Scientific Journal of Impact Factor (SJIF): 4.72

e-ISSN (O): 2348-4470
p-ISSN (P): 2348-6406

International Journal of Advance Engineering and Research Development

Volume 4, Issue 12, December -2017

Optimization of Cutting Parameters to Minimize Surface Roughness in the Hard Turning Process using Particle Swarm Optimization Algorithm

Dipeshkumar Bansilal Trivedi

Mechanical Engineering Department, Government Engineering College, Godhra, Gujarat, India

Abstract — *Surface finish plays an important role in the manufacturing industries as it impacts on safety and reliability of the components being manufactured. Therefore, it is evident that the parametric optimization is important. In this present study, an advanced optimization technique called Particle Swarm Optimization (PSO) is used in the parametric optimization of the hard turning of EN-31 steel. Cutting parameters such as cutting speed, feed rate, depth of cut and nose radius are optimized for the surface roughness as an objective function. The results verified the effectiveness of PSO optimization algorithm on the hard turning process parameters.*

Keywords*- Hard turning process; Parametric optimization; Particle swarm optimization algorithm*

I. INTRODUCTION

Machining of hardened steel is important aspect and matter of current research among the researchers. Conventionally, hardened steels were machined with the grinding process but it is time consuming and having limitations with the reference to geometry of the surfaces produced. Therefore, hard turning is a substitute of grinding process which gives many benefits [1,2]. Effective material removal depends upon selection of machine parameters, nature of the work-piece material and tool materials. In this process, there are several input parameters such as cutting speed, feed rate, depth of cut, nose radius, mechanical properties of work material, tools materials geometry and compositions are considered. On the other side, the output parameters can be enlisted as productivity, tool life, accuracy, cutting forces, temperature and surface integrity. We cannot simply rely upon the randomization while selection of input parameters because optimum setting of these parameters is highly desirable. So and that’s why selection off most suitable and optimal process parameters are important to achieve desired objectives [3]. In the following paragraph, the research work done in the past by various researchers has been described.

D’Mello et al. [4] studied optimization techniques in high speed turning of Ti-6Al-4V and concluded that PSO technique was found better as compared to other optimization techniques used. They had also observed that use of high cutting speed, lower feed rate, medium depth off cut and high flank wear were the conditions of getting minimum surface roughness. Rao et al. [5] used teaching leaning based optimization algorithm to optimize process parameters for multi-pass turning process. They have concluded that the TBLO is more effective as compared to other optimization algorithms. Nithynandam et al. [6] applied Taguchi method for optimization of turning process. They have stated that minimum R_a was recorded at cutting speed of 175 m/min, feed rate of 0.05 mm/rev, nose radius of 1.5 mm and diameter of 0.5 mm. Gao et al. [7] optimized milling process parameters through cellular PSO technique. The suggested method combines two constraints handling techniques, including the penalty function method and the constraints handling strategy of PSO. The technique was novel and effective over other techniques used in the past. Baskar et al. [8] described development and utilization of an optimization system, to achieve optimum milling parameters. On the basis of results, they have suggested to use PSO optimization algorithm for getting better results in milling process. Homani et al. [9] conducted experiments on turning process and used artificial neural network(ANN) and genetic algorithm (GA) to optimize surface roughness. GA gave optimized value of cutting speed=45 m/min, feed rate=0.22 mm/rev, nose radius=1.6 mm and approach angle= 600. Raja et al. [10] studied PSO technique for parametric optimization in turning process. The variation was very less among actual and predicted surface roughness and 85% overall accuracy was achieved for all the work piece materials used in this study.

As per literature review studied, it is found that there is a scope for use of PSO optimization algorithm in the parametric optimization of hard turning. The objective of this study is to carry experimentation and use of advanced PSO technique to optimize surface roughness of machined surface.

Nomenclature

c_1, c_2	acceleration parameter (for PSO algorithm)
p	number of particle (For PSO algorithm)
v	particle velocity (for PSO algorithm)
w	inertia weight (for PSO algorithm)

II. MATERIALS AND EXPERIMENTAL METHOD

2.1 Work-piece material

EN-31 steel was used in turning experiments. Table 1 describes the chemical compositions of the work piece materials. Work-piece of ϕ 25 mm and 100 mm length was used in this study.

2.2 Machine Tool

An automatic lathe machine equipped with experimental setup was used. The cutting tool utilized was HSS tool, see Figure 1.

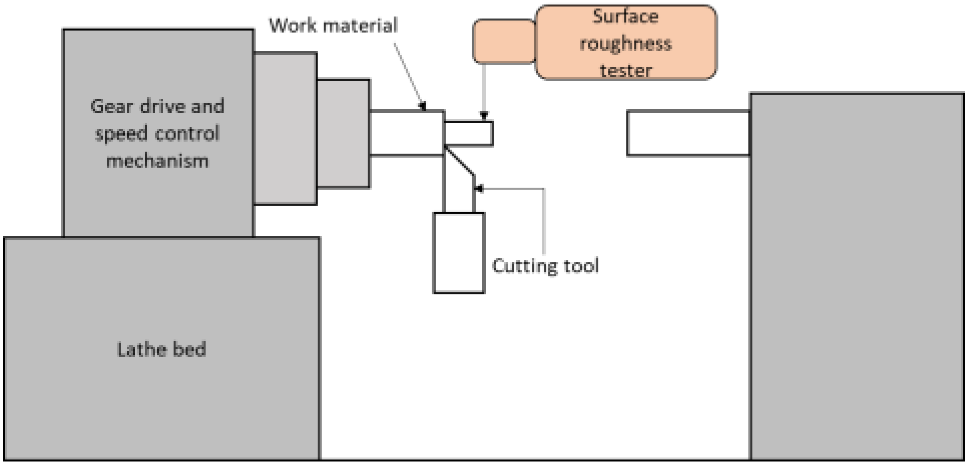

Figure 1. Schematic of experimental setup

2.3 Cutting conditions

Based on the basis of preliminary experiments conducted, the feasible range of cutting parameters for a given cutting tool-work piece system were selected as shown in Table-2.

2.4 Experimental procedure

Turning experiments were performed as per Taguchi's L_{25} design of experiments by using different cutting conditions as shown in Table 2.

2.5 Surface roughness measurement

Mitutoyo Surface roughness tester was used to measure surface roughness of the turned work-piece and the measured values are presented in Table 3.

Table 1. Chemical constituents of steel alloy (en-31) work-piece

Composition	C	Si	Mn	Cr	Co	S	P
Wt. %	0.95-1.2	0.10-0.35	0.30-0.75	1.0-1.6	0.025	0.040	0.04

Table 2. Factors and their levels

Factors	Level 1	Level 2	Level 3	Level 4	Level 5
Cutting Speed	40	76	113	150	189
Feed rate	0.06	0.08	0.10	0.12	0.15
Depth of Cut	0.2	0.3	0.4	0.5	0.6
Nose Radius	0.4	0.6	0.8	1.0	1.2

III. PARTICLE SWARM OPTIMIZATION (PSO)

It is kwon that the particle swarm optimization (PSO) is an evolutionary multiplication technique for solving comprehensive optimization problems established by Kennedy and Eberhart [11]. It is a computational technique through individual improvement plus population co-operation and competition which is established on the simulation of simplified social models, such as bird flocking, fish schooling and the swarm theory. Researchers originate that the synchrony of animal's behaviour was through sustaining optimal spaces between individual members and their neighbours. Thus, velocity plays an important role of adjusting each other for the optimal distances. Additionally, they had replicated the scenario in which birds search for food and observe their social behaviour. They observed that in order to find food the individual members determine their velocities by two factors first their own past experience and the second top experience of all other members.

It reveals common evolutionary computation attributes including initialization with a population of random solutions and searching for optima by updating generations. Potential solutions, called 'birds' or 'particles', are then "flown" through the problem space by following the current optimum particles. The particle swarm concept was initiated as a simulation of a simplified social system. The original intent was to graphically simulate the graceful but unpredictable composition of a bird flock. Each particle keeps track of its coordinates in the problem space, which are associated with the best solution (fitness) it has achieved so far. This value is called 'pBest'. Another "best" value that is traced by the global version of the particle swarm optimization is the overall finest value and its location obtained so far by any particle in the population. This location is called 'gBest'.

The particle swarm optimization model consists of, at each step, changing the velocity (i.e. accelerating) of each particle toward its 'pBest' and 'gBest' locations (global version of PSO). Acceleration is weighted by an arbitrary term with separate arbitrary numbers being generated for acceleration toward 'pBest' and 'gBest' locations. The updates of the particles are accomplished as per the following equations [11].

$$V_{i+1} = wV_i + c_1 r_1 \left(pBest_i - X_i \right) + c_2 r_2 \left(gBest_i - X_i \right) \tag{1}$$

$$X_{i+1} = X_i + V_{i+1} \tag{2}$$

Equation (1) determines a new velocity (V_{i+1}) for each particle (potential solution) based on its previous velocity, the best position it has achieved ('pBest') so far, and the universal best location ('gBest'), the population has achieved. Equation (2) updates individual particle's position (Xi) in solution hyperspace. The two arbitrary numbers 'r_1' and 'r_2' in equation (1) are independently generated in the range [0, 1]. The acceleration constants 'c_1' and 'c_2' in equation (1) represent the weighting of the stochastic hastening terms that pull each particle towards 'pBest' and 'gBest' positions. 'c_1' represents the assurance the particle has in itself (cognitive parameter) and 'c_2' represents the assurance the particle has in swarm (social parameter). Thus, alteration of these constants changes the amount of tension in the system. Low values of the constants allow particles to roam far from target regions before being tugged back, while high values result in sudden movement toward, or passed through target regions [12]. Particle's velocities on each dimension are confined to an extreme velocity parameter V_{max}, specified by the user. If the sum of accelerations would cause the velocity on that dimension to exceed V_{max}, then the velocity on that dimension is limited to V_{max}. Also if X_{i+1} is greater than maximum value for the corresponding design variable than it is set to the maximum value for that design variable and if X_{i+1} is less than the corresponding design variable than it is set to the minimum value for that design variable.

The inertia weight 'w' plays an important role in the PSO convergence behaviour since it is employed to control the exploration abilities of the swarm. The large inertia weights allow wide velocity updates allowing to globally explore the design space while small inertia weights concentrate the velocity updates to nearby regions of the design space. The optimum use of the inertia weight "w" provides improved performance in a number of applications [13].

The optimum selection of operating parameters of the algorithm like acceleration constants c1 and c_2 as well as inertia coefficient w is very essential for convergence of the algorithm. To ensure the convergence of PSO algorithm, the condition specified by Eq. (3) must be satisfied [13].

$$\max(|\lambda_1|,|\lambda_2|) < 1 \tag{3}$$

Where, λ1 and λ2 are the Eigen values given by Eq. (4) and (5).

$$\lambda_1 = \frac{\left(1 + w - \alpha_1 - \alpha_2 + \gamma\right)}{2} \tag{4}$$

$$\lambda_2 = \frac{\left(1+w-\alpha_1-\alpha_2-\gamma\right)}{2} \tag{5}$$

$$\gamma = \left[\left(1+w-\alpha_1-\alpha_2\right)^2 - 4w\right]^{0.5}, \ \alpha_1 = r_1c_1, \quad \alpha_2 = r_2c_2 \tag{6}$$

The feasible range of the value $\alpha_1+\alpha_2$ is 0-4 and that of w is 0–1.

The above version of PSO algorithm uses the constant value of inertia weight (w) in every generation during the optimization process. In the present work, this version of PSO algorithm is modified by considering the following variable weight,

$$w = (w-0.1)\left(\frac{no.of\ generations - current\ generation}{no.of\ generations}\right)+0.1 \tag{7}$$

3.1 Implementing steps of PSO algorithm

In this study, to fulfil the objective of minimum surface roughness for the given cutting parameters, number of iterations are kept as 100. Other details are as follows.

Step 1: Prepare a swarm with random positions and velocities. No. of particle in swarm is equal to number of variables in optimization problem i.e. each particle is corresponding to one variable. These random particles formed between the allowed ranges of their matching variable.
Step 2: Evaluate the objective function values (i.e. fitness values) of all particles. For every particle set 'pBest' (best position each particle ever meets) and its matching fitness value equal to current position and fitness value of the particle. In the same way set 'gBest' (best global position) and its matching fitness value equal to the best position and fitness value attained by swarm.
Step 3: Update the velocity and position of every particle according to equation (1) and (2).
Step 4: Evaluate the fitness values of all particles for updated position.
Step 5: For each particle, compare its current fitness value with the best fitness value it attained so far ('pBest'). If the current value is better, then update 'pBest' and its fitness value with the current position and fitness value.
Step 6: Regulate the best particle of the current population with the best fitness value. If the fitness value is better than the fitness value of 'gBest', then update it and its fitness value with the position and fitness value of the current best particle.
Step 7: Repeat the method from step II till the termination criteria is met. At termination 'gBest' and its fitness value is the optimum value of the considered problem for single run of the algorithm. The MATLAB software was used for implementation steps of PSO algorithm [13].

3.2 Control parameters of PSO algorithm

Number of particles (population) = 50
Maximum number of generations (iterations) = 100
Variation of inertial weight (w) = 0.9 - 0.4
Cognitive parameter (c1) = 2
Social parameter (c_2) = 2
Maximum velocity (V_{max}) = 1

IV. RESULTS AND DISCUSSIONS

To minimize the surface roughness R_a, by taking into account the control factors such as, cutting speed, feed rate, depth of cut and nose radius is the main aim of this study. The PSO algorithm is considered and compared with the experimental results. The optimum results obtained through this technique are shown in Table 4. The minimum value of surface roughness is achieved through the PSO algorithm is R_a=0.015 µm, which is obtained for 100 iterations with the population size 50. Though the R_a obtained by experimental method is having variation with the R_a calculated through optimization method, for getting optimum results, higher cutting speed, medium feed rate, lower depth of cut and lower nose radius are recommended.

Table 3. Design matrix with responses (Surface roughness)

Design of Experiments				**Measured surface roughness**
Speed	**Feed rate**	**Depth of cut**	**Nose radius**	**R_a(μm)**
1	1	1	1	0.29
1	2	2	2	0.27
1	3	3	3	0.25
1	4	4	4	0.22
1	5	5	5	0.3
2	1	2	3	0.23
2	2	3	4	0.24
2	3	4	5	0.26
2	4	5	1	0.25
2	5	1	2	0.27
3	1	3	5	0.29
3	2	4	1	0.34
3	3	5	2	0.31
3	4	1	3	0.24
3	5	2	4	0.56
4	1	4	2	0.27
4	2	5	3	0.25
4	3	1	4	0.21
4	4	2	5	0.29
4	5	3	1	0.77
5	1	5	4	0.32
5	2	1	5	0.29
5	3	2	1	0.26
5	4	3	2	0.28
5	5	4	3	0.31

Regression equation was generated through Minitab software and the same was applied in optimization algorithm for getting optimized values of cutting parameters.

R_a = 0.586-0.0038 Speed + 0.9 Feed – 2.88 DOC + 0.64 NR + 0.000012 Speed*Speed + 9 Feed * Feed + 1.04 DOC*DOC + 1.450 NR*NR – 0.031 Speed*Feed + 0.0280 Speed*DOC – 0.0933 Speed*NR + Feed*DOC (8)

Table 4. Optimized result of surface roughness for different cutting parameters

Speed	169.8
Feed	0.071
Depth of Cut	0.23
Nose Radius	0.67
Optimized surface roughness, R_a	0.0150728

V. CONCLUSIONS

In this paper, the effort had been made to achieve optimum cutting parameters with the objective function of minimum surface roughness. Different range of cutting speed, feed rate, depth of cut and nose radius were selected for experimentation. Following are some conclusions drawn on the basis of results.

- From the results obtained, it was suggested to take high cutting speed, lower federate, medium depth of cut and low nose radius to accomplish minimum surface roughness value.
- PSO algorithm was found effective in term of getting minimum surface roughness and optimum cutting parameters.
- In terms of mathematical calculations and effecting timing, the PSO responded fast.
- More than one optimization algorithm may be used simultaneously depends upon objective function for getting better results.

REFERENCES

[1] Davim JP, Machining of Hard Materials. London: Springer; 2011.

[2] Khaider Bouacha, Mohamed Athmane Yallese, Samir Khamel, SalimBelhadi, “Analysis and optimization of hard turning operation using cubic boron nitride tool,” Int. Journal of Refractory Metals and Hard Materials, vol. 45, pp. 160–178, 2014.

[3] Dereli D, Filiz IH, Bayakosoglu A, “Optimizing cutting parameters in process planning of prismatic parts by using genetic algorithms,” Int J Prod Res, vol. 39, pp. 3303–28, 2001.

[4] Grynal D’Mello, P. Srinivasa Pai, N.P. Puneet, “Optimization studies in high speed turning of Ti-6Al-4V,” Applied Soft Computing, vol. 51, pp. 105–115, 2017.

[5] R. Venkata Rao, V.D. Kalyankar, “Multi-pass turning process parameter optimization using teaching–learning-based optimization algorithm,” Scientia Iranica E, vol. 20 no.3, pp. 967–974, 2013.

[6] J. Nithyanandam, Sushil Lal Das, K. palnikumar, “Surface roughness analysis of titanium alloy by nano-coated carbide insert,” Procedia Materials Science, vol. 5, pp. 2159 – 2168, 2014.

[7] Liang Gao, Jida Huang, Xinyu Li, “An effective cellular particle swarm optimization for parameters optimization of a multi-pass milling process,” Applied Soft Computing, vol. 12, pp. 3490–3499, 2012.

[8] N. Baskar, P. Asokan R. Saravanan, G. Prabhaharan, “Optimization of Machining Parameters for Milling Operations Using Non-Conventional Methods,” Int J Adv Manuf Technol, vol. 25, pp. 1078–1088, 2005.

[9] Rasool Mokhtari Homami & Alireza Fadaei Tehrani &med Mirzadeh & Behrooz Movahedi & Farhad Azimifar, “Optimization of turning process using artificial intelligence technology,” Int J Adv Manuf Technol, vol. 70, pp. 1205–1217, 2014.

[10] S. J. Kennedy, R. Eberhart, “Particle swarm optimization,” In Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia, pp. 1942-1948, 1995.

[11] Y. Dong, J. Tang, B. Xu, D. Wang, “An application of swarm optimization to nonlinear programming,” Computer & Mathematics with Applications, vol. 49, pp. 1655-1668,2005.

[12] F. Bergh, A.P. Engelbrecht, “A study of particle swarm optimization particle trajectories,” Information Sciences, vol. 176, pp. 937-971, 2006.

[13] R.V. Rao, V. K. Patel, “Thermodynamic optimization of cross flow plate-fin heat exchanger using a particle swarm optimization algorithm,” International Journal of Thermal Sciences, vol. 49, no. 9, pp. 1712-1721, 2010.