Sādhanā Vol. 39, Part 3, June 2014, pp. 533–548. © Indian Academy of Sciences

Particle swarm optimization of a neural network model in a machining process

SAURABH GARG[1], KARALI PATRA[2,*] and SURJYA K PAL[3]

[1]Department of Mechanical Engineering, University of California, Berkeley 94720, California, USA
[2]Department of Mechanical Engineering, Indian Institute of Technology Patna, Patna 800 013, India
[3]Department of Mechanical Engineering, Indian Institute of Technology Kharagpur, Kharagpur 721 302, India
e-mail: kpatra@iitp.ac.in

MS received 7 July 2012; revised 31 July 2013; accepted 29 January 2014

Abstract. This paper presents a particle swarm optimization (PSO) technique to train an artificial neural network (ANN) for prediction of flank wear in drilling, and compares the network performance with that of the back propagation neural network (BPNN). This analysis is carried out following a series of experiments employing high speed steel (HSS) drills for drilling on mild steel workpieces, under different sets of cutting conditions and noting the root mean square (RMS) value of spindle motor current as well as the average flank wear in each case. The results show that the PSO trained ANN not only gives better prediction results and reduced computational times compared to the BPNN, it is also a more robust model, being free of getting trapped in local optimum solutions unlike the latter. Besides, it offers the advantages of a straight-forward logic, simple realization and underlying intelligence.

Keywords. Particle swarm optimization; artificial neural network; back propagation algorithm; flank wear; drilling.

1. Introduction

Tool Condition Monitoring (TCM) is an important area of research for manufacturing industries with pressing demands for large scale unmanned or automated production. Consequently, there is an ever-increasing need to develop efficient techniques for tool-life estimation and tool-breakage detection. Since tool life is critically affected by the tool wear, accurate prediction of this wear becomes very important.

Drilling is one such machining process of extreme importance to the industry. Apart from the different types of wear occurring during the drilling operation, the drill failure itself

*For correspondence

assumes a stochastic nature under constant cutting conditions (Jantunen 2002), thereby adding considerable inherent complexity. Some such associated reasons for varying drill life include non-homogeneities in the workpiece and drill materials, the irregularities in the cutting fluid motion and the unavoidable asymmetry introduced during the grinding of the cutting edges. Drill wear is very complex process which is a result of the combined actions of some physical and thermo-chemical phenomena like abrasion, adhesion, diffusion and fatigue. It is highly a complicated process associated with many factors such as contact stress, nature and composition of work piece and cutting tool, temperature on the cutting tool and cutting conditions. All these factors contribute significant uncertainty in the drill wear estimation. Hence a deterministic approach can not estimate the drill wear accurately. It has been earlier shown that an artificial neural network (ANN) model predicts drill wear more accurately than that by a deterministic method (Patra *et al* 2007).

The use of artificial neural network (ANN) for modelling wear phenomenon in drilling has generated a lot of interest for quite some time now, because they have proved to be excellent function approximators, mapping any function to an arbitrary degree of accuracy, coupled with their ability for generalization, self-organization and self-learning (Fausett 1994).

Amongst the neural networks, the most common and the most popular is the MLP neural network, implemented with the standardized back propagation (BP) algorithm or one of its variants, and popularly known as the BPNN. The BP algorithm, employing a gradient descent search technique is seriously prone to getting trapped in local optimum solutions, if the initial set of weights is not selected properly. In other cases, the calculations may even overflow or fluctuate between the optima. These limitations of the BPNN have prompted researchers to look beyond the existing model and search for more powerful optimizations techniques that can help reach the optimal solution in an improved fashion.

An effective find for such robust techniques has been the application of evolutionary algorithm (EA) for neural network optimization. One such evolutionary computation technique, particle swarm optimization (PSO), the subject of this study, was proposed by Eberhart & Kennedy (1995), and originates from the behavioural simulation of bird flocking and fish schooling. The original intent was to graphically simulate the graceful but unpredictable choreography of a bird flock. At some point in the evolution of the algorithm, it was realized that the conceptual model was, in fact, an optimizer following which a number of parameters extraneous to optimization were eliminated leading to the basic PSO algorithm.

The PSO technique is being recently used in a number of optimization problems dealing with discrete and continuous functions, model identification, etc. It is also generating some good amount of interest because of its potential application in the field of machining. The following literature review presents some of the works that have employed this technique in various fields.

Zhou *et al* (2006) used the PSO based neural network optimization for prediction of diameter errors in a boring machine. In their work, they established an improvement in the quality of optimization of the neural networks and error compensations with the use of the PSO algorithm, which achieves a better machining precision with fewer numbers of iterations. Elbeltagi *et al* (2005) compared the performance of the PSO technique with other EAs for both continuous and discrete optimization problems in terms of processing time, convergence speed, and quality of the results. A powerful evolutional PSO learning algorithm that self-generates radial basis function neural network (RBFN) to deal with three non-linear problems was used by Feng (2006). It was found that the proposed PSO allows a high accuracy within a short training time when determining RBFN with small number of radial basis functions. A multi-objective optimization of hard turning using neural network modelling and swarm intelligence

was presented by Karpat & Ozel (2005). They used the ANN to model the surface roughness and tool wear characteristics of hard turning. A PSO algorithm was developed by them to find the optimum process parameters which satisfy given limit, tool wear and surface roughness values and maximize the productivity at the same time. Zhang & Shao (2000) described a new evolutionary algorithm for evolving the ANN which was based on the PSO technique. Both the architecture and the weights of the ANN were adaptively adjusted according to the quality of the neural network until the best architecture or a terminating criterion was reached. The performance of the basic PSO algorithm with the constriction PSO (with and without mutation) on some test functions of different dimensions was compared by Stacey *et al* (2003). They found that the use of constriction PSO with mutation provided significant improvement in certain cases. Zhao *et al* (2005) presented an improved PSO algorithm for neural network training employing a population diversity method to avoid premature convergence. Asokan *et al* (2005) used the PSO technique to optimize the grinding process parameters such as wheel and workpiece speed, depth and lead of dressing, etc. subjected to suitable constraints with the objective of minimizing the production cost and obtaining the finest possible surface finish. They also compared their results with that of genetic algorithms and quadratic programming techniques. The PSO algorithm for the optimal machining tolerance allocation of over-running clutch assembly to obtain the global optimum solution with the objective to obtain minimum cost of manufacturing was used by Haq *et al* (2006). Gaitonde & Karnik (2012) applied artificial neural network – particle swarm optimization approach for selection of optimum process parameters for minimizing burr size in drilling process. The PSO algorithm was applied for optimization of multi objective problem in tile manufacturing process (Navalertporn & Afzulpurkar 2011) and also for machinery fault detection (Samanta & Nataraj 2009). Malviya & Pratihar (2011) used PSO to tune the Radial Basis Function (RBF) networks for modelling of MIG welding Process.

In this paper, a comparison of the performance of the PSO optimized neural network with the standard BPNN is presented for the prediction of flank wear in drilling operation using motor current signals. This is done not only to give a measure of the improved prediction capabilities of this relatively newer computational intelligence technique over and above the conventional existing ones, but also to highlight the underlying intelligence in flock migration, involving both individual experience and social collaboration, the combination of which can be labelled as swarm intelligence.

2. Back propagation neural network (BPNN)

Figure 1 shows the BPNN model which utilizes the generalized delta rule algorithm for the network training. The most general structure of the network consists of an input layer, a variable number of hidden layers each containing any number of nodes and an output layer. The input layer receives the information from an external source, which is subsequently modified by the interconnection weights between it and the adjacent hidden layer. The sum of the modified signals (total activation) is then operated upon by a transfer function (with sigmoid function being used in this study), and these activated values in turn become the starting signals for the next adjacent layer. In this way, the modified signal finally reaches the output layer where it is communicated to the external receptor(s).

Let I_i^p be the ith input data from the pth training pattern of the data set to the input layer. L is the total number of input parameters applied to the network. In the forward path, outputs from

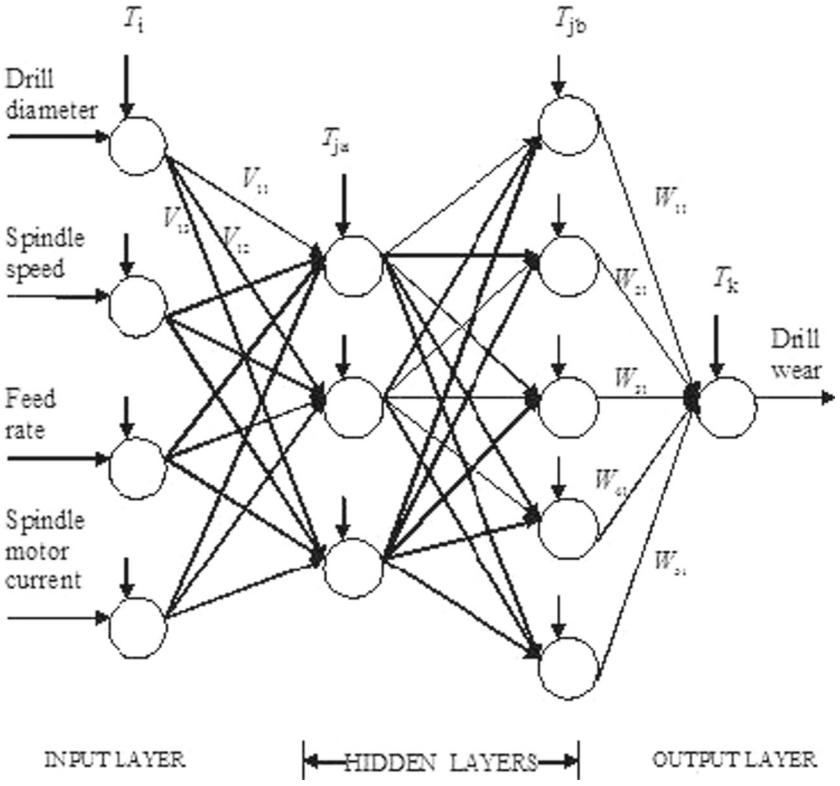

Figure 1. Schematic diagram of BPNN model.

the *i*th input node, *j*th neuron in the hidden layer, and *k*th neuron of the output layer are expressed by Eqs. 1, 2 and 3, respectively:

$$O_i^p = \frac{1}{1 + \exp\left\{-(I_i^p + T_i)\right\}}, \tag{1}$$

$$O_{hj}^p = \frac{1}{1 + \exp\left\{-\left(\sum_{i=1}^{L} V_{ij}^p O_i^p + T_j\right)\right\}}, \tag{2}$$

$$O_{ok}^p = \frac{1}{1 + \exp\left\{-\left(\sum_{j=1}^{M} W_{jk}^p O_{hj}^p + T_k\right)\right\}}, \tag{3}$$

where, V_{ij} is the weight between the *i*th neuron of the input layer and the *jth* neuron of the hidden layer, W_{jk} the weight between *jth* neuron of the hidden layer and *k*th neuron of the output layer, T_i the bias at *i*th neuron of the input layer, T_j the bias at *jth* neuron of the hidden layer, T_k the bias at *k*th neuron of the output layer, and M, N are the number of nodes in the hidden and output layers, respectively.

The interconnection weights V_{ij} and W_{jk} are adjusted using backpropagation algorithm based on gradient descent method for error minimization (Fausett 1994). The incremental weights are given by Eqs. (4) and (5):

$$\Delta W_{jk}^{p} = \eta \delta_{k}^{p} O_{hj}^{p} + \alpha \Delta W_{jk}^{p-1}, \tag{4}$$

$$\Delta V_{ij}^{p} = \eta \delta_{j}^{p} O_{i}^{p} + \alpha \Delta V_{ij}^{p-1}, \tag{5}$$

where

$$\delta_{k}^{p} = O_{ok}^{p}(1 - O_{ok}^{p})(T_{ok}^{p} - O_{ok}^{p}), \tag{6}$$

$$\delta_{j}^{p} = O_{hj}^{p}(1 - O_{hj}^{p}) \sum_{k=1}^{N} \delta_{k}^{p} W_{jk}^{p}. \tag{7}$$

In Eqs. (6) and (7), and are the gradient descent terms of output layer and hidden layer, respectively for the pth training pattern, η is the learning rate and α the momentum coefficient.

3. Particle Swarm Optimization (PSO) algorithm

Particle swarm optimization (PSO) is a population based (evolutionary) stochastic optimization technique in which a collection of particles move around in search of space looking for the best solution to an optimization problem. The concept is derived from the motion of a flock of birds that communicates and learns from each other in search for food.

All birds have their own velocity that drives the direction they move in. Each bird looks in a particular direction and while communicating with others, they identify the bird that is in the best location. Accordingly, each bird speeds towards the best bird using a velocity that depends on its current position. Each bird, then, investigates the search space from its new local position, and the process continues until the flock reaches a desired destination.

Thus, we see that the velocity of each particle in a PSO algorithm is affected both by the position of the particle with the best fitness as well as each particle's own best fitness. Fitness refers to how well a particle performs: In a flock of birds this might be how close a bird is to a food source, in an optimization algorithm this refers to the proximity of the particle to optima. It is important to note that the process involves both social interaction and intelligence so that birds learn from their own experience (local search) and also from the experience of others around them (global search). Figure 2 depicts a flowchart for implementation of the PSO algorithm.

Given S as the number of optimization variables, the PSO algorithm starts by randomly initializing a collection (swarm) of N particles in the S-dimensional search space. At any stage of the optimization algorithm, each ith particle keeps track of three parameters (Elbeltagi *et al* 2005):

(1) Current position: $X_i = (x_{i1}, x_{i2}, \ldots\ldots\ldots\ldots, x_{is})$.
(2) Best previous position: $P_i = (p_{i1}, p_{i2}, \ldots\ldots\ldots\ldots, p_{is})$.
(3) Flying velocity: $V_i = (v_{i1}, v_{i2}, \ldots\ldots\ldots\ldots, v_{is})$.

In each cycle, the swarm also keeps track of the best particle position Pg calculated as the best fitness of all the particles. Thus, using each of these three parameters X_i, P_i and P_g, each particle updates its velocity V_i to catch up with the best particle g, and uses this new velocity in turn to update its position X_i as represented through Eqs. (8) and (9), respectively.

$$V_{inew} = \omega V_i + c_1 (P_i - X_i)\, rand() + c_2 \left(P_g - X_i\right) rand(), \tag{8}$$

$$X_{inew} = X_i + V_{inew}, \tag{9}$$

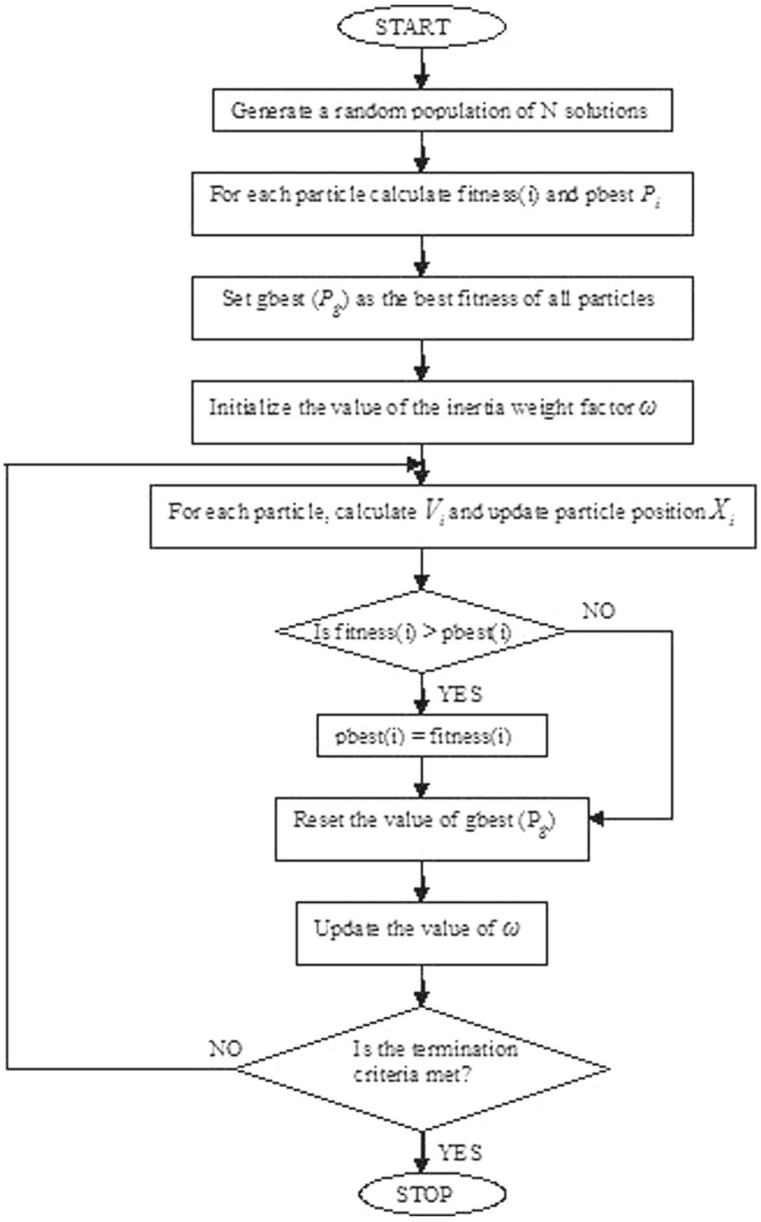

Figure 2. Flowchart for PSO algorithm.

where c_1 and c_2 are two positive constants called acceleration constants while *rand*() is a random functions in the range [0,1]. An inertia weight ω is employed as an improvement proposed by Shi & Eberhart (1998) to control the impact of the previous history of velocities on the current velocity. The operator ω plays the role of balancing the global search and the local search; and was proposed to decrease linearly with time.

The velocity V_i is allowed to vary within a particular range specified by an upper limit $V_{\max}$ (Kennedy & Eberhart 1995) as given by Eq. (10).

$$V_{\max} \geq V_i \geq -V_{\max}. \tag{10}$$

In a modification to the standard PSO algorithm, with intent to ensure the convergence of the search, Clerc (1999) introduced a constriction factor k, with Eq. (8) being converted to:

$$V_{inew} = k\left[V_i + c_1\left(P_i - X_i\right) rand() + c_2\left(P_g - X_i\right) rand()\right], \tag{11}$$

$$k = \frac{2}{\left|2 - \phi - \sqrt{\phi^2 - 4\phi}\right|}, \tag{12}$$

where $\phi = c_1 + c_2$, $\phi > 4$.

From Eq. (12), k is seen to be constricted by c_1 and c_2. The constriction factor was assumed by Clerc (1999) to eliminate the need for the maximum search velocity by mathematically ensuring that a convergence was achieved. However, from subsequent experiments and applications by Shi & Eberhart (2000), it has been concluded that a better approach to use as a 'rule of thumb' is to limit $V_{\max}$ to $X_{\max}$, the dynamic range of each variable on each dimension.

In this work, a combination of the three approaches (using a maximum search velocity, an inertia weight factor and a constriction factor) has been adopted as it was found by a series of trial runs that the effect of none of these three parameters in improving the search solution could be completely eliminated in the presence of other two. Thus, the particle velocity change is finally expressed by Eq. (11); with X_i, $V_{\max}$ and k given by Eqs. (9), (10), and (12), respectively.

4. Experimental set-up

4.1 *Data acquisition apparatus*

Figure 3 shows a schematic representation of the experimental set-up used for performing the drilling operation as well as for online data acquisition during the process. A column type drilling machine (HMT make) was used for this purpose. As shown in the figure, a three phase line connects the power source to the rotating spindle motor.

The online acquisition of current signals entering the spindle motor is achieved through Hall effect AC current sensors (Tektronix make, model A621) which are fitted to all the phases of the three phase line. The output from these sensors is in analog form and needs to be converted into corresponding digital signals before storage into the acquisition system (IBM PC). This is achieved by means of an A/D board (PCI-DAS 4020/12) attached to the PC. The sampling rate i.e., the frequency at which the current signals are collected and stored, was kept at 1000 Hz. The spindle motor current is expressed in units of millivolt (mV) with an equivalent conversion of 100 mV=1 A current.

4.2 *Cutting conditions*

Three different diameter twist drills 8, 9 and 10 mm are used in the drilling operations for drilling holes on mild steel workpieces. The drills are made of high speed steel (HSS). They are used in conjunction with a wide range of cutting conditions to obtain an exhaustive set of experimental data for which the current signals and flank wear have been measured. Five different values

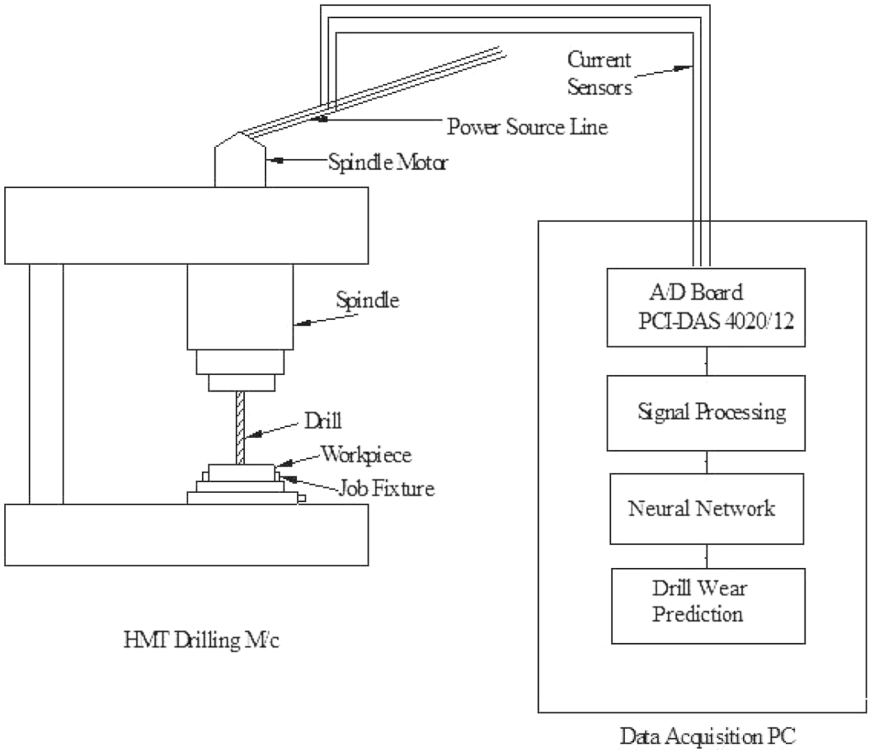

Figure 3. Schematic diagram of the experimental set-up.

of spindle speed were employed: 315, 450, 630, 900 and 1200 rpm. The feed-rate was also incremented gradually to include a set of four values: 0.063, 0.1, 0.16 and 0.25 mm/rev. These three process parameters are taken in 43 different combinations, which are tabulated for mild steel in table 1.

4.3 *Measurement of drill wear*

After each drilling experiment, carried out under the different cutting conditions mentioned in table 1, the drill bit was taken out for wear measurement. Different types of wear can be observed at different parts of the drill bit as shown in figure 4 (Kanai *et al* 1978). *Chisel wear* takes place on the chisel edge due to erosion, whereas *crater wear* takes place due to diffusion at high temperature along the rake face. *Flank wear* takes place on the two flanks or clearance faces of the drill point. Wear mechanisms in flank face are abrasive and adhesive type due to high friction (rubbing) between the flank face and workpiece because of loss of clearance angle on the flank faces. *Outer corner wear* takes place on both the outside corner of the drill bit. This is due to high friction and impact force between the drill and the walls of the machined holes. *Margin wear* takes place on the margins of the drill bit, the causes being similar to those of outer corner wear.

Due to the complexity of the drill bit geometry, it is very difficult to measure all types of wear. Flank wear measurement is comparatively easier. It is progressive in nature, and increases rapidly towards the end of tool life. Crater wear is also a progressive wear. Though the rate of

Table 1. Experimental data for mild steel workpiece.

Sl. no	Drill diameter (mm)	Spindle speed (rpm)	Feed rate (mm/rev)	Motor current sensor value (mv)	Flank wear (mm)
1	8	315	0.1	444	0.23
2	8	315	0.25	502	0.25
3	8	630	0.063	357	0.066
4	8	900	0.16	404	0.085
5	8	1250	0.063	481	0.099
6	8	1250	0.1	516	0.25
7	8	1250	0.16	540	0.262
8	9	315	0.1	469	0.433
9	9	450	0.063	439	0.357
10	9	450	0.1	456	0.384
11	9	450	0.16	477	0.433
12	9	900	0.063	381	0.406
13	10	315	0.1	526	0.064
14	10	315	0.16	528	0.081
15	10	450	0.1	446	0.095
16	10	450	0.16	503	0.14
17	10	630	0.063	342	0.046
18	10	630	0.063	346	0.064
19	10	630	0.063	351	0.081
20	10	630	0.063	372	0.193
21	10	630	0.063	377	0.22
22	10	630	0.063	382	0.226
23	10	630	0.063	392	0.26
24	10	630	0.1	354	0.01
25	10	630	0.1	357	0.027
26	10	630	0.1	358	0.041
27	10	630	0.1	388	0.15
28	10	630	0.16	366	0.145
29	10	630	0.25	382	0.162
30	10	900	0.063	364	0.14
31	10	900	0.1	380	0.148
32	10	900	0.16	442	0.148
33	10	1250	0.063	514	0.145
34	10	1250	0.1	508	0.106
35	10	1250	0.16	583	0.187
36	10	315	0.063	524	0.175
37	10	630	0.063	360	0.142
38	10	630	0.063	366	0.162
39	10	630	0.1	377	0.14
40	10	630	0.1	384	0.147
41	8	315	0.16	452	0.24
42	9	630	0.063	401	0.308
43	10	630	0.063	384	0.241

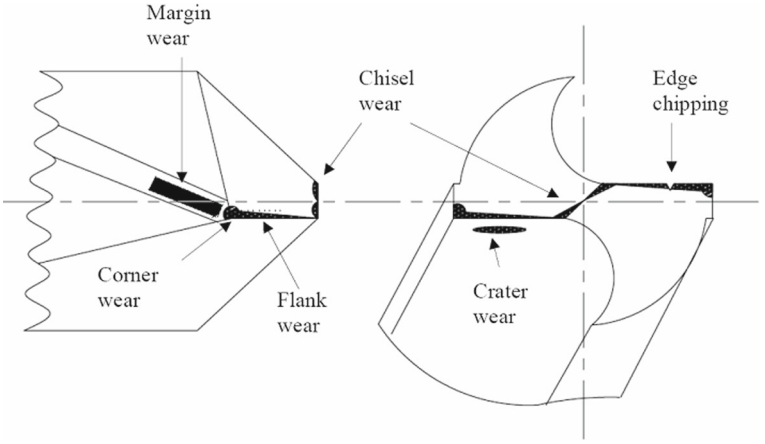

Figure 4. Classification of drill wear.

crater wear is much higher than flank wear, the volume required to be worn away before total destruction is much greater for crater wear than for flank wear. As a consequence, life of a tool subjected to crater or flank wear are approximately the same (Shaw 1992). Therefore, the monitoring of flank wear alone can be considered for the condition monitoring of the drill bit.

An inverted metallurgical microscope (RADICAL, RMM77B), was used to measure the drill flank wear. The magnification of this microscope can be varied from 40X to 600X. Flank wear measurements were manually taken at a magnification of 100X using a micrometer scale fitted to the eyepiece of the microscope. The minimum value of flank wear which could be measured was 10 μm. Two measurements of flank wear length from each of the two cutting edges (one near the corner, and another near the chisel edge) were taken, and average flank wear was calculated. Figure 5 schematically demonstrates how the measurement of average flank wear was obtained. Here the flank wear was quantified by measuring the wear-land length on the flank surface from the cutting edge.

5. Neural network prediction

5.1 *Normalization of the dataset*

The dataset as shown in table 1 was normalized before using it for training and testing the neural network. Normalization is done for each of parameters in the range of 0.1 to 0.9 through Eq. (13).

$$y = 0.1 + 0.8 * \left(\frac{x - x_{\min}}{x_{\max} - x_{\min}} \right), \tag{13}$$

where, $x_{\max}$ = maximum value of the process parameter in the dataset,
$x_{\min}$ = minimum value of the process parameter in the dataset,
x = actual value,
y = normalized value of x.

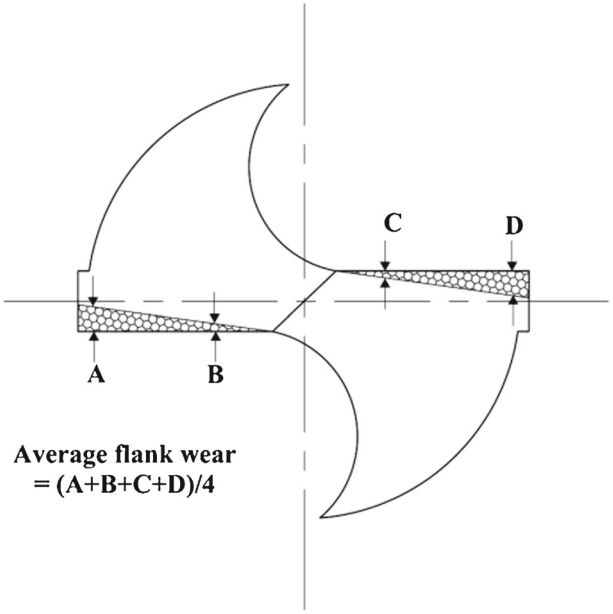

Figure 5. Measurement of average flank wear of a twist drill bit.

5.2 *Neural network training and testing*

Generalized computer codes using C++ have been developed for the MLP neural network trained through both the BP and the PSO algorithms.

After normalization as described above, the dataset containing 43 patterns was segregated randomly into 29 training patterns, and the remaining 14 were used for testing purposes. Four process parameters were used as inputs for the network: drill diameter, spindle-speed, feed-rate and spindle motor current. The corresponding average flank wear was used as the network output.

5.3 *Network performance criterion*

The performance of the MLP neural network trained through both the BP and the PSO algorithms is expressed in terms of the mean square error (MSE), defined by Eq. (14).

$$MSE = \frac{1}{P}\sum_{p=1}^{P}\sum_{k=1}^{M}(d_k^p - c_k^p)^2, \tag{14}$$

where, P is the number of patterns, M is the number of nodes in the output layer, d_k^p is the desired output of the kth node of pth pattern, c_k^p is the calculated output of the kth node of pth pattern.

In case of the PSO trained neural network, the fitness of the ith particle comprising the weights of the network as its variables, is expressed in terms of its ability to minimize the MSE, and is defined through Eq. (15).

$$f_i = \frac{1}{MSE + 1}. \tag{15}$$

6. Results and discussion

6.1 *Parameter settings for both the algorithms*

The main parameters used in the PSO technique are the population size (number of birds), number of generation cycles, the values of the acceleration constants c_1 and c_2 (and hence the value of the constriction factor k) and the inertia weight ω. For the drill wear prediction model developed here, a four-layer neural network is used: four input neurons, three neurons in the first hidden layer, five in the second and one output neuron. A series of trial runs (using a fixed value of parameters c_1, c_2) was carried out to determine the optimum architecture by computing the average errors in training the network upto an iteration value of 1000, as shown in table 2. The stated architecture was best able to minimize the errors during the training process and was thus used as a subject of the comparative study. There are consequently 32 weights to be optimized. This implies that the swarm of birds moves in a 32-dimensional space to search for the weights with the minimum value of MSE. In this model, a population of 10 birds is used while the number of iterations, as stated, is kept to 1000.

As discussed previously, $V_{\max}$ is limited by $X_{\max}$, the dynamic range of each variable on each dimension. Since the weights of the interconnections between the various neurons have been taken to constitute the particle variables, and these weights are intended to be in their normalized form, therefore $X_{\max}$ and hence $V_{\max}$ are restricted to [-1, 1], with the new position X_i always remaining between [0, 1]. The inertia weight ω was decreased linearly from a value of 1.425 to 0.125, the bounds being decided by a series of trial runs. The acceleration constants c_1 and c_2 representing the weighting of the stochastic acceleration terms that pull each particle towards the *pbest* (P_i) and *gbest* (P_g) positions, were found through earlier experience to equal 2.0 for almost all the applications (Elbeltagi *et al* 2005). In this work, slightly offset values of 2.115 and 1.95 for c_1 and c_2, respectively have been found to give a better optimal solution.

In case of BPNN, the momentum coefficient α was kept at 0.25 while the learning rate η was decided by trial and error to be 0.01. A higher value of α helps push the error minimization routine out of any local minima while a smaller value of η ensures that the network slowly but accurately learns to update the weight factors, without compromising too much with the training and testing times.

Table 2. Comparison of network architectures for determining the optimum one.

Network architecture	Average initial training error	Average final training error
4-2-1	0.15035	0.03907
4-3-1	0.14820	0.03490
4-3-3-1	0.34150	0.03569
4-3-4-1	0.40475	0.03667
4-3-5-1	0.40317	0.02979
4-3-5-3-1	0.16661	0.07811
4-3-7-1	0.40221	0.04427
4-4-1	0.15743	0.04017
4-5-1	0.16012	0.04126

Table 3. Determining the optimum balance between individual and social components of intelligence for PSO algorithm.

Sl. no.	c_1	c_2	Initial MSE	Final MSE
1.	2.515	1.95	0.32252	0.319585
2.	2.415	1.95	0.32252	0.316620
3.	2.315	1.95	0.32252	0.119043
4.	2.215	1.95	0.32252	0.104716
5.	2.115	1.95	0.32252	0.019176
6.	2.0	2.0	0.32252	0.223725
7.	2.115	2.10	0.32252	0.134112
8.	2.115	2.20	0.32252	0.116177
9.	2.115	2.30	0.32252	0.318355
10.	2.115	2.40	0.32252	0.319968

6.2 *Improvement of particle fitness*

It was mentioned previously that in each iterative step the particle tries to adapt its velocity (i.e., speed and direction) in order to try and improve its fitness. This is done by moving in the search space in a direction that can bring its fitness closer to the particle with the best fitness, while at the same time keeping track of its previous best fitness level, to avoid being misdirected in the solution space. This fact is succinctly represented in figure 6 which shows how the particle fitness aligns itself with its previous and the best particle fitness levels to reach at the optimum solution. It is interesting to see how the particle fitness fluctuates (within a limit defined by $V_{\max}$) as the particle navigates in the solution space, but does not completely lose track because of the inherent intelligence in remembering individual and global best positions.

6.3 *Balance between individual and social thinking*

It is to be noted that the second term in Eq. (11) represents *cognition*, or the private thinking of the particle when comparing its current position to its own best. The third term in this equation, on the other hand, represents the *social* collaboration among the particles, when comparing a particle's current position to that of the best particle. Table 3 shows that for the network model considered in this study, an optimum balance is required between the cognitive and the social intelligence imparted to a particle to improve its fitness.

As table 3 shows, if any of the two components of particle intelligence tends to outweigh the other beyond the optimum values, the performance of the network drops rapidly, indicating the inability of the particle either to remember its past or to learn from the swarm.

Table 4. Comparison of performance of BPNN and PSO trained neural network.

Sl. no.	Network model	Training MSE	Testing MSE	No. of iterations
1.	BPNN	0.10106	0.04014	572
2.	PSO trained NN	0.04673	0.01907	357

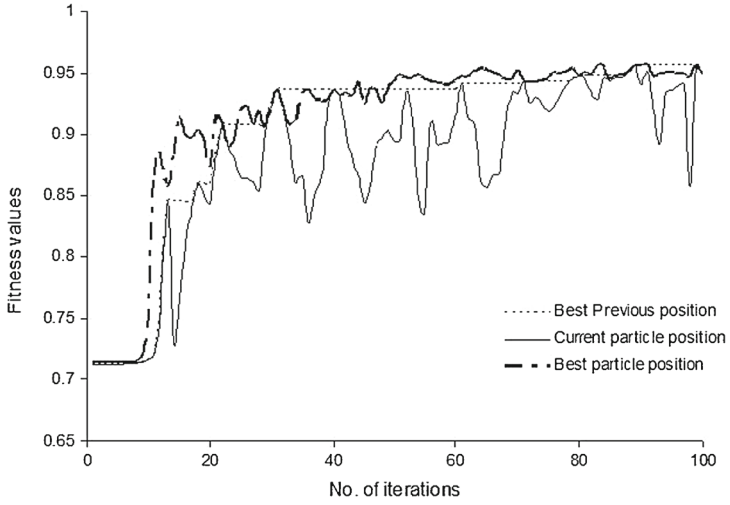

Figure 6. Comparison of current particle, previous best and best particle fitness.

6.4 *Convergence comparison*

The performance of the PSO trained neural network is compared with the standard BPNN model in terms of the training and testing MSE and the computational expense involved in the respective algorithms for the network architecture described earlier. The results are tabulated in table 4. The table shows that the PSO trained neural network gives improved prediction results (testing MSE) and that too in a shorter time as compared to the BPNN. The former is also much better trained than the BPNN which shows the enhanced learning capability of the network arising out of the

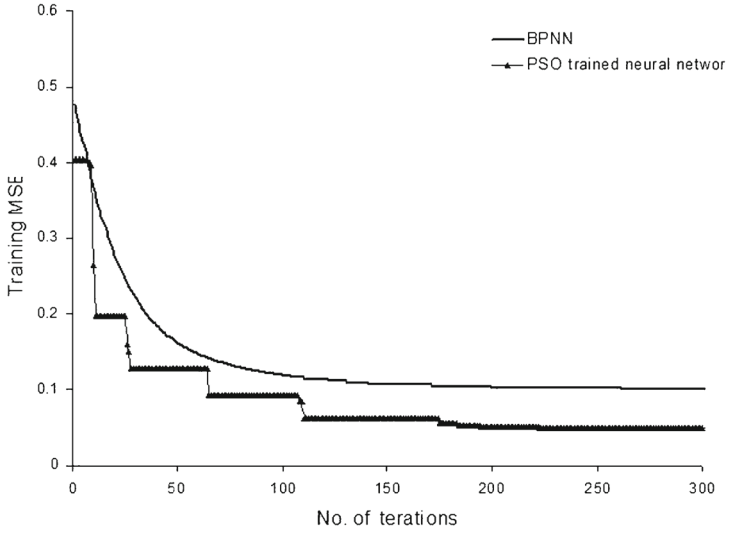

Figure 7. Comparison of training MSE for BPNN and PSO trained neural networks.

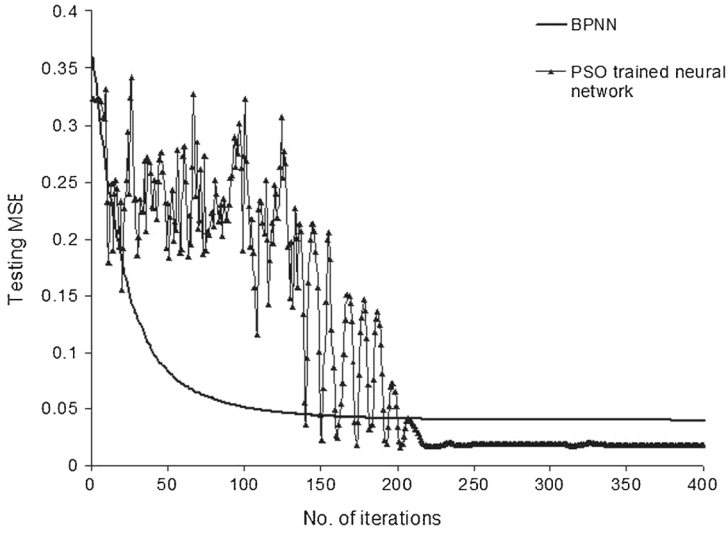

Figure 8. Comparison of testing MSE for BPNN and PSO trained neural networks.

inherent robustness of the PSO algorithm in searching for the optimal solution. These facts are amply demonstrated by figures 7 and 8 also, which show the variation of training and testing MSE, respectively, with increasing number of iterations for the two algorithms.

7. Conclusion

This study highlights the importance of using swarm intelligence through EAs like the PSO in an important machining application i.e., prediction of flank wear in drilling. This is done by comparing the performance of a PSO trained neural network with the standard BPNN. The improved prediction results show the robustness of the PSO in utilizing both individual and group intelligence in searching for the optimal solution in multi-dimensional space in comparison to the conventional gradient descent approach followed by the BP algorithm. The importance of an optimum balance between 'cognition' and 'social collaboration' in contributing towards the total swarm intelligence is also highlighted.

References

Asokan P, Baskar N, Babu K, Prabhaharan G and Saravanan R 2005 Optimization of surface grinding operations using particle swarm optimization technique. *J. Manufacturing Sci. Eng.* 127: 885–892

Clerc M 1999 The swarm and the queen: towards a deterministic and adaptive particle swarm optimization. *Proceedings of Congress on Evolutionary Computation*, Washington DC, pp. 1951–1957

Eberhart R C and Kennedy J 1995 A new optimizer using particle swarm theory. *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, Nagoya: Japan, pp. 39–43

Elbeltagi E, Hegazy T and Grierson D 2005 Comparison among five evolutionary-based optimization algorithms. *Advanced Eng. Informatics* 19: 43–53

Fausett L 1994 *Fundamentals of Neural Networks,* Engle-wood Cliffs, NJ: Prentice-Hall

Feng H M 2006 Self-generation RBFNs using evolutional PSO learning. *Neurocomputing* 70: 241–251

Gaitonde V N and Karnik S R 2012 Minimizing burr size in drilling using artificial neural network (ANN)-particle swarm optimization (PSO) approach. *J. Intelligent Manufacturing* 23: 1783–1793

Haq A N, Sivakumar K, Saravanan R and Karthikeyan K 2006 Particle swarm optimization (PSO) algorithm for optimal machining allocation of clutch assembly. *Int. J. Advance Manufacturing Technol.* 27: 865–869

Jantunen E 2002 A summary of methods applied to tool condition monitoring in drilling. *Int. J. Machine Tools and Manufacture* 42: 997–1010

Kanai M, Inata K, Fujii S and Kanda Y 1978 Statistical characteristics of drill wear and drill life for the standardized performance tests. *Annals of CIRP* 27(1): 61–66

Karpat Y and Ozel T 2005 Hard turning optimization using neural network modelling and swarm intelligence. *Transactions of NAMRI/SME* 33: 179–186

Kennedy J and Eberhart R 1995 Particle swarm optimization. *Proceedings of the IEEE international conference on neural networks*, Perth, Australia, Piscataway NJ: IEEE Service Center, pp. 1942–1948

Malviya R and Pratihar D K 2011 Tuning of neural networks using particle swarm optimization to model MIG welding process. *Swarm and Evolutionary Computation* 1: 223–235

Navalertporn T and Afzulpurkar N V 2011 Optimization of tile manufacturing process using particle swarm optimization. *Swarm and Evolutionary Computation* 1: 97–109

Patra K, Pal S K and Bhattacharyya K 2007 Artificial neural network based prediction of drill flank wear from motor current signals. *Appl. Soft Comput.*, 7: 929–935

Samanta B and Nataraj C 2009 Use of particle swarm optimization for machinery fault detection. *Eng. Appl. Artificial Intelligence*, 22: 308–316

Shaw M C 1992 *Metal cutting principles*. PCBS Publishers & Distributors, 1st Indian edition, ISBN 81-239-0136-4, pp. 224–250

Shi Y and Eberhart R 1998 A modified particle swarm optimizer. *Proceedings of the IEEE international conference on evolutionary computation*, Piscataway, NJ: IEEE Press; pp. 69–73

Shi Y and Eberhart R 2000 Experimental study of particle swarm optimization. *Proceedings of SCI 2000 Conference*, Orlando: FL

Stacey A, Jancic M and Grundy I 2003 *Particle swarm optimization with mutation. Proceedings of IEEE, pp. 1425–1430*

Zhang C and Shao H 2000 An ANN's evolved by a new evolutionary System and its application. *Proceedings of the 39th IEEE Conference on Decision and Control*, Sydney, Australia, pp. 3562–3563

Zhao F, Ren Z, Yu D and Yang Y 2005 Application of an improved particle swarm optimization algorithm for neural network training. *Proceedings of IEEE International Conference on Neural Networks and Brain*, Beijing, China, pp. 1693–1698

Zhou J, Duan Z, Li Y, Deng J and Yu D 2006 PSO-based neural network optimization and its utilization in a boring machine. *J. Material Process Technol.* 178: 19–23