by adjusting position and velocity based on its own flying experience (*pbest*) and its companions' experience (*gbest*). The inertia weight w was later introduced to improve the PSO optimizer. It is very attractive because gradient information is not needed. Hence, it is unaffected by discontinuities of the objective function. The equations used consist of flexible and well-balanced mechanisms to enhance the global and local exploration abilities. These allow a thorough search and simultaneously avoid the premature convergence. In addition, PSO uses probabilistic rules for particle's movements. Therefore, it is quite robust to local optima. The steps of the PSO are well documented in [21, 22]. The regression model for roughness prediction could be developed by minimizing the sum of the errors between the actual and the predicted outputs while searching the optimal values for those unknown coefficients in Eq. (6). The optimization model can be described as follows:

$$\min \quad Error = \sum_i \left| Ra_i - \widehat{Ra}_i \right| \tag{7}$$

where Ra is the experimental surface roughness and $\widehat{Ra}$ is the predicted surface roughness [11].

The PSO-based model was formulated to determine a near-optimal set of a, b, c, and d. A population of 40 particles was used to search those unknowns (four dimensions) for 300 iterations [11]. Inertia weight was linearly decreased from 0.9 to 0.4. The acceleration constants represent the weighing of the stochastic terms that pull each particle toward *pbest* and *gbest* positions. They were set to 2.0 to give it a mean of 1 for the cognition and social parts, so that the particles would thoroughly search the settled regions [21]. Particles' velocities on each dimension are normally clamped to a maximum velocity to control the exploration ability of particles. This velocity was fixed at 0.1. The particle with the best performance was then picked as a solution. The PSO consists of five main components, position of particle on each dimension, particle's velocity on each dimension, particle's personal best value, position of the particle with personal best value, and group's best value (index of the particle having the best value among all particles). One-dimensional and two-dimensional arrays were used to handle the above components.

2.2 Optimization of machining parameters

In practice, efficient operation of CNC end milling requires various goals simultaneously, including desired roughness of the machined surface and minimum machining time. Once a reliable roughness prediction model is obtained, it can be used not only to evaluate cutting parameters but also to determine the proper process parameters. An optimal selection of these parameters can be accomplished by formulating them as an optimization problem. The objective function was to minimize machining time by maximizing feed rate subject to the surface roughness required and limits of roughness parameters employed by the prediction model:

$$\max \; x_F \tag{8}$$

subject to

$$\begin{aligned} &\widehat{Ra}(\mathbf{x}) \le d \\ &x_i \ge lb_i\,;\ \text{for } i = F, S, D \\ &x_i \le ub_i\,;\ \text{for } i = F, S, D \\ &x_i \ge 0\,;\ \text{for } i = F, S, D \end{aligned}$$

where

x_F	represents feed rate decision variable in inch per minute (ipm),
x_S	represents spindle speed decision variable in revolution per minute (rpm),
x_D	represents depth of cut decision variable in inch,
lb_i and ub_i	represent lower bound and upper bound of corresponding machining parameter i,
$\hat{Ra}(\mathbf{x})$	represents predictive roughness function with decision variable vector $\mathbf{x}$=(x_F,x_S,x_D), and
d	is specified roughness in microinch.

Since SVR was utilized as the model for surface roughness, gradient-based nonlinear programming methods were not practical. The PSO was the method of choice for the above optimization problem due to its direct fitness information instead of functional derivatives and other advantages mentioned in Subsect. 2.1.3. The PSO must be additionally constrained to handle this optimization problem. Some simple and effective modifications were done in the PSO by initiating and updating particles in feasible region [23]. Otherwise, each particle would loop until it was in the feasible space. The optimum feed rate, spindle speed, and depth of cut were determined by using a population of 40 particles searching for 350 iterations. The same inertia weight, acceleration constants, and maximum velocity as above were simply employed.

3 Numerical modeling

Surface roughness is dependent upon various factors, including spindle speed, feed rate, depth of cut, tool geometry, tool wear, and material properties of both tool and workpiece. Each affects surface finish in varying degree. Only common factors such as spindle speed, feed rate, and depth of cut were chosen to construct a roughness model due to their high impact and ease of acquisition. They were clearly shown to be appropriate for a successful prediction of surface roughness [3, 9, 11]. The previously published data from [3, 9, 11] were used to train and test the proposed method since those work represent excellent prediction results.

The experiments for these datasets were conducted by using Fadal CNC vertical machining center with 6061 aluminum [3, 8, 11]. Summary of each data set is illustrated in Table 1. It is important to note that the depth of cut in those datasets represented slightly different cutting conditions in their experiments. The radial depth of cut was employed in [11], whereas the axial depth of cut was experimented in [3] and [9].

4 Results and discussions

Section 4.1 elaborates on the predictive ability of the prediction methods. Section 4.2 describes the effectiveness and efficiency of the combined PSO and the selected roughness model for optimizing productivity and still obtaining the desired roughness.

4.1 On prediction of surface roughness

Each selected parameter and roughness output show drastic differences in magnitude. Hence, preprocessing was done to avoid numerical interference for both SVR-based and NN-based roughness models. The inputs and targets were normalized so that they fell in the interval [−1,1]. The discussed SVR method and the PSO algorithm were implemented in MATLAB 7 running on a Pentium IV 2.4 GHz with Microsoft Windows XP operating system. The quadratic programming function, quadprog with MATLAB7, was invoked to solve Eq. (3) for the SVR method. A neural network toolbox with MATLAB 7 was used for the prediction of roughness by using the feed-forward back-propagation network.

The performance of SVR is related to the number of dependent and independent variables as well as the combination of parameters used. One of the major advantages of SVM is its fewer parameter settings. There are only two primary parameters involved, the capacity C and the kernel function used. Unavoidably, there is no systematic way yet to determine these settings. Fortunately, SVM's efficient algorithm makes model selection easier as compared to other methods. In this study, Dataset 1 was chosen for model selection since it is quite large for both training and test sets. Subsequently, the parameters obtained would be used for the other two datasets, Datasets 2 and 3, as well. Due to a large number of data in Dataset 1, the holdout method was chosen as a validation technique for model selection and performance estimation of the constructed SVR model. The values of C's was varied from 0.1 to 100,000 with various kernel functions such as linear, polynomial, and Gaussian radial basis functions (GRBF). The degree of polynomial and the width of the GRBF were set at 2. The graphical results are depicted in Fig. 2. It shows that the GRBF kernel function performs better than the polynomial kernel and the default linear functions. The C's ≥ 10 give high accuracy of nearly 80% for the unseen test set and peak at 10,000 with accuracy of 81.37%. Therefore, this combination of GRBF kernel and C=10,000 was chosen since it provides the best performance for SVR.

The holdout method was again used with Dataset 1 as a validation technique for NN-based roughness model selection. Its architecture was chosen based on generalization performance indicator by using trial and error approach for 10 runs for each varied structure from 2 to 8 hidden nodes. The

Table 1 Summary of modeled data sets

	Dataset 1 (from [11])	Dataset 2 (from [3])	Dataset 3 (from [9])
No. of training data	125	60	48
No. of test data	18	24	24
Spindle speed (rpm)	600, 800, 1000, 1200, 1400	750, 1000, 1250, 1500	750, 1000, 1250, 1500
Feed rate (inch per minute or ipm)	3.94, 4.43, 4.91, 5.41, 5.91	6, 9, 12, 15, 18, 21, 24	6, 12, 18, 24
Feed rate (centimeter per minute or cm/m) *1 inch=2.54 cm	10.0076, 11.2522, 12.4714, 13.7414, 15.0114	15.24, 22.86, 30.48, 38.1, 45.72, 53.34, 60.96	15.24, 30.48, 45.72, 60.96
Depth of cut (inch)	0.063, 0.109, 0.156, 0.2031, 0.25	0.01, 0.03, 0.05	0.01, 0.03, 0.05
Depth of cut (cm) *1 inch=2.54 cm	0.16002, 0.27686, 0.39624, 0.515874, 0.635	0.0254, 0.0762, 0.127	0.0254, 0.0762, 0.127

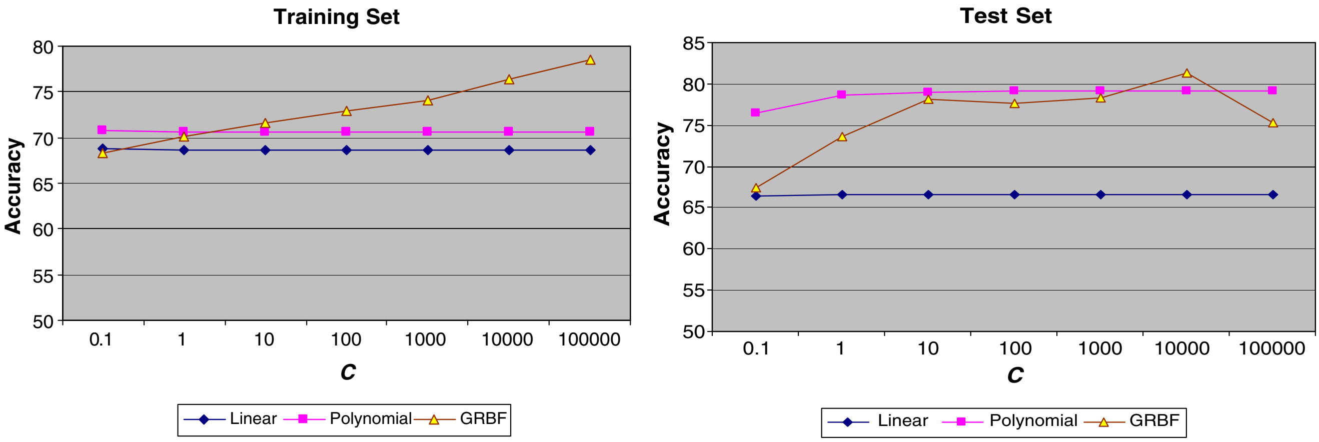

Fig. 2 Dependence of accuracy level (%) on *C* using different kernel functions

final architecture was 3–5–1 which represented number of input terminals-number of hidden nodes-number of output node, respectively. The activation functions selected were the hyperbolic tangent sigmoid transfer function or “tansig” for hidden nodes. It is commonly used in backpropagation network because it is differentiable and covers the bipolar continuous range (-1,1). The linear transfer function or “purelin” was selected for the output node in the last layer since the network outputs could take on any value. The number of epochs was set at 600. The same topology was also applied to the other datasets as well. Results shown in Table 2 are the best one from 10 replications of every dataset.

The time to fit an SVR roughness approximation was on the order of a few seconds and prediction with a fitted SVR model took less than a second for both types of kernel attempted. These computational times obviously depended on the number of data available. The fitting and prediction times of NN and PSO-based roughness models were comparable to each other and slightly faster than those of the SVR model. This was due to the simple mathematical operators employed by the PSO and the speed-up optimizer, Levenberg–Marquardt backpropagation, utilized by the NN toolbox. Note that in Table 2 the measures used in the first training and test columns are average percentage error [3] and those in the last two columns are percentage total error [11]. They are given by

$$\text{average percentage error} = \frac{\sum_{i=1}^{n} \frac{\left|Ra_i - \widehat{Ra}_i\right| \times 100\%}{Ra_i}}{n} \tag{9}$$

$$\text{and percentage total error} = \frac{\sum_{i=1}^{n} \left|Ra_i - \widehat{Ra}_i\right|}{\sum_{i=1}^{n} Ra_i} \times 100\%. \tag{10}$$

The resultants of predicted surface roughness by using SVR model were quite close to actual measured surface roughness for both training and test sets as illustrated in Table 2 and Figs. 3 to 5. It could generalize well for unseen test sets. The comparative accuracy results made against previously published reference values and those of other methods are also depicted in Table 2. Clearly, the outcomes obtained for Datasets 2 and 3 were in close agreement to those of other methods with accuracy over 90%. These were benefited by the SVR’s structural risk minimization which theoretically attempted to balance between training error and confidence interval. In other words, the SVR model could learn pattern of complex relationships between machining conditions and surface roughness and did not adjust to very specific random features that have no causal relation to the target function. As a result, the SVR-based prediction model would be able to avoid overfitting problem and good generalization performance on unseen data would be obtained. This means that the SVR can

Table 2 Comparative accuracy results

Dataset 1	Training [a]	Test [a]	Training [b]	Test [b]
SVR+GRBF Kernel	68.73%	83.27%	76.42%	81.37%
SVR+Poly Kernel	60.48%	82.70%	70.62%	79.06%
PSO+a closed-form prediction model	65.95%	76.70%	71% ([11])	79% ([11])
NN	70.65%	74.72%	73.46%	78.94%
Dataset 2	Training[a]	Test[a]	Training[b]	Test[b]
SVR+GRBF Kernel	93.92%	90.48%		
SVR+Poly Kernel	90.46%	88.35%		
[3]	90.29%	90.03%		
NN	91.33%	89.58%		
Dataset 3	Training[a]	Test[a]	Training[b]	Test[b]
SVR+GRBF Kernel	95.86%	95.45%		
SVR+Poly Kernel	90.00%	93.91%		
[9]	N/A	95.35%		
NN	91.19%	91.54%		

[a] average percentage error,
[b] percentage total error

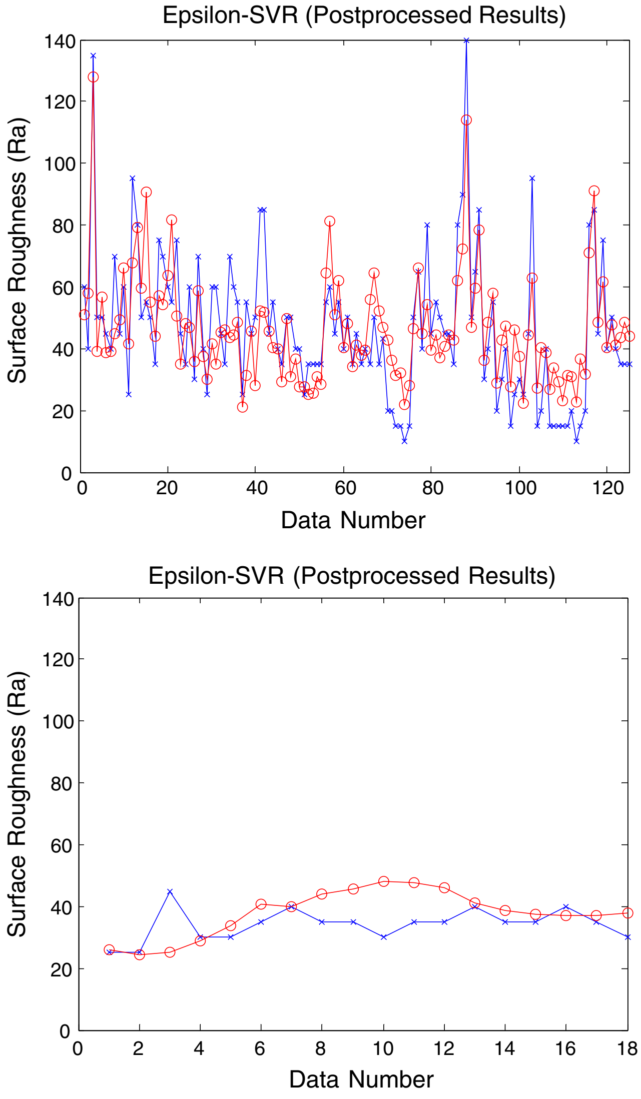

Fig. 3 Comparison of actual *Ra* (x) and predicted *Ra* (o) in µ inch on training (top) and test (bottom) sets of Dataset 1

accurately predict such surface finish, especially with GRBF kernel.

The accuracy results of every predictive model for Dataset 1 were in the range of 70% due partly to noise in the data [11]. This noise should be removed from the data set or more experimental data should be used in the training to allow for a reliable and accurate prediction model. It is important to note that data used for any method must be properly collected. Otherwise, the predictive model developed will not be as reliable as desired. This shows drastic difference as opposed to the results from Datasets 2 and 3.

Normally, quite a number of cutting parameters impact the manufactured part's surface texture, including all of them in the prediction model, require numerous experiments and extensive time for data collection. That is why only major factors are taken into account in literature. This implies that the identification process of major factors is very critical as well. The chosen major factors significantly affect the reliability of the prediction model. In this work, the noted difference between those considered datasets was the use of radial depth of cut in Dataset 1 and axial depth of cut in Datasets 2 and 3. This might in part contribute to varying degrees of success of the prediction model. To obtain more comprehensive and realistic prediction model, both types of depth of cut should then be taken into consideration.

The NN-based prediction model also shows good results, especially for the training sets. This is NN's main strength as a universal approximator. However, to obtain good performance on unseen data, a validation technique like the holdout method was used with a series of experiments on

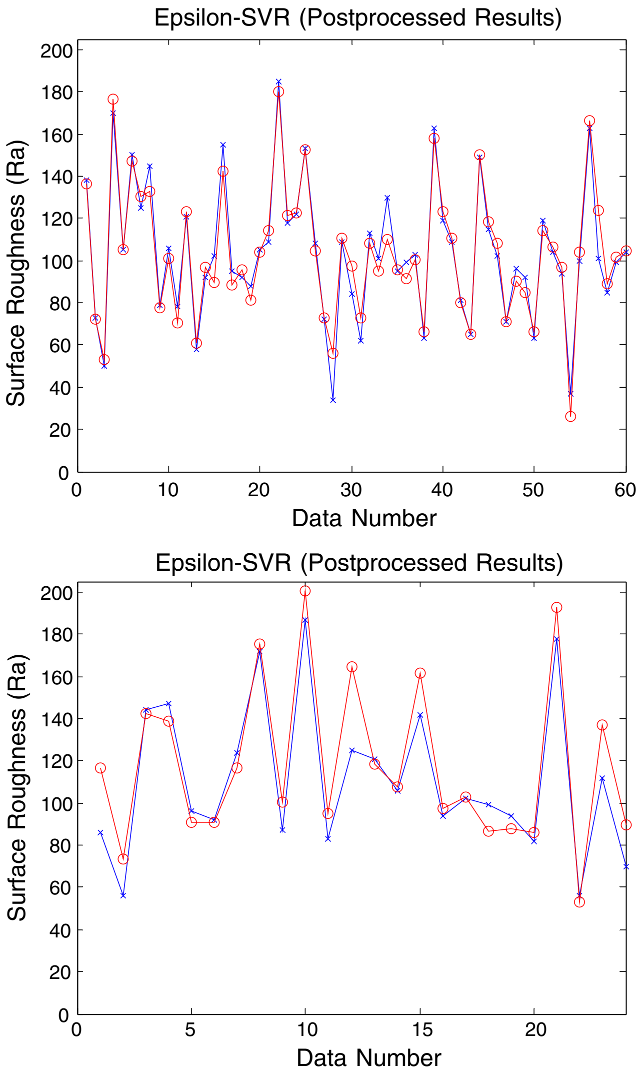

Fig. 4 Comparison of actual *Ra* (x) and predicted *Ra* (o) in µ inch on training (top) and test (bottom) sets of Dataset 2

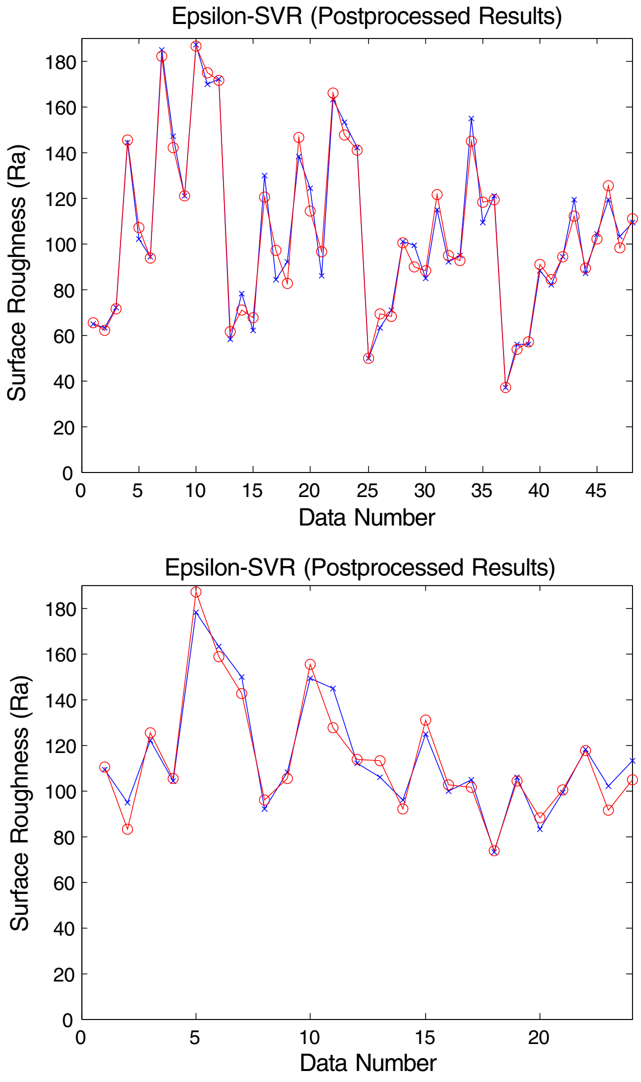

Fig. 5 Comparison of actual *Ra* (x) and predicted *Ra* (o) in μ inch on training (top) and test (bottom) sets of Dataset 3

number of hidden nodes. Thus, a longer development time for determination of an acceptable prediction model is normally resulted.

The main characteristic of the PSO-based model is the use of the available closed form model representing relationship between some specific cutting conditions and surface roughness. Its strength is the ease of use and simple in concept by applying the optimization technique with a proper formulation. Its main drawback, however, can easily be overlooked; that is, the availability of the closed-form relationship. Such a closed form model is quite difficult to be extended to cover more machining conditions for more complex nonlinearities and interactions model. In other words, a similar model covering relationships of more machining conditions is very expensive to develop in terms of difficulty, time, and accuracy.

Generally, the accuracy of test results is slightly lower than that of the training results. However, this was not the case for every technique on Dataset 1. The following factors can cause such results: poor quality of data and the relatively small size of the test set compared to the size of the training set. The quality of data is very critical for modeling. The proposed method in [11], which was repeated here, also exhibited the smaller error on the test set than that on the training set. As already pointed out in [11], when noise was filtered out, the training accuracy was increased to 82% and testing increased to 80.76%. Hence, a data collection process played an important role in modeling. Moreover, even though Dataset 1 contains a rather large training set (125 data points or 87.41%) but its test set of 18 data points is very small, only 12.59%. This can produce a relatively high predictive accuracy on the test set due to the chance effect of small sample size. If the differences between the predicted *Ra* and the actual *Ra* are not too much, the small sample size will likely attenuate these differences no matter what measure, Eqs. (9) or (10), is used. The numerical measures are a bit deceptive here with very small sample size. Figure 3 does not illustrate a well-captured pattern by the prediction model. Similar graphical results were also obtained in [11]. Even though these techniques were different, they processed on the same dataset. They heavily relied on characteristics of the data such as quality and sample size.

If a huge database storing a wide range and high precision of each cutting parameter and their corresponding roughness values can be had, the roughness prediction model is not necessarily needed. A search in this database to obtain a proper set of cutting conditions for a desired roughness can simply be done. To maximize productivity, an optimization procedure with qualified search results of a desired roughness can also be accomplished rather easily. However, this scenario is never the case in practice because of the very large cost involved for time, labor, tools, material, and energy. The affordably collected data are normally in tens or hundreds. This warrants the investigation of the dependably prediction roughness model to

Table 3 Results of the optimization model using Dataset 1

d (μ inch)	*Ra* (μ inch)	*F* (ipm)	*S* (rpm)	*D* (inch)	Convergence
30	26.94881	5.91	998.9379	0.101337	18 iterations
30	26.31474	5.91	1043.7017	0.097798	24 iterations
30	29.72597	5.91	1080.1118	0.111489	20 iterations
30	27.33841	5.91	1003.3268	0.078254	32 iterations
30	27.25179	5.91	1084.5201	0.101141	33 iterations

Table 4 Results of the optimization model using Dataset 2

d (μ inch)	Ra (μ inch)	F (ipm)	S (rpm)	D (inch)	Convergence
90	89.99992	17.59797	1325.257	0.039283	161 iterations
90	90	17.59809	1325.7	0.03926	171 iterations
90	89.99995	17.59782	1326.119	0.039349	215 iterations
90	90	17.59333	1325.834	0.038804	220 iterations
90	90	17.59808	1325.778	0.039239	123 iterations

regress for nonexisting data. Therefore, the SVR-based method with GRBF kernel was selected since it appeared to be the most reliable and could effectively and efficiently find a complex roughness model leading to a very good generalization result. It can be easily expanded to cover more cutting parameters reflecting more comprehensive relationships.

4.2 On optimization of machining parameters

The benefit of the prediction model would not be fully utilized if it is merely used to evaluate cutting parameters. More practical application would be the determination of proper cutting condition while satisfying required roughness. This led to the optimization of machining parameters by including the predictive roughness model (SVR with GRBF kernel) obtained as a constraint. The PSO was then used to find those optimum parameters. Since the predictive roughness model used was very reliable, the optimum cutting parameters obtained by a very good optimizer like the PSO should be very trustworthy as well. The constrained PSO was implemented in the same software and hardware set as the SVR roughness model. Five experiments for each data set were conducted and their results are tabulated in Tables 3, 4, 5.

The outcomes obtained show excellent performance by the PSO in terms of quality of solution and consistency, especially for Datasets 2 and 3. Even though Datasets 2 and 3 were not identical, they were originated by the same group of researchers [3, 8] and could be captured with a similar pattern of cutting conditions by the SVR-based roughness model as evident in Tables 4 and 5. Rather quickly, the PSO found similar near-optimal results for the feed rate by almost binding the found roughness, *Ra*, to the limit of the given roughness, *d*. The outcomes for every parameter were very consistent in every trial of both data sets. This implies that the proposed approach, the prediction by the SVR with well-controlled data collection process and the optimization by the PSO with practical formulation, can be combined to find the optimum cutting parameters when the specification of roughness is given. Even though the results obtained from Dataset 1 might not be as good as those from the other two datasets, similar outcomes could also be obtained. The PSO found the optimal result, which was the upper limit of feed rate, *F*, used in the training of the SVR-based roughness model and could converge very quickly with the given roughness value, *d*. Other parameters obtained do not show high consistency across experiments. They depicted a bit of variations for every parameter but feed rate. This was a consequence of noisy pattern contained in this dataset, which led to less reliable prediction performance of every method attempted as illustrated in Table 2. This in turn dampens the results of the optimization model. However, the PSO could still find cutting parameters for every experiment that were not discernibly far from one another. This in fact verified that the PSO could still perform well in terms of quality of solution and consistency.

Besides the advantages mentioned earlier, the original PSO was also built for speed since only primitive mathematical operators were computed [21]. To handle constraints, some simple and effective mechanisms were additionally included [23]. Consequently, its computational time increased. Moreover, the chosen roughness, *d*, played an essential role for the time taken as well. Nevertheless, the constrained PSO could converge somewhat quickly for every dataset attempted and took about 4.5 to 5.5 seconds

Table 5 Results of the optimization model using Dataset 3

d (μ inch)	Ra (μ inch)	F (ipm)	S (rpm)	D (inch)	Convergence
90	89.99192	20.13713	1464.296	0.033641	260 iterations
90	90	20.19891	1472.983	0.033638	249 iterations
90	89.99999	20.10071	1457.551	0.033944	160 iterations
90	89.99999	20.07185	1456.013	0.034588	225 iterations
90	89.99969	20.17385	1469.362	0.033952	171 iterations

with the above parameter settings and constraints to reach the preset maximum iteration of 350.

Based on simplicity of its concept, primitive mathematical operators, consistent and near-optimal results, and short computer codes, the PSO was shown to be a very effective and very efficient optimizer.

5 Conclusions

This work demonstrated that the combined SVM and PSO could be applied to effectively and efficiently predict surface roughness and determine optimal cutting condition for the roughness specification, thus responding to the needs of users on the subjects of surface texture and productivity. The SVM showed excellent performance in predicting surface roughness by considering common cutting conditions of spindle speed, feed rate, and depth of cut. High accuracy for both training and test sets were obtained, which also experimentally verified good generalization in theory of the SVM for unseen data. In addition, no prior assumptions were made on a statistical model for the data used. Comparative results with other works also validated the potential of the proposed method. Even though the SVM may be quite complex in concept, its implementation and use were quite simple and fast. Moreover, its parameters could be set rather easily. The PSO was then combined with the predictive roughness model obtained to provide practical solutions for the user by giving an optimal set of machining parameters for required surface roughness and minimum machining time. The PSO was shown to be an effective and efficient algorithm by robustly finding near-optimal and consistent results with short computer code and simple mathematical operator while converging rather quickly depending on the specified roughness. It is also suitable for use with SVR-based model where no explicit relation between inputs and outputs is available. Therefore, the combination of both techniques, SVM and PSO, looks very promising for process modeling and its optimal parameters selection.

More input factors such as material types of workpiece and cutting tools, tool geometry and wear can additionally and simply be taken into consideration to comprehensively capture their relationships with roughness and consequently obtain greater predictive accuracy. Just like the black-box nature of ANNs, the SVR-based model still lacks the clear interpretability in expressing and explaining relationships between input (cutting conditions) and output (roughness). This issue should also be dealt with in the future. Systematic parameters selection of the PSO will certainly enhance the ease of use of the presented algorithm.

Acknowledgment The first author was partially supported by the Thailand Research Fund (TRF) grant MRG4980170.

References

1. Kalpakjian S, Schmid SR (2001) Manufacturing engineering and technology. 4th edn. Prentice-Hall, Upper Saddle River
2. Kline WA, Devor RE, Shareef IA (1982) The prediction of surface accuracy in end milling. J Eng Ind 104:272–278
3. Lou MS, Chen JC, Li CM (1998) Surface roughness prediction technique for CNC End-Milling. J Ind Technol 15 (1):1–6
4. Mansour A, Abdalla H (2002) Surface roughness model for end milling: A semi-free cutting carbon casehardening steel (EN32) in dry condition. J Mater Process Technol 124:183–191 DOI 10.1016/S0924–0136(02)00135–8
5. Arbizu IP, Perez CJL (2003) Surface roughness prediction by factorial design of experiments in turning processes. J Mater Process Technol 143-144:390–396 DOI 10.1016/S0924–0136(03) 00407–2
6. Feng CX, Wang X (2002) Development of empirical models of surface roughness prediction in finish turning. Int J Adv Manuf Technol 20:348–356 DOI 10.1007/s001700200162
7. Lou SJ, Chen JC (1997) In-process surface recognition of a cnc milling machine using the fuzzy nets method. Comp Ind Eng 33(1–2):401–404 DOI 10.1016/S0360–8352(97) 00122–8
8. Lou SJ, Chen JC (1999) In-process surface roughness recognition (isrr) system in end-milling operations. Int J of Adv Manuf Technol 15:200–209 DOI 10.1007/s001700050057
9. Lo SP (2003) An adaptive-network based fuzzy inference system for prediction of workpiece surface roughness in end milling. J Mater Process Technol 142:665–675 DOI 10.1016/S0924–0136 (03)00687–3
10. Ozel T, Karpat Y (2005) Predictive modeling of surface roughness and tool wear in hard turning using regression and neural networks. Int J Mach Tools Manuf 45:467–479 DOI 10.1016/j. ijmachtools.2004.09.007
11. El-Mounayri H, Dugla Z, Deng H (2003) Prediction of surface roughness in end milling using swarm intelligence. IEEE Swarm Intelligence Symposium, 220–227
12. Smola AJ, Scholkopf B (1998) A Tutorial on Support Vector Regression, NeuroCOLT Technical Report NC-TR-98–030. Royal Holloway College, University of London
13. Lawtrakul L, Prakasvudhisarn C (2005) Correlation studies of HEPT derivatives using swarm intelligence and support vector machines. Monatsh Chem/Chem Month 136:1681–1691
14. Prakasvudhisarn C, Trafalis TB, Raman S (2003) Support vector regression for determination of minimum zone. ASME J Manuf Sci Eng 125:736–739 DOI 10.1115/1.1596572
15. Joachims T (1999) Making large-scale support vector machine learning practical. In: Scholkopf B, Burges CJC, Smola A (eds) Advances in kernel methods: support vector learning. MIT Press, Cambridge, pp 169–184
16. Platt JC (1999) Fast training of support vector machines using sequential minimal optimization. In: Scholkopf B, Burges CJC, Smola A (eds) Advances in kernel methods: support vector learning. MIT Press, Cambridge, pp 185–208
17. Tandon V, El-Mounayri H, Kishawy H (2002) NC end milling optimization using evolutionary computation. Int J Mach Tools Manuf 42:595–605 DOI 10.1016/S0890–6955(01)00151–1

18. Vapnik V, Golowich SE, Smola A (1997) Support vector method for function approximation, regression estimation, and signa lprocessing. In: Mozer M, Jordan M, Petsche T (eds) Advances in neural information processing systems 9. MIT Press, Cambridge, pp 281–287
19. Drucker H, Burges CJC, Kaufman L, Smola A, Vapnik V (1997) Support vector regression machines. In: Mozer M, Jordan M, Petsche T (eds) Advances in neural information processing systems 9. MIT Press, Cambridge, pp 155–161
20. Vapnik V (1995) The Nature of Statistical Learning Theory. Springer, New York
21. Kennedy J, Eberhart RC (1995) Particle Swarm Optimization. Proc of the IEEE Int Conf on Neural Networks (Perth, Australia), IEEE Service Center, Piscataway, NJ, 4, 1942–1948
22. Eberhart RC, Shi Y (2001) Particle Swarm Optimization: Developments, Applications and Resources. Proc of the 2001 Congress on Evolutionary Computation, IEEE Service Center, Piscataway, NJ, 81–86
23. Hu X, Eberhart RC, Shi Y (2003) Engineering optimization with particle swarm. Proc of the IEEE Swarm Intelligence Symposium 2003 (SIS 2003), Indianapolis, Indiana, USA. 53–57