Transaction on Evolutionary algorithm and Continuous Optimization
ISSN: 2229-8711 Online Publication, June 2011
www.pcoglobal.com/gjto.htm
GA-O31/GJTO

Global Journal of Technology & Optimization
Volume 2, 2011

OPTIMIZING CNC TURNING PROCESS USING REAL CODED GENETIC ALGORITHM AND DIFFERENTIAL EVOLUTION

Pinkey Chauhan[1], Kusum Deep[1], Millie Pant [2]

[1]Department of Mathematics, [2]Dept. of Paper Technology, Indian Institute of Technology, Roorkee, Uttaranchal, India
Email: kusumfma@iitr.ernet.in, pinkeychauhan030@gmail.com, millifpt@iitr.ernet.in

Received January 2011, Revised May 2011, Accepted June 2011

Abstract
This paper proposes the applications of non-traditional stochastic optimization techniques as real coded genetic algorithm namely "LXPM" and Differential Evolution(DE) for determination of optimal machining conditions for turning process on Computer Numerically Controlled (CNC) machines. The problem, discussed in the present study comprises several nonlinear constraints with an optimum criteria based on minimizing total production time that affects the production rate as well. The various constraints arise due to restricted machining features and are imposed on cutting force, power, tool-chip interface temperature and surface. The determination of optimal cutting parameters has significant importance for economic machining and plays an important role in reducing machining errors as tool breakage, tool wear, tool chatter etc. as well. The performances of employed heuristics are compared with several other optimization methods available in literature. The computational results demonstrate convincingly, the reliability and efficiency of considered methods for predicting optimal machining conditions for achieving the desired goal.

Keywords: CNC Machining, Cutting Speed, Feed Rate, Production time, Constrained Optimization, Real Coded GA (LXPM), Differential Evolution

1. Introduction

In today's competitive and dynamic market environment, mostly large or small manufacturing industries, have assigned a high priority to economic machining due to sensitiveness of machining conditions to production optimization. Being a crucial issue, it has always been a field of interest for researchers and manufacturers to develop an efficient and economic machining. Machining economics has touched new horizon, since the advent of Computer Numerically Controlled (CNC) machines, which revolutionized the manufacturing process by introducing significant changes in manufacturing industry. CNC machines, controls all the processes by computer programs prepared using different programming languages or specific software's designed for machining processes. A CNC machine is mainly designed to meet high accuracies and introduces flexibility in machining processes as well. Some of the common machine tools that can run on the CNC are: Lathe, Milling machines, Drilling Machine etc., used for shaping metal parts by removing inevitable waste material. It has become more extensive with the wide use of CNC machining. High capital investment and operational cost involved in manufacturing encourage researchers to optimize machining parameters for economic yield.

Economic machining operations includes minimization of production cost per piece, minimizing production time and maximizing production rate as an economic criteria seeking optimal solutions of properly formulated models. At the same time several machining constraints must be satisfied, specified by machinists. The selection of optimal machining parameters is traditionally carried out by process planners based on their experience and industrial handbooks. But the parameter values suggested, do not considered economic aspect of machining and also are not appropriate for CNC machining, due to high precision requirement. Therefore an efficient and effective optimization algorithm is desirable to obtain optimal conditions for CNC (Computer numerically controlled) machines.

The optimization analysis of machining processes usually comprises two steps: the first step involves, formulation of a mathematical optimization model, following some economic criteria for machining conditions with various realistic constraints; the second step is to select an appropriate algorithm to seek the optimal or near optimal solutions.

The present study concerns with analyzing the performance of various methods and selecting a suitable one. The objective here is to minimize the total production time, under the restrictions imposed on cutting force, power, tool-chip interface temperature, surface roughness and process parameters. The operating parameters to be determined are cutting speed, feed rate, depth of cut.

The organization of the paper is outlined as follows: Section 2 presents a brief literature review on CNC machining. Section 3 presents a detailed formulation of objective function and constraints along with process parameters of model under consideration. In Section 4 the methodologies, employed to optimize the process parameters of considered CNC turning problem are presented. Experimental results are described and discussed in Section 5 and the paper is concluded in Section 6.

2. Literature Review on CNC Machining

The research for investigating factors, having considerable effect on cutting process and improving its performance has continuously received much attention globally. Since the advent of CNC machining researchers/manufacturers are looking forward for methods that provide high productivity and high speed machining. Consideration of machining parameter optimization started out as early as 1907, when [7] acknowledged the existence of an optimum cutting speed for maximizing material removal rate in single-pass turning operations. Research on machining parameter optimization has increased since the 1950's. In 1950, [24] presented a theoretical analysis of optimization of machining process and proposed an analytical procedure to determine the cutting speed for a single-pass turning operation with fixed feed rate and depth of cut by using two different objectives (i) maximum production rate and (ii) minimum machining cost. As an obvious outcome, the results obtained using these two objectives are always different; therefore in subsequent investigations researchers have used an objective of maximum profit rate that produces a 'compromise' result. In a subsequent study [22] two optimum criteria as total manufacturing cost and total weighted completion time has been considered simultaneously, on a single CNC machine and the resulting problem resembles with that of job sequencing, on decision-making basis. Initially, most of these problems were modeled as unconstrained optimization problems however, with the passage of time, researchers started concentrating on aspects of machining under constrained environment and used different optimization techniques for obtaining optimal parameters [4], [8], [10],[11]. So far, machining optimization problems have been discussed by many researchers using conventional techniques like Geometric programming [2], [4], convex programming [8] and Nelder-Mead simplex search method [10], [11]. Conventional Optimization techniques such as Nelder Mead simplex method and boundary search procedure were used by [18] and a comparison was made with stochastic optimization techniques like Genetic Algorithms and Simulated Annealing that concluded that all methods produces almost equivalent results on quality and feasibility basis. Although single pass turning is much cheaper and safer than multipass turning, but high quality demands led researchers to analyze multipass turning operations using various deterministic or heuristic mehods [17],[23]. A review of artificial intelligence techniques for CNC machining parameter optimization in manufacturing industry was presented by [15] for providing a better understanding of these techniques in machining control. Recently a study on manufacturing of freeform surfaces or sculptured surfaces using CNC machines has been performed in [1], which primarily focuses on three aspects in freeform surface machining: tool path generation, tool orientation identification, and tool geometry selection. A standard optimization technique using genetic algorithm was developed by [19] to solve different machining optimization problems such as turning, face milling and grinding [20]. Recently Particle Swarm Optimization has been applied for CNC turning operation problem in [13] which concluded that PSO performs better than other conventional and non conventional methods.

The present study is mainly focused on optimization of process parameters of CNC turning problem considering minimization of total production time as the objective function with constraints due to cutting force, surface finish, temperature and cutting power. Feed rate and cutting speed are considered as process parameters with specified ranges.

Here we have applied LXPM [12] (a real coded GA) and DE for solving the constrained problem of optimizing the process parameters of CNC turning problem.

3. Machining optimization model

Machining process is basically a manufacturing process for shaping of metal parts by removing unwanted material. During machining, one should satisfy given quality specifications as accuracy, surface finish and surface integrity with an objective of minimum production or machining time. The resulting mathematical models for obtaining optimal combination of machining parameters including cutting speed, feed rate and depth of cut, subject to various constraints are nonlinear and non-convex in nature. In the present study model proposed by [10], [11] is adopted and analyzed.

3.1 Objective function

When developing optimization models, objective functions are determined by optimization criteria. Here in present study, the minimum production time, which is the sum of different processing time such as machining time, tool changing time, quick return time and work piece handling time, is adopted as optimization criteria. The objective function is defined as [11], [12]:

$$T_u = t_m + t_{cs}(t_m / T) + t_r + t_h z$$

Where cutting time per pass is

$$(t_m) = \frac{\pi DL}{1000\, Vf}$$

Taylor's tool life equation is given by

$$V\, f^{a1}\, doc^{a2}\, T^{a3} = K$$

This equation is valid over a region of speed and feed for which tool life (T) is obtained. The notations used in equation (1) are described below.

T_u = Total production time
t_{cs} = Tool changing time (min/edge)
t_r = Quick return time(min/pass)
t_h = Loading and unloading time (min/pass)

3.1 Process parameters

Feed Rate: The maximum allowable feed has a pronounced effect on both the optimum spindle speed and production rate. Feed changes have a more significant impact on tool life than depth of cut changes. The system energy requirement reduces with feed, since the optimum speed becomes lower. Therefore, the largest possible feed consistent with the allowable machine power and surface finish is desirable, in order for a machine to be fully used. It is often possible to obtain much higher metal removal rates without reducing tool life by increasing the feed and decreasing the speed. In general, the maximum feed in a roughing operation is limited by the force that the cutting tool, machine tool, work piece and fixture are able to withstand. The maximum feed in a finish operation is limited by the surface finish requirement and can often be predicted to a certain degree, based on the surface finish ad tool nose radius.

Cutting Speed: Cutting speed is a vital component of tool life equation. When compared with depth of cut and feed, the cutting speed has only a secondary effect on chip breaking, when it varies in the conventional speed range. There are certain combinations of speed, feed and depth of cut which are preferred for easy chip removal which are mainly dependent on the type of tool and work piece material. Charts providing the feasible region for chip breaking as a function of feed versus depth of cut are sometimes available from the tool manufacturers for a specific insert (or) tool, and can be incorporated in the optimization systems.

3.2 Constraints

During machining, some constraints are imposed on machining processes and parameters, which effect the optimal selection of machining conditions and therefore need to be handled carefully while optimizing the machining model.

The parameters depth of cut, feed rate and cutting speed are bounded by upper and lower limits, specified by machinist or tool maker. These bounds are defined as:

(i) $doc_{\min} \le doc \le doc_{\max}$

(ii) $f_{\min} \le f \le f_{\max}$

(iii) $V_{\min} \le V \le V_{\max}$

Other then bounding machining parameters, some constraints are imposed on required machining features. The relative forces as cutting force, thrust force and radial force, in a turning operation are important in the design of machine tools. The machine tool and its components must be able to withstand these forces without causing significant deflections, vibrations, or chatter during the operation. The cutting force constraint is used to prevent the tool chatter and to limit the deflection of work piece/cutting tool that results in dimensional error. The cutting force, F, should be less than the max specified limit F_{max}. The empirical relation for cutting force is given by:

$$F = 844\ V^{-0.10133}\ f^{0.725}\ doc^{0.75} \le F_{\max}$$

During Machining, the cutting power, P, should not exceed the maximum allowable power (P_{max}) of machining tool. The constraint for cutting power is expressed as:

$$P = 0.0373\ V^{0.91}\ f^{0.78}\ doc^{0.75} \le P_{\max}$$

The tool life is difficult to predict and is strongly affected by the chip-tool interface temperature. The hardness and the sharpness of the tool decrease, if the temperature generated at the chip tool interface, exceeds the available limit and the tool cannot be used for cutting anymore. The constraint can be expressed as:

$$74.96\, V^{0.4}\ f^{0.2}\ doc^{0.105} \le \theta_{\max}$$

During finishing, the surface roughness, R, must not be greater than the specified value of surface roughness (R_{max}). The surface roughness constraint follows the following relation

$$14.785\ V^{-1.52}\ f^{1.004}\ doc^{0.25} \le R_{\max}$$

The notations and symbols used in mathematical formulation of considered machining model are defined in Table 1, along with the required data for optimizing machining conditions of considered model.

4. Methodology:

In the present study two non traditional methods; a real coded GA, namely "LXPM" and Differential Evolution(DE), a recently developed population based optimization technique, are considered for determining optimal conditions for CNC turning process. Both the methods are discussed in detail in this section.

4.1 Genetic Algorithms

Genetic Algorithms, introduced by [9], are a family of population based search heuristics, inspired by Darwin's principal of "Survival of fittest". Genetic algorithms are often viewed as function optimizers, although the range of problems to which genetic algorithms have been applied is extremely wide. An implementation of genetic algorithms begins with a population of random solutions called chromosomes, then each chromosome is individually evaluated and reproduction is carried out in such a way that those chromosomes which represents better solution to the target problem are given more chances to "reproduce" than those chromosomes which are poorer solutions. The goodness of a solution is typically defined with respect to the current population.

Genetic Algorithms starts with the encoding of variables, usually binary encoding is used in GA to encode solutions in earlier implementations [5], [9]. The performance of binary GA's are found to be satisfactory on small and moderate size problems, requiring less precision in the solution but in case of high dimensional problems in which higher degree of precision is desired, binary GA's require huge computational time and memory [6]. To overcome the shortcomings of binary coded GA's, real coded GA's came into existence, in which decision variables are encoded as real numbers. It has now been established that real coded GA's are superior to binary coded GA's for continuous optimization problems [3].

4.1.1 LXPM: A real coded GA

LXPM [12] is a real coded GA which uses Laplace Probability Distribution in the crossover phase (the crossover operator is termed as *Laplace crossover*) and uses exponential distribution during the mutation phase, with mutation operator being named as '*power mutation*'. These two operators are defined as follows.

4.1.1.1 Laplace crossover

Let $x^1 = (x_1^1, x_2^1, x_3^1 \ldots x_n^1)$ and $x^2 = (x_1^2, x_2^2, x_3^2, \ldots x_n^2)$ be two parents (known individuals) then the two offspring $y^1 = (y_1^1, y_2^1, y_3^1, \ldots\ldots\ldots, y_n^1)$ and $y^2 = (y_1^2, y_2^2, y_3^2, \ldots\ldots\ldots, y_n^2)$ are generated as follows:

First, uniform random numbers u_i, $r_i \in [0,1]$ are generated. Then a random number β_i is generated satisfying the Laplace distribution, as under:

$$\beta_i = \begin{cases} a - b\log(u_i), & r_i \le 1/2; \\ a + b\log(u_i), & r_i > 1/2, \end{cases}$$

Where a is a location parameter and $b>0$ is scaling parameter. With smaller values of b, offsprings are likely to be produced nearer to parents and for larger values of b, offsprings are expected to be produced far from parents. Having computed β_i, the two offsprings are obtained as under

$$y_i^1 = x_i^1 + \beta_i\left|x_i^1 - x_i^2\right| \qquad y_i^2 = x_i^2 + \beta_i\left|x_i^1 - x_i^2\right|$$

Also one important thing to be noticed here is that difference of above two offspring equations gives:

$$\left|y_i^1 - y_i^2\right| = \left|x_i^1 - x_i^2\right|$$

Which defines the parent centric nature of this operator i.e. spreading of offspring is proportional to the spread of parents.

4.1.1.2 Power mutation

Let $\bar{x}$ be a parent solution then a mutated solution x is created in the following manner. First, a uniform random number $t \in [0, 1]$ and a random number s which follows the power distribution (based on exponential distribution), $s = (s_1)^p$, where s_1 is a uniform random number between 0 and 1, are created, here p is called the index of mutation. It governs the strength of perturbation of power mutation. Having determined s a mutated solution is created as:

$$x = \begin{cases} \bar{x} - s(\bar{x} - x^l), & t < r; \\ \bar{x} + s(x^u - \bar{x}), & t \geq r. \end{cases}$$

Where, $t = \dfrac{\bar{x} - x^l}{x^u - \bar{x}}$ and x^l, x^u being the lower and upper bounds on the value of the decision variable and r a uniformly distributed random number between 0 and 1.

For selection process LXPM uses *Tournament Selection*. The other parameters are set to be as:

a) Population (Np) = $D*10$
b) Generation = *200*
c) Run= *100*
d) Crossover rate (CR) = varies from 0.86 to 0.9
e) Mutation Rate (P_m) = varies from 0.006 to 0.06

Constraint handling: In LXPM parameter free, penalty function approach based on feasibility approach proposed by Deb [14] is used. Fitness value, $fitness\,(X_i)$ of an individual X_i is evaluated using the following relation

$$fitness\,(X_i) = \begin{cases} f(X_i), & if\ X_i\ is\ feasible \\ f_{worst} + \sum_{j=1}^{m} \left|\Phi_j(X_i)\right|, & otherwise \end{cases}$$

Where, f_{worst} is the objective function value of the worst feasible solution currently available in the population. Thus, the fitness of an infeasible solution not only depends on the amount of the constraint violation, but also on the population of solutions at hand. However, the fitness of a feasible solution is always fixed and is equal to its objective function value. $\Phi_j(x_j)$ refers to value of the left hand side of the inequality constraint (equality constraints are transformed into inequality constraints using a tolerance). If there are no feasible solutions in the population also, then f_{worst} is set zero.

The Algorithm for LXPM is given in the box shown:

LXPM Algorithm:

Step 1 (Initialization):
- Initialize population;
- Set Generation=0;

Step 2(Evaluation): Evaluate the fitness for each individual

Step 3(Termination check): Check the termination criteria, set to Maximum number of generations.

If satisfied stop; else goto 4.

Step 4 (GA Operations)
- Select individuals according to Tournament selection to build a mating pool
- Apply Laplace Crossover to the population in mating pool with given crossover probability
- Apply Power Mutation to the current population with given mutation probability

Step 5 (Replacement): Replace the old population with new population while retaining the best individual for next generation

Step 6
- Evaluate the best fitness and find optimal individual
- Increment generation; go to step 3.

4.2 Differential evolution

Differential Evolution (DE) is a recently developed stochastic, population based algorithm, developed by Storn and Price in 1995 [21]. It shares many similarities with other Evolutionary Algorithms (EA) on the basis of operators employed to update population from one generation to next. Like EA, the optimization process in DE is also achieved by applying crossover, mutation and selection operators. However, DE differs from other evolutionary algorithms quite significantly on the working of these operators, particularly the mutation operator. These operators are defined in the following subsections.

4.2.1 Mutation

The mutation operator in DE produces a trial vector corresponding to each individual of the current population by mutating a target vector with a weighted differential. This trial vector is then used by crossover operator to produce offspring. The trial vector $u_i(t)$, corresponding to the target vector $X_i(t)$, is generated, as follows;

Select a target vector X_{i_1}, from the population, such that $i \neq i_1$.

Then, randomly select two individuals, X_{i_2} and X_{i_3} from the population such that $i \neq i_1 \neq i_2 \neq i_3$ and $i_2, i_3 \sim U(1, n_s)$. Using these individuals, the trial vector is calculated by perturbing the target vector as follows:

$$u_i(t) = X_{i_1}(t) + \beta(X_{i_2}(t) - X_{i_3}(t))$$

Where $\beta \in (0, i+)$ is the scale factor which controls the amplification of the differential variation, $(X_{i_2}(t) - X_{i_3}(t))$. The smaller value of β leads to smaller step sizes that increases the computational time of algorithm, on the other hand the larger value of β provides faster convergence but may result in premature convergence. Therefore an appropriate value of β should be chosen to maintain exploration-exploitation trade off.

4.2.2 Crossover

The Crossover operator, combines the trial vector $u_i(t)$ and the parent vector $X_i(t)$, to produce offspring, using the following rule

$$X'_{ji}(t) = \begin{cases} u_{ji}(t) & if\ randb(j) \leq CR\ or\ j = rnbr(i) \\ X_{ji}(t) & if\ randb(j) > CR\ or\ j \neq rnbr(i) \end{cases}$$

Where, $randb(j) \in [0,1]$ is the jth evaluation of random number generator, $rnbr(i)$ is a randomly chosen index $\in \{1,2,\ldots\ldots d\}$, which ensures that offspring, $X'_i(t)$ has at least one component from trial vector $u_i(t)$. CR is the crossover constant to be determined by the user.

4.3.3 Selection

Selection operator decides which individual should be forwarded to next generation, the offspring $X'_i(t)$ is compared to the target vector $X_i(t)$ using the greedy criterion. If the vector $X'_i(t)$ has better fitness value than target vector $X_i(t)$, it will replace the target vector in next generation, otherwise the target vector retains its place for at least one more generation. By comparing each offspring with its target vector from which it inherits parameters, DE strongly integrates recombination and selection in comparison to other EAs:

$$X_i(t+1) = \begin{cases} X'_i(t) & \text{if } f(X'_i(t)) \le f(X_i(t)) \\ X_i(t) & \text{otherwise} \end{cases}$$

Once the new population is installed, the process of mutation, recombination and selection is repeated until the optimum is located, or a pre specified termination criterion is satisfied.
The Algorithm for DE is given as follows:

DE Algorithm

Step 1. Initialization:
- Set the generation number $G = 0$
- Randomly initialize a population of *NP* individuals .

Step 2. DE operations:
WHILE the stopping criterion is not satisfied
Do
For i = 1to *NP* //do sequentially for each individual
Step 2.1 Mutation Step
Generate a donor vector corresponding to the i-th target vector using mutation schemes
Step 2.2 Crossover Step
Generate a trial vector for the i-th target vector through binomial crossover
Step 2.3 Selection Step
Evaluate the trial vector and target vector to compare the best one to move to next generation
End For
Step 2.4 Increase the Generation Count
G = G +1
END WHILE

In the present study, the standard DE version [16], denoted by *DE/rand/1/bin*, is used. The parameter setting of DE for solving the considered CNC model is as follows:

a) Population (Np) = $D*10$
b) Generation =*200*
c) Run= *100*
d) Scaling factor (β) = *0.5*
e) Crossover rate (CR) = *0.8*

Constraint Handling: The constraints are dealt by using penalty method. The penalty parameter is set to 10^3 for each constraint.

5. Computational results and discussion

In this section we discuss the performance evaluation of LXPM and DE *vis-a-vis* other methods such Boundary Search Procedure (BSP), Nelder-Mead Simplex Method (NMS), Binary GA, Simulated Annealing (SA) and Particle Swarm Optimization (PSO). The optimization process using LXPM and DE is performed on a Celeron PC with 1.4 GHz, 1.256 GB RAM implementing in VC++.

Optimal cutting parameters as feed rate, cutting speed and the corresponding total production time, obtained using LXPM and DE for different values of depth of cut are shown in Table 2.

An analysis for LXPM and DE on the basis of different aspects as mean objective function value, standard deviation, average number of function evaluations and average computational time for different values of depth of cut over 100 simulations are quoted in Table 3. Here average function evaluations of successful runs has been calculated, where a run is considered successful if 99% of the obtained global minimum is reached.

The results in Table 3, shows that LXPM performs better than DE on different performance aspects considered.

The performance evaluation of different algorithms on the basis of total production time, taking BSP as reference algorithm is presented in Table 4. From this Table it can be observed that LXPM, PSO and DE perform better than other algorithms on the considered model. Results quoted in Table 4 also predicts that for different values of depth of cut, LXPM gives significant improvement over Binary GA and other methods such as NMS, BSP, GA and SA. On the other hand DE also produces results, better than BSP, NMS, GA, and SA and produces approximately similar results as that of LXPM and PSO.

Fig.1 shows the comparison of performance of all algorithms for different values of depth of cut. From this Fig it can be observed that LXPM, PSO and DE perform better in compared to other algorithms such as NMS, BSP, GA, SA.

Fig.2 illustrates how total production time varies generation wise for LXPM for different values of depth of cut, which shows that for all cases the considered algorithm converges in about 100 generation.

6. Conclusion

Optimizing machining parameters for different processes is intended to improve the machining efficiencies by reducing the cost and time involved in manufacturing processes as per today's economic need. In this study, we dealt with controllable processing times where processing time decisions are taken over the constrained machining environment that affects the machining performance and product quality. A real coded GA called LXPM and Differential Evolution (DE) are employed for optimizing machining parameters of CNC turning process. LXPM uses Laplace crossover and power mutation as evolutionary operators, while DE uses random mutation strategy with binomial crossover operator. The performance of both the algorithms is compared with several other optimization algorithms like PSO, Binary GA, SA, NMS, and BSP. The discussion of results shows that both LXPM and DE performed better than Binary GA, NMS, SA, and BSP. However, if LXPM and DE are compared to each other than it can be observed that LXPM performed better than DE. This is probably due to the fact that in the present study the basic version of DE has been used. Advanced versions of DE may give a better performance.

At the present stage it may be concluded that both LXPM and DE are efficient and reliable algorithms that can be applied to optimize parameters of CNC turning processes for the considered model.

Acknowledgement: The first author acknowledges the financial support provided by the Ministry of Human Resources, New Delhi, India.

References

[1] A. Lasemi, D. Xue, P. Gu, "Recent development in CNC machining of freeform surfaces: A state-of-the-art review", Computer-Aided Design, 42, 641--654(2010).

[2] B. Gopalakrishnan, F. Al-Khayyal, "Machine parameter selection for turning with constraints an analytical approach based on geometric programming", Int. J. Prod. Res. 1897--1908 (1991)

[3] C.Z Janikow, Z. Michalewicz "An experimental comparison of binary and floating point representation in genetic algorithms", In: 4th International Conference on Genetic Algorithms, Morgan Kaufmann, San Francisco, 31--36 (1991).

[4] D.S. Ermer, "Optimization of the constrained machining economics problem by geometric programming", J Eng Ind Trans ASME. 1067--1072 (1971)

[5] D.E. Goldberg, "Genetic algorithm in Search, Optimization and Machine Learning",Addison-Wesley, Reading, MA, New York (1989).

[6] D.E. Goldberg, "Real coded Genetic algorithms, Virtual alphabets, and blocking", Complex Systems, 5 (2), 139--168 (1991)

[7] F.W. Taylor, "On the art of cutting metals", Transactions of the American society of mechanical Engineers, 28, 31--35,(1907)

[8] G. Draghici, C. Paltinea, "Calculation by convex mathematical programming", Int. J. Mach Tool Res. 14 (1974)

[9] Turan Paksoya, Eren Özceylana, Gerhard-Wilhelm Weber, "A Multi Objective Model for Optimization of a Green Supply Chain Network", GJTO, Vol.2, No.1, 2011, pp. 84-96

[10] J.S. Agapiou, "The optimization of machining operations based on a combined criterion. Part 1: The use of combined objectives in single pass operations", J Eng Ind Trans ASME. 114, 500--507 (1992)

[11] J.S. Agapiou,. "The optimization of machining operations based on a combined criterion, Part 2: Multipass operations", J Eng Ind Trans ASME 114, 508--513 (1992).

[12] K. Deep, M. Thakur, "A new mutation operator for real coded genetic algorithms", Appl. Math. Comput..193, 21--230 (2007)

[13] K. Deep, J.C. Bansal, "Performance Analysis of Turning Process via Particle Swarm Optimization", Studies in Computational Intelligence (SCI). 129, 453--460 (2008)

[14] K. Deb, "An efficient constraint Handling Method for Genetic Algorithms", Comput. Methods Appl. Mech. Engrg, 186, 311--338(2000)

[15] K.S. Park and S. H. Kim, "Artificial intelligence approaches to determination of CNC machining parameters in manufacturing: A Review", Artificial Intelligence in Engineering, 12, 121--134 (1998).

[16] K.V. Price, R. M. Storn and J. A. Lampinen, "Differential Evolution: A Practical Approach to Global Optimization", Natural Computing Series, Springer-Verlag Berlin Heidelberg, Germany, (2005).

[17] R. Mesquita, E. Krasteva, S. Doytchinov, "Computer-aided selection of optimum machining parameters in multi-pass turning", Int J Adv Manuf Technol. 10, 19--26 (1995).

[18] R. Saravanan, P. Asokan, M. Sachithanandam, "Comparative Analysis of Conventional and Non-Conventional Optimization Techniques for CNC Turning Process", Int J Adv Manuf Technol. 17, 471--476 (2001)

[19] R. Saravanan, G. Sekar,, M. Sachithanandam, "Optimization of CNC machining operations subject to constraints using genetic algorithm (GA)", In: International Conference on Intelligent Flexible Autonomous Manufacturing Systems, Coimbatore, India, 472--479 (2000)

[20] R. Saravanan, M. Sachithanandam, "Genetic algorithm (GA) for multivariable surface grinding process optimization using a multiobjective function model", Int J Adv Manuf Technol. 17, 330—338 (2001)

[21] R. Storn and K. Price, "Differential Evolution: A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces", Journal of Global Optimization, 11, 341--359(1997)

[22] S. Gurel, M. S. Akturk, "Considering manufacturing cost and scheduling performance on a CNC turning machine", European Journal of Operational Research, 177, 325--343(2007)

[23] S. Satishkumar, P. Askon, S.Kumanan, "Optimization of depth of cut in multi-pass turning using non-traditional optimization techniques", International Journal of Advance Manufacturing Technology, 29, 230--238(2006)

[24] W.W. Gilbert, "Economics of machining, Machining theory and practice", American Society of metals. 465--485 (1950)

Tables and Figures

Table 1: Symbols and notations used in mathematical model

Symbol	*Significance*	*Numerical value*
D	diameter of the workpiece (*mm*)	152
L	length of the workpiece (*mm*)	203
V	cutting speed (*m/min*)	-
V_{min},	minimum allowable cutting speed	30
V_{max}	maximum allowable cutting speed	200
f	feed rate (*mm/rev*)	-
f_{min},	minimum allowable feed rate	0.254
f_{max}	maximum. allowable feed rate	0.762
R_a	surface roughness (μm)	-
R_{max}	max. surface roughness of rough and finished cut	50
P_{max}	max. power of the machine (*kW*)	5
F_{max}	max. cutting force (N)	900
Θ_{max}	max. temperature of tool workpiece interface (°C)	550
doc	depth of cut (*mm*)	-
doc_{min},	minimum allowable depth of cut (finish)	2.0
doc_{max}	maximum. allowable depth of cut (finish)	5.0
T	tool life (*min*)	
t_m	machining time (*min*)	
t_{cs}	tool change time (*min/edge*)	0.5
t_h	loading and unloading time (*min/pass*)	1.5
t_r	quick return time (*min/pass*)	0.13
T_u	total production time (*min*)	-
C_0	operating cost (R_s */piece*)	3.5
C_t	tool cost per cutting edge (R_s */edge*)	17.5
C_T	total production cost (R_s */edge*)	-
a1, a2, a3, K	constants used in tool life equation	0.29 ; 0.35 ; 0.25; 193.3

Table 2: optimal parameter values obtained using LXPM and DE for different values of depth of cut

Depth of cut (d)	LXPM			DE		
	V^*(m/min)	f*(mm/rev)	T_u	V^*(m/min)	f*(mm/rev)	T_u
2.0	139.26	0.762	2.78	139.26	0.761	2.77
2.5	129.07	0.762	2.87	129.07	0.761	2.87
3.0	122.72	0.686	3.06	121.56	0.685	3.06
3.5	122.43	0.585	3.30	122.61	0.585	3.31
4.0	134.54	0.517	3.55	123.53	0.510	3.57
4.5	127.92	0.454	3.82	124.34	0.452	3.83
5.0	132.15	0.410	4.08	125.08	0.405	4.09

Table 3: Mean, standard deviation , average functional evaluations and average computational time over 100 runs for different values of depth of cut using LXPM and DE

Depth of cut (d)	LXPM				DE			
	Mean (obj func value)	Std. deviation	Avg fun Eval	Avg. comput. time	Mean (obj func value)	Std. deviation	Avg fun Eval	Avg. comput. time
2.0	2.780401	4.135e-05	308	0.0453	2.78	0.00147145	801.8	0.0489
2.5	2.8733757	1.40e-04	338	0.0468	2.87276	2.16583e-005	826.2	0.0542
3.0	3.0660640	0.0024726	405	0.0460	3.06573	0.0022352	993.2	0.0558
3.5	3.336070	0.0294902	465	0.0474	3.31947	0.00303094	1000	0.0681
4.0	3.5686617	0.0108513	426	0.0462	3.57587	0.000988971	997.8	0.0586
4.5	3.8364324	0.017543	474	0.1045	3.83784	0.0214352	999.2	0.0599
5.0	4.0990033	0.012372	414	0.0752	4.09841	0.00449684	996.6	0.0574

Table 4: The total production time obtained using different methods with percentage deviation of production time with reference to BSP methods

Algorithm		BSP	NMS		GA		SA		PSO		LXPM		DE	
S.no	doc	T_u	T_u	% dev.	T_u	% dev.	T_u	% dev.	T_u	% dev.	T_u	% dev.	T_u	% dev
1	2.0	2.84	2.87	-1.06	2.85	-0.35	2.85	-0.35	2.78	+2.11	2.78	+2.11	2.77	+2.47
2	2.5	2.93	2.97	-1.37	3.12	-6.48	2.93	0	2.87	+2.05	2.87	+2.05	2.87	+2.05
3	3.0	3.11	3.15	-1.29	3.13	-0.64	3.15	-1.27	3.04	+2.25	3.06	+1.61	3.06	+1.61
4	3.5	3.34	3.44	-2.99	3.46	-3.59	3.34	0	3.29	+1.50	3.30	+ 1.20	3.31	+0.89
5	4.0	3.59	3.69	-2.79	3.51	+2.23	3.59	0	3.55	+1.11	3.55	+ 1.11	3.57	+0.55
6	4.5	3.84	3.88	-1.04	3.96	-3.13	3.85	-0.26	3.82	+0.52	3.82	+0.52	3.83	+0.26
7	5.0	4.10	4.23	-3.17	4.14	-0.98	4.12	-0.49	4.08	+0.49	4.08	+0.49	4.09	+0.24

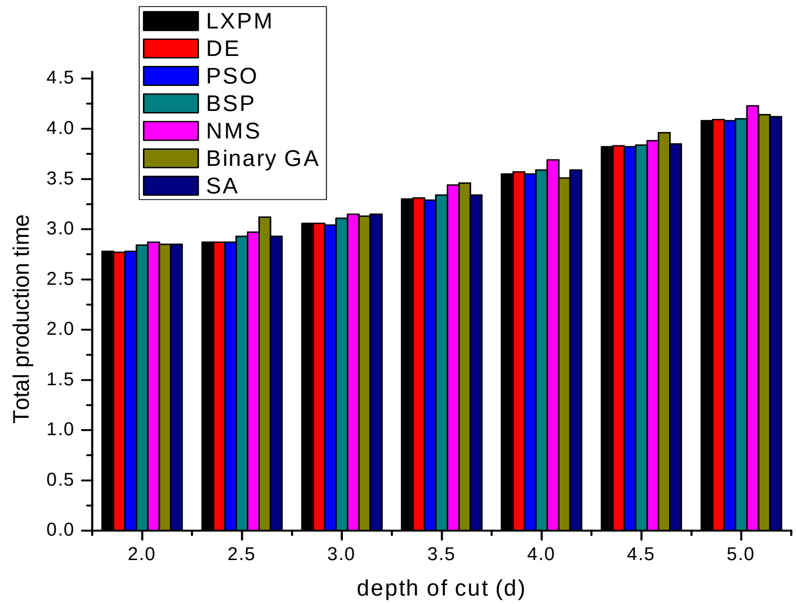

Fig.1: Comparison of different algorithms for minimizing total production time for different values of depth of cut

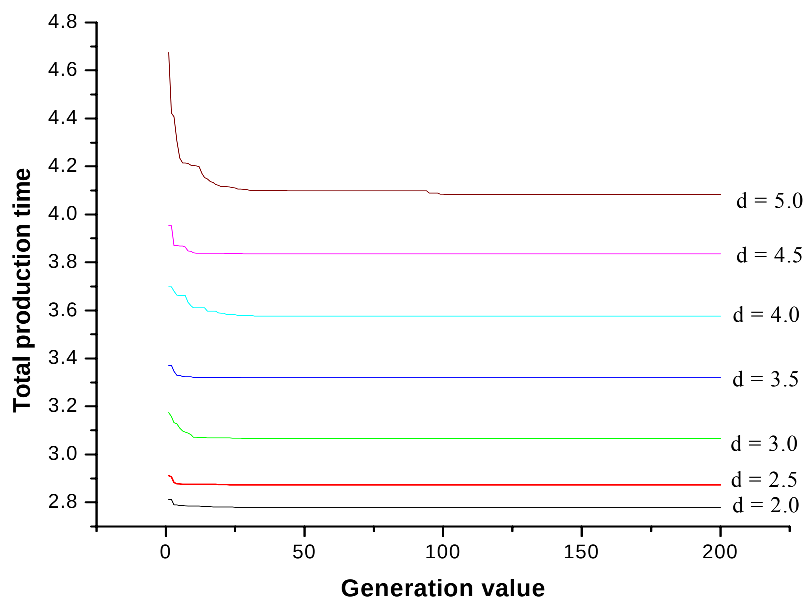

Fig. 2: Convergence graph for LXPM for different values of depth of cut (d)