



# Particle swarm intelligence based optimisation of high speed end-milling

**F. Cus, U. Zuperl\***

Faculty of Mechanical Engineering, University of Maribor,  
Smetanova 17, 2000 Maribor, Slovenia

\* Corresponding author: E-mail address: uros.zuperl@uni-mb.si

Received in a revised form 12.06.2008

## ABSTRACT

**Purpose:** Selection of machining parameters is an important step in process planning therefore a new evolutionary computation technique is developed to optimize machining process. This study has presented multi-objective optimization of milling process by using neural network modelling and Particle swarm optimization. Particle Swarm Optimization (PSO) is used to efficiently optimize machining parameters simultaneously in high-speed milling processes where multiple conflicting objectives are present. The goal of optimization is to determine the objective function maximum (predicted cutting force surface) by consideration of cutting constraints.

**Design/methodology/approach:** First, an Artificial Neural Network (ANN) predictive model is used to predict cutting forces during machining and then PSO algorithm is used to obtain optimum cutting speed and feed rates.

**Findings:** During optimization the particles 'fly' intelligently in the solution space and search for optimal cutting conditions according to the strategies of the PSO algorithm. The simulation results show that compared with genetic algorithms (GA) and simulated annealing (SA), the proposed algorithm can improve the quality of the solution while speeding up the convergence process.

**Research limitations/implications:** The experimental results show that the MRR is improved by 28%. Machining time reductions of up to 20% are observed.

**Practical implications:** While a lot of evolutionary computation techniques have been developed for combinatorial optimization problems, PSO has been basically developed for continuous optimization problem. PSO can be an efficient optimization tool for solving nonlinear continuous optimization problems, combinatorial optimization problems, and mixed-integer nonlinear optimization problem.

**Originality/value:** An algorithm for PSO is developed and used to robustly and efficiently find the optimum machining conditions in end-milling. This paper opens the door for a new class of EC based optimization techniques in the area of machining. This paper also presents fundamentals of PSO optimization techniques.

**Keywords:** Machining; End-milling; Particle Swarm Optimization

**Reference to this paper should be given in the following way:**

F. Cus, U. Zuperl, Particle Swarm Intelligence based optimisation of high speed end-milling, Archives of Computational Materials Science and Surface Engineering 1/3 (2009) 148-154.

## MANUFACTURING AND PROCESSING OF ENGINEERING MATERIALS

### 1. Introduction

Increasing productivity, decreasing costs, and maintaining high product quality at the same time are the main challenges

manufacturers face today. The proper selection of machining parameters is an important step towards meeting these goals and thus gaining a competitive advantage in the market [15]. Many

researchers have studied the effects of optimal selection of machining parameters of end milling [9]. This problem can be formulated and solved as a multiple objective optimization problem [6]. In practice, efficient selection of milling parameters requires the simultaneous consideration of multiple objectives, including maximum tool-life, desired roughness of the machined surface, target operation productivity, metal removal rate, etc [2]. In some instances, parameter settings that are optimal for one defined objective function may not be particularly suited for another objective function. Solving multi-objective problems with traditional optimization methods is difficult and the only way is to reduce the set of objectives into a single objective and handle it accordingly. Therefore evolutionary algorithms such as genetic algorithms (GA) and particle swarm optimization (PSO) are more convenient and usually utilized in multiobjective optimization problems. These methods are summarized by [3]. The PSO is an efficient alternative over other stochastic and population-based search algorithms, especially when dealing with multi-objective optimization problems. It is relatively easy to implement and has fewer parameters to adjust compared to genetic algorithms.

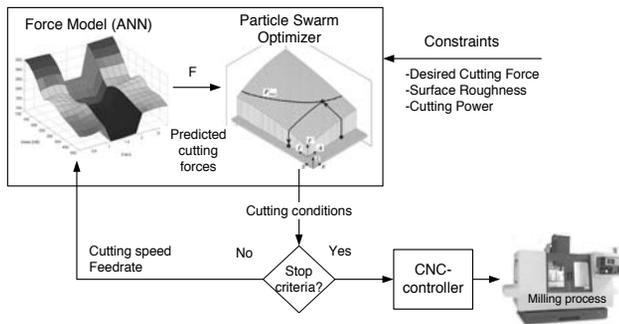


Fig. 1. PSO based neural network optimization scheme

In our research neural networks are used to model complex relationships in the process, and an integrated system of neural networks and particle swarm optimizer is utilized in solving multi-objective problems observed in milling operations (Fig. 1).

## 2. Particle swarm optimization

Particle Swarm Optimization (PSO) is a relatively new technique, for optimization of continuous non-linear functions [11]. It was first presented in 1995 [12].

They discovered the method through simulation of a simplified model, the graceful but unpredictable movement of a bird swarm [11]. Mohan developed a swarm model with simple rules and generated complicated swarm behaviour [10]. These researches are called "Swarm Intelligence"

PSO is a very simple concept, and paradigms are implemented in a few lines of computer code. It requires only primitive mathematical operators, so is computationally inexpensive in terms of both memory requirements and speed. PSO has been recognized as an evolutionary computation technique [12] and has features of both genetic algorithms (GA)

and evolution strategies (ES). Other evolutionary computation (EC) techniques such as genetic algorithm also utilize some searching points in the solution space. It is similar to a GA in that the system is initialized with a population of random solutions.

While GA can handle combinatorial optimization problems, PSO can handle continuous optimization problems. However, unlike a GA each population individual is also assigned a randomized velocity, in effect, flying them through the solution hyperspace. PSO has been expanded to handle also the combinatorial optimization problems. As is obvious, it is possible to simultaneously search for an optimum solution in multiple dimensions. Unlike other EC techniques, PSO can be realized with only small program. Natural creatures sometimes behave as a swarm. One of the main goals of artificial life researches is to examine how natural creatures behave as a swarm and reconfigure the swarm models inside a computer.

PSO has two simple concepts. Swarm behaviour can be modelled with a few simple rules. Even if the behaviour rules of each individual (particle) are simple, the behaviour of the swarm can be very complex. The behaviour of each agent inside the swarm can be modelled with simple vectors. This characteristic is the basic concept of PSO.

According to Mohan examination [10], people utilize two important kinds of information in decision process. The first one is their own experience; they have tried the choices and know which state has been better so far, and they know how good it was. Therefore each person decides his decision using his own experiences and other peoples' experiences. This characteristic is another basic concept of PSO. The applications of PSO are: Neural network learning algorithms, Rule extraction in fuzzy neural networks, computer controlled milling optimization, power and voltage control [1]. Application of PSO to other fields is at the early stage. More applications can be expected. Most of papers are related to the method itself, and its modification and comparison with other EC methods [1, 4, 8, 10].

## 3. Basic of particle swarm optimization

PSO is developed through simulation of bird flocking in two-dimension space. The position of each agent is represented by XY axis position and also the velocity is expressed by vx (the velocity of X axis) and vy (the velocity of Y axis). Modification of the agent position is realized by the position and velocity information. Bird flocking optimizes a certain objective function. Each agent knows its best value so far (pbest) and its XY position. This information is analogy of personal experiences of each agent. Further, each agent knows the best value so far in the group (gbest) among (pbests). This information is analogy of knowledge of how the other agents around them have performed. Each agent tries to modify its position using the following information: - the current positions (x, y), - the current velocities (vx, vy), - the distance between the current position and (pbest) - the distance between the current position and (gbest). This modification can be represented by the concept of velocity.

Velocity of each agent can be modified by the following equation:

$$v_i^{k+1} = w \cdot v_i^k + c_1 \cdot \text{rand}_1 \cdot (pbest_i - s_i^k) + c_2 \text{rand}_2 \cdot (gbest - s_i^k) \quad (1)$$

where,

- $v_{ik}$  : velocity of agent  $i$  at iteration  $k$ ,
- $w$  : weighting function,
- $c_j$  : weighting factor,
- $rand$  : random number between 0 and 1,
- $s_{ik}$  : current position of agent  $i$  at iteration  $k$ ,
- $pbest_i$  : pbest of agent  $i$ ,
- $gbest$  : gbest of the group.

The following weighting function is usually utilized (1):

$$w = w_{max} - \frac{w_{max} - w_{min}}{iter_{max}} \times iter \tag{2}$$

where,

- $w_{max}$  : initial weight,
- $w_{min}$  : final weight,
- $iter_{max}$  : maximum iteration number,
- $iter$  : current iteration number.

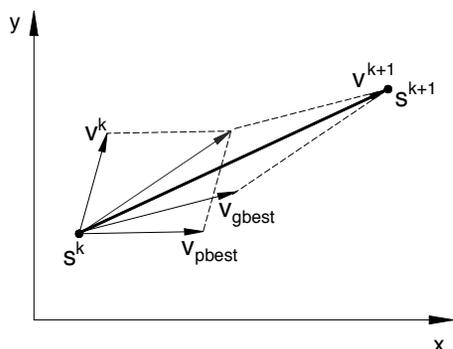


Fig. 2. Concept of modification of a searching point according to PSO algorithm

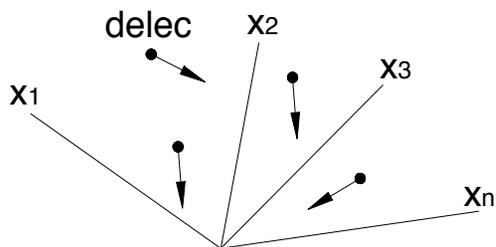


Fig. 3. Concept of searching with agents in a solution space

Using the above equation, a velocity, which gradually gets close to  $pbest$  and  $gbest$  can be calculated. The current position (searching point in the solution space) can be modified by the following equation:

$$S_i^{k+1} = S_i^k + v_i^{k+1} \tag{3}$$

Figure 2 shows a concept of modification of a searching point by PSO algorithm.

Figure 3 shows a searching concept with agents in a solution space. Each agent changes its current position using the integration of vectors as shown in Figure 2.

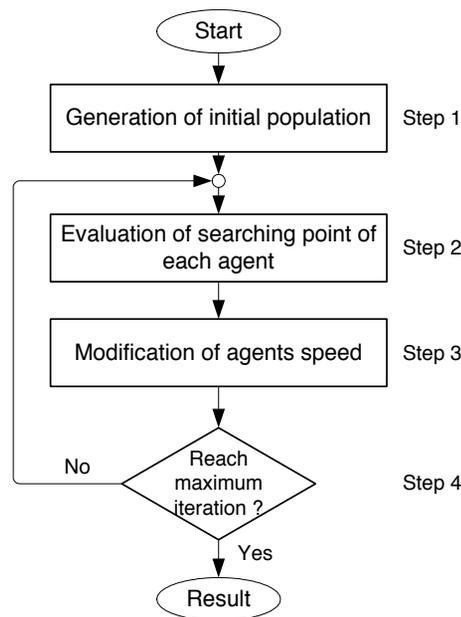


Fig. 4. A general PSO algorithm

The general flow chart of PSO method can be described as follows:

- Step 1: Generation of initial condition of each agent Initial searching points ( $s_{i0}$ ) and velocities ( $v_{i0}$ ) of each agent are generated randomly within the allowable range. The current searching point is set to  $pbest$  for each agent. The best-evaluated value of  $pbest$  is set to  $gbest$  and the agent number with the best value is stored.
- Step 2: Evaluation of searching point of each agent The objective function value is calculated for each agent. If the value is better than the current  $pbest$  of the agent, the  $pbest$  value is replaced by the current value. If the best value of  $pbest$  is better than the current  $gbest$ ,  $gbest$  is replaced by the best value and the agent number with the best value is stored.
- Step 3: Modification of each searching point The current searching point of each agent is changed using (1)(2)(3).
- Step 4: Checking the exit condition The current iteration number reaches the predetermined maximum iteration number, then exit. Otherwise, go to step 2.

The features of the PSO procedure can be summarized as follows:

1. As shown in (1)(2)(3), PSO can essentially handle continuous optimization problem.
2. PSO utilizes several searching points like genetic algorithm (GA) and the searching points gradually get close to the optimal point using their  $pbest$ s and the  $gbest$ .
3. The above concept is explained using only XY-axis (two-dimension space). However, the method can be easily applied to n-dimension problem.

Figure 4 shows the general flow chart of PSO strategy.

With the objective to improve the rate of convergence of the PSO algorithm, researchers [10] proposed some modifications to the existing PSO. These modifications relate to the use of best ever position, maximum velocity, inertia, craziness, elite particle and elite velocity.

**Maximum velocity**

Based on numerical experimentation, we select a starting value  $v_{0max} = 100$  and then decrease this value by the fraction  $\psi$ . Numerical experimentation suggests that this approach improves the convergence rate of the algorithm.

$$v_{max}^{k+1} = \psi \cdot v_{max}^k ; \quad 0 \leq \psi \leq 1 \quad (4)$$

**Best ever position**

It means that the best ever position in the swarm replace the best position of the swarm. This procedure increases the pressure exerted on the agent to converge towards the global optimum without additional function evaluations. Numerical experimentation suggests that this approach improves the convergence rate of the algorithm.

**Craziness**

Craziness operator mimics the random (temporary) departure of birds from the flock. Craziness has some similarity to the mutation operator in the genetic algorithm, since it increases the directional diversity in the flock. ‘‘Crazy’’ birds explore previously uncovered ground, which in general increases the probability of finding the optimum.

**Elite agent**

The concept is borrowed from the GA where the gene with the best fitness never vanishes. The elite particle replaces the worst positioned particle in the swarm. Numerical results indicate that the elite particle improves convergence rates.

**4. Adaptation of PSO technique to milling optimization problem**

In order to search for optimal process parameters, neural network model of cutting force was integrated with particle swarm optimizer. The architecture of system is shown in Figure 1.

Multiple neural network models are grouped together under the general neural network model, and its output is fed into the multi-objective particle swarm optimizer where the objective functions and constraints are defined. PSO algorithm is initiated with randomly generated particles that are optimum solution candidates. Neural network model predicts cutting forces for each of the particles. Predicted forces are used in calculation of objective function in which PSO tries to maximize.

The optimization process executes in two phases. In first phase, the neural prediction model on the basis of recommended cutting conditions generates 3D surface of cutting forces, which represent the feasible solution space for the PSO algorithm. The cutting force surface is limited with planes which represent the constraints of cutting process. Seven constraints, which arise from technological specifications, are considered during the optimization process. Those constraints are listed in Table 1. Here we are faced with a non-linear objective function along with a set of inequality constraints that may also be highly non-linear. The presence of non-linearities creates additional problems for finding the minimum.

The biggest problem in the implementation of PSO technique is the construction of a fitness (objective) function which adequately epitomizing the nature of the problem. The objective function serves as the only link between the optimization problem and the PSO-algorithm. For the objective function a surface of max. cutting forces is selected, generated by ANN.

PSO algorithm generates a swarm of particles on the cutting force surface during the second phase. Swarm of particles flies over the cutting force surface and search for maximal cutting force. The coordinates of a particle which has found the maximal (but still allowable) cutting force represent the optimal cutting conditions. Figure 5 shows the PSO flowchart of optimization of milling process.

The optimization process is depicted by the following steps:

1. Generation and initialization of an array of 50 particles with random positions and velocities. Velocity vector has 2 dimensions, feed rate and spindle speed. This constitutes Generation 0.
2. Evaluation of objective (cutting force surface) function for each particle.
3. The cutting force values are calculated for new positions of each particle. If a better position is achieved by particle, the pbest value is replaced by the current value. 1.
4. Determination if the particle has found the maximal force in the population. If the new gbest value is better than previous gbest value, the gbest value is replaced by the current gbest value and stored. The result of optimization is vector gbest (feedrate, spindle speed).
5. Computation of particles’ new velocity.
6. Update particle’s position by moving towards maximal cutting force.
7. Steps 1 and 2 are repeated until the iteration number reaches a predetermined iteration.

Table 1. Used constraints and their expressions

| Constraints         | Expression   | Variables  |
|---------------------|--|--|
| Feedrate            | $f_{min} \leq \frac{1000 \cdot z}{\pi \cdot D} v_c$  | $z$ – number of teeth,<br>$f_z$ – feeding per tooth,<br>$D$ – diameter of cutter |
| Spindle speed       | $n_{min} \leq \frac{1000}{\pi \cdot D} v_c \leq v_c$ | $v_c$ – cutting speed  |
| Radial depth of cut | $R_D \leq ae_{max}$                                  | $ae_{max}$ – max. radial depth of cutting  |
| Axial depth of cut  | $A_D \leq ap_{max}$                                  | $ap_{max}$ – max. axial depth of cutting   |
| Power of cutting    | $\frac{MRR \cdot Kc}{60} \leq P_{dov}$               | $MRR$ – metal removal rate,<br>$Kc$ – specific cutting force                     |
| Cutting force       | $F(f, n) \leq F_{ref}$                               | $F_{ref}$ – desired cutting force  |
| Surface roughness   | $R_a \leq R_{a ref}$                                 | $R_{a ref}$ – desired surface roughness  |

Figure 6 shows simplified principle of optimization of cutting conditions by the use of PSO. In this case the swarm flies over the force surface and searches for optimal feeding at constant cheap

cross-section A. Optimal feed rate is located at the cross-section of the following three planes: cutting force surface, plane with the constant cheap cross-section (vertical plane) and the desired cutting force plane. The coordinate of the particle which is the nearest to mentioned cross-section represent the optimal feed rate.

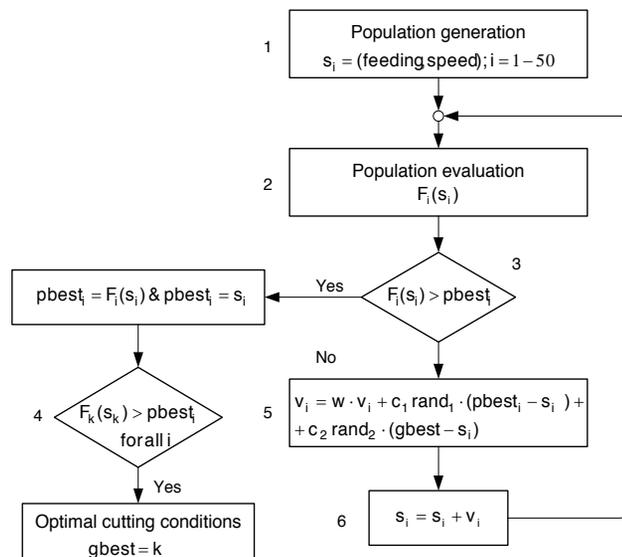


Fig. 5. PSO algorithm for optimization of cutting conditions

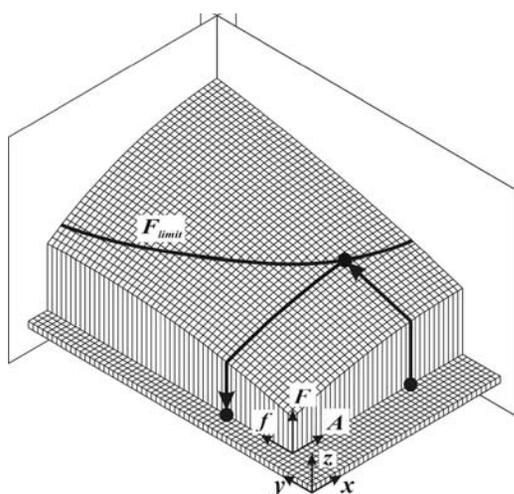


Fig. 6. Optimal feeding searching procedure

### 5. Computer software for PSO optimization

A collection of Matlab's m- files forms PSO software for optimization. This software can be used for optimization of arbitrary non-linear system. The required input values can be

inserted in a software window shown on Figure 7. On the left side of the window, the parameters required for executing PSO algorithm can be set. The result of optimization (optimal cutting parameters) is shown in the middle of the window.

The process of optimization is monitored on graph.

### 6. PSO optimization of cutting conditions-Test case

The repeatability and robustness of the PSO algorithm, is demonstrated with the following test case. To examine the stability and robustness of the proposed optimization strategy, the system is first analyzed by simulations, then the system is verified by experiments on a CNC milling machine (type HELLER BEA1) for Ck 45 and 16MnCrSi5 XM steel workpieces [13,14]. The ball-end milling cutter (R220-20B20-040) with two cutting edges, of 20 mm diameter and 10° helix angle was selected for experiments. The following cutting parameters and constraints are used: milling width  $R_D=3$  mm, milling depth  $A_D=5$  mm, cutting speed  $v_c=80$  m/min,  $n \leq 2000$  min<sup>-1</sup>,  $10 \leq f \leq 900$  mm/min,  $F(f, n) \leq F_{ref} = 600$  N. The objective function is determined by neural cutting force model (cutting force simulator). The goal of this case is to maximize the objective function under given constraints [8,13]. This problem is solved using the PSO algorithm. In PSO, 50 particles were used and search continues until error gradient is smaller than a specified value [5]. Matlab® simulates the trained neural network to predict cutting forces at given cutting conditions and these values are used to calculate the objective function which PSO algorithm attempts to maximize. The results are tabulated in Table 2. Each run corresponds to each time the program is run to find the optimum machining parameters. Table 2 shows optimal cutting conditions along with the number of generations it took to reach that optimum. This optimization method has higher convergence, unlike traditional methods and it is always successful in finding the global optimum. The machining time is reduced by 35% as a result of optimizing the feed and speed. A sample of the evolution of the particle swarm is presented in Figure 9.

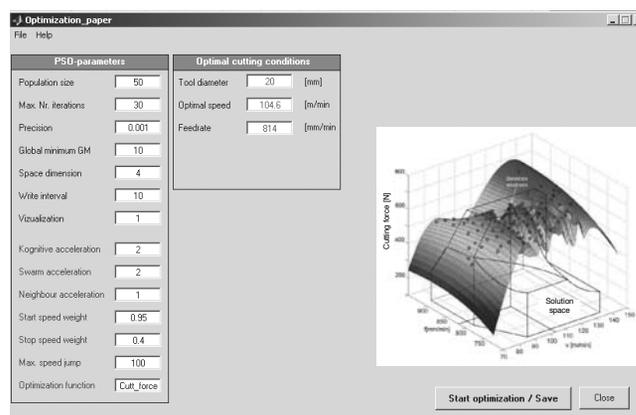


Fig. 7. Software window for PSO optimization

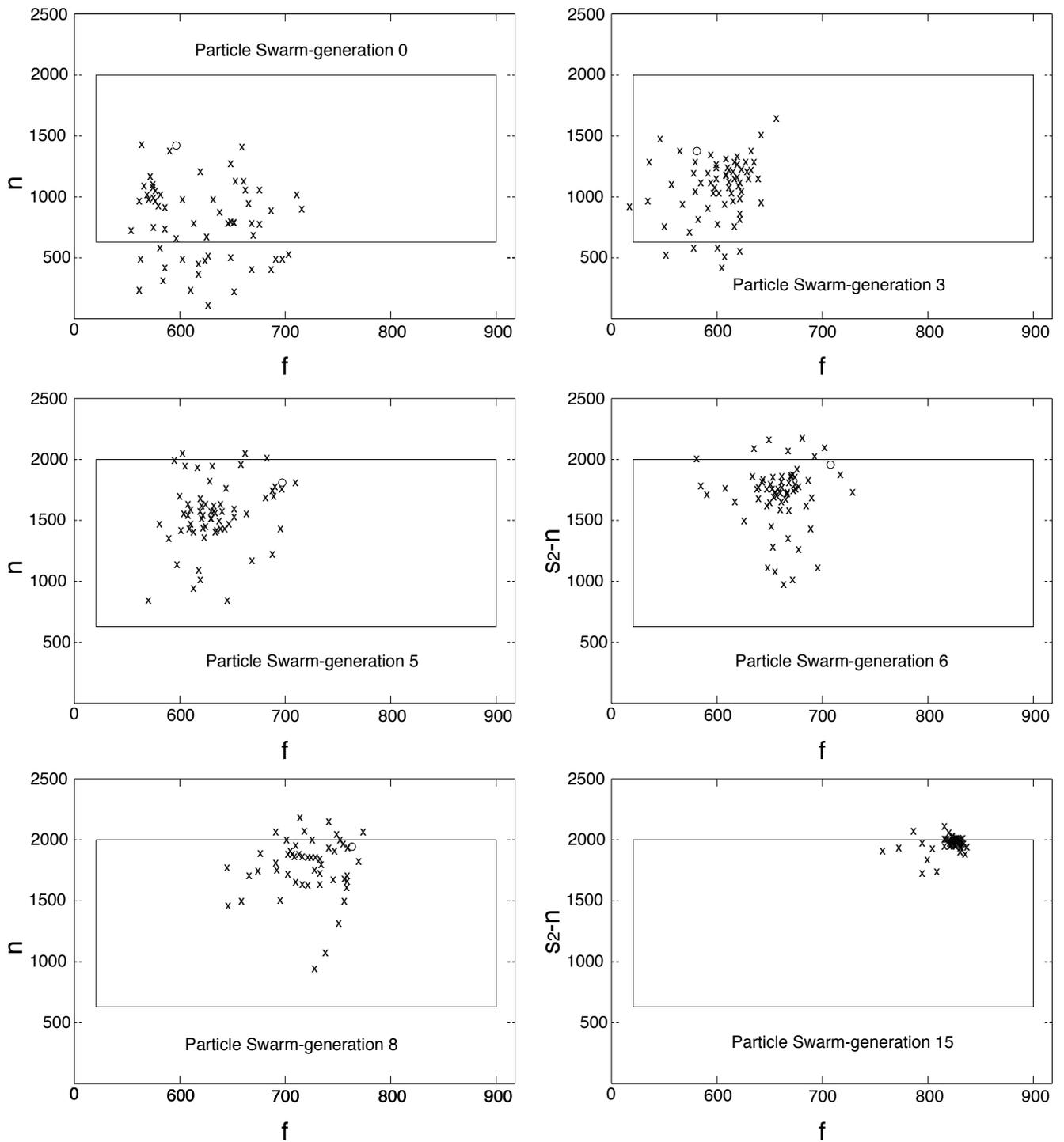


Fig. 8. PSO simulation

Figure 8 shows a typical particle swarm movement pattern toward the optimum solution. Generation 0 represents the random initialization of the particle's coordinates in the solution space. In subsequent generations, the swarm is tracked with "x". The best member in population is presented with "O". The solution space

is graphed by the rectangle. An acceptable solution has to be found within this two-dimensional space. The third constraint on force is also active and as such is not part of these illustrations. By simulations the robustness of the algorithm is demonstrated.

Table 2.  
Repeatability of results

| Run | $n$ [ $\text{min}^{-1}$ ] | $f$ [ $\text{mm}/\text{min}$ ] | $F$ [N] | Nr. of iterations |
|-----|---------------------------|--------------------------------|---------|-------------------|
| 1   | 1998                      | 808.2                          | 598     | 22                |
| 2   | 1995                      | 810.1                          | 600     | 25                |
| 3   | 1997                      | 811.2                          | 600     | 28                |
| 4   | 1997                      | 819.7                          | 598     | 32                |
| 5   | 2000                      | 819.1                          | 598     | 22                |
| 6   | 1999                      | 819.2                          | 598     | 31                |
| 7   | 1999                      | 808                            | 597     | 26                |
| 8   | 1998                      | 808.8                          | 598     | 21                |
| 9   | 1998                      | 808.9                          | 598     | 32                |
| 10  | 2000                      | 808.1                          | 597     | 30                |

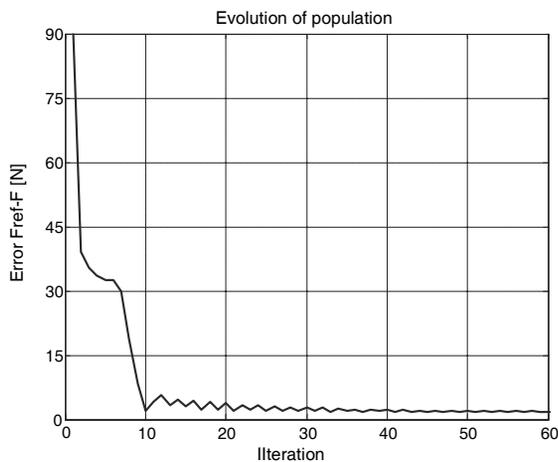


Fig. 9. Decrease of optimization error during swarm evolution

## 7. Conclusions and future research

This study has presented multi-objective optimization of milling process by using neural network modelling and Particle swarm optimization. A neural network model was used to predict cutting forces during machining and PSO algorithm was used to obtain optimum cutting speed and feed rate. A set of seven constraints were used during optimization. Next, neural force model was used to predict the objective function. Next, the PSO algorithm is used to optimize both feed and speed for a typical case found in industry. The experimental results show that the MRR is improved by 28%. Machining time reductions of up to 20% are observed. This paper opens the door for a new class of EC based optimization techniques in the area of machining. This paper also presents fundamentals of PSO optimization techniques. While a lot of evolutionary computation techniques have been developed for combinatorial optimization problems, PSO has been basically developed for continuous optimization problem. PSO can be an efficient optimization tool for solving nonlinear continuous optimization problems, combinatorial optimization problems, and mixed-integer nonlinear optimization problem.

## References

- [1] M.A. Abido, Optimal Power Flow Using Particle Swarm Optimization, *International Journal of Electrical Power & Energy Systems* 24 (2002) 563-571.
- [2] J. Balic, A new NC machine tool controller for step-by-step milling, *International Journal of Advanced Manufacturing Technology* 18 (2001) 399-403.
- [3] J. Balic, Optimization of cutting process by GA approach, *Robotics and Computer Integrated Manufacturing* 19 (2003) 113-121.
- [4] C. Chen, M. Zhibin, An intelligent approach to non-constant feed rate determination for high-performance 2D CNC milling, *International Journal of Manufacturing Technology and Management* 9 (2006) 219-236.
- [5] F. Cus, U. Zuperl, V. Gecevaska, High speed end-milling optimization using Particle Swarm Intelligence, *Journal of Achievements in Materials and Manufacturing Engineering*, 22/2 (2007) 75-78.
- [6] F. Cus, U. Zuperl, E. Kiker, M. Milfelner, Adaptive controller design for feedrate maximization of machining process, *Journal of Achievements in Materials and Manufacturing Engineering* 17 (2006) 237-240.
- [7] L.A. Dobrzański, K. Gołombek, J. Kopač, M. Soković, Effect of depositing the hard surface coatings on properties of the selected cemented carbides and tool cermets, *Journal of Materials Processing Technology* 157-158 (2004) 304-311.
- [8] W. Grzesik, J. Rech, T. Wanat, Surface integrity of hardened steel parts in hybrid machining operations, *Journal of Achievements in Materials and Manufacturing Engineering* 18 (2006) 367-370.
- [9] J. Kopac, Influence of high speed cutting on the structure of machined high speed steel material, *Proceedings of the 11<sup>th</sup> Scientific Conference "Contemporary Achievements in Mechanics, Manufacturing and Materials Science", CAM<sup>3</sup>S'2005, Gliwice-Zakopane, 2005, (CD-ROM)*.
- [10] E. Ozcan, C. Mohan, Analysis of a simple Particle Swarm Optimization system, *Intelligent Engineering Systems Through Artificial Neural Networks* 8 (1998) 253-258.
- [11] Y.Y. Peng, A Discrete Particle Swarm algorithm for optimal polygonal approximation of digital curves, *Journal of Visual Communication and Image Representation* 15 (2004) 241-260.
- [12] Y. Shi, R. Eberhart, 'Parameter selection in particle swarm optimization', In *Evolutionary Programming VII: Proc. EP98*, New York: Springer-Verlag, 1998, 591-600.
- [13] M. Sokovic, M. Cedilnik, J. Kopac, Use of 3D-scanning and reverse engineering by manufacturing of complex shapes, *Proceedings of the 13<sup>th</sup> International Scientific Conference "Achievements in Mechanical and Materials Engineering", AMME'2005, Gliwice-Zakopane, 2005, 601-604*.
- [14] S. Zhang, A. Xing, L. Jianfeng, F. Xiuli, Failure analysis on clamping bolt of milling cutter for high-speed machining, *International Journal of Machining and Machinability of Materials* 1 (2006) 343-353.
- [15] U. Zuperl, F. Cus, M. Milfelner, Fuzzy control strategy for an adaptive force control in end-milling, *Journal of Materials Processing Technology* 164 (2005) 1472-1478.