Machining Parameter Optimization of Poly Tetra Fluoro Ethylene (PTFE) Using Genetic Algorithm

M.Sanjeev kumar[1], V.Kaviarasan[2], R.Venkatesan [3]

1 PG Scholar - Product Design and Development
Sona College of Technology, Salem-5
2 Assistant professor, Department of Mechanical Engineering,
Sona College of Technology, Salem-5
3 professor/Head, Department of Mechanical Engineering
Sona College of Technology, Salem-5

ABSTRACT

Poly Tetra Fluoro Ethylene (PTFE) has emerged as an important class of material in aerospace air-conditioning systems, which are increasingly being utilized in recent years. Application of these materials in many areas is due to light weight, corrosion resistant, etc., Surface Roughness (Ra) is an important value for determining the surface quality. Maximum Surface Roughness (Ra) of the tube reduces the air flow pressure, velocity and volumetric air flow rate in air-conditioning systems. Due to high Surface Roughness (Ra) of the air flow tube, the compressor efficiency is reduced and also increases power consumption. The principal machining parameters that control roughness characteristics are cutting speed, feed rate, depth of cut, and type of cutting tools and temperature etc., Genetic algorithm is used for the optimal search of cutting conditions, the chromosomes represent cutting conditions defined according to a sequential scale and is composed by random keys. The present review is focused on the influence of cutting parameters on the surface finish. This result will provide an insight into selecting the optimum machining parameters for machining of PTFE to achieve minimum Surface Roughness (Ra).

Keywords **- PTFE, Surface Roughness, Speed, Feed, Depth of Cut, Genetic Algorithm**

INTRODUCTION

Roughness plays an important role in determining how a real object will interact with its environment. Rough surfaces usually wear more quickly and have higher friction coefficients than smooth surfaces do. Roughness is often a good predictor of the performance of a mechanical component since irregularities in the surface may form nucleation sites for cracks or corrosion. Poly Tetra Fluoro Ethylene (PTFE) has emerged as an important class of material in aerospace air-conditioning systems, which are increasingly being utilized in recent years. Application of these materials in many areas is due to light weight, corrosion resistant, etc., the term machinability refers to the case with which a material can be machined to an acceptable surface finish. Materials with good machinability require little power to cut, can be cut quickly, easily obtain a good finish, and do not wear the tooling much; such materials are said to be free machining. Machinability can be difficult to predict machining has so many variables. In most cases, the strength and toughness of a material are the primary factors. Strong, tough materials are usually more difficult to machine simply because greater force is required to cut them. Other important factors include the chemical composition, thermal conductivity and microstructure of the material, the cutting tool geometry, and the machining parameters. The machinability can evaluate by different methods. Some of the important methods are Tool life method, Tool forces and power consumption method, Surface finish method and Machinability rating.

In this project, surface roughness is predicted in turning operation using genetic algorithm an optimization technique. Surface finish is an important parameter in this PTFE material because Due to high Surface Roughness (Ra) of the air flow tube, the compressor efficiency is reduced and also increases power consumption. For these reasons, there have been research developments with the objective of optimizing the machining parameters to obtain a good surface finish.

1. Problem Formulation

In machine tools, the finished component is obtained by a number of rough passes and finish passes. The roughing operation is carried out to machine the part to a size that is slightly larger than the desired size, in preparation for the Finishing cut. The finishing cut is called single-pass contour machining, and is machined along the profile contour.

In this paper, during the turning operation carried out in CNC Lathe under variation of the parameters such as speed, feed, depth of cut in order to minimum surface roughness is predicted.

2. Machining Model

The objective of this model is to minimize the surface roughness. The formula for calculating the above surface roughness is as given by,

$$R_a = -0.309 + 0.675V + 0.870f + 0.175d - 0.234V.f - 0.002f.d - 0.143V.d$$

Finally, by using the above mathematical processes, Surface roughness is obtained.

Where,

V = Cutting Speed (m/min)
f = Feed Rate (mm/rev)
d = Depth of Cut (mm)

Machine range:

1. Machine : CNC Lathe
2. Speed range : 150 – 275 m/min
3. Feed range : 0.1 – 0.3 mm/rev
4. Depth of cut : 0.5 – 2.5 mm

Outstanding properties of PTFE:

1. Chemical Inertness
2. Non Stick
3. Low Friction
4. Self Lubricating
5. Dielectric Strength
6. Weather Resistance/Non Ageing
7. Insensitive to UV
8. Non Toxic
9. Broad Temperature Range (-200°C to 260°C)
10. Non Flammable
11. Water Absorption = 0!

3. Genetic Algorithm

Genetic algorithm [4, 6, 7] is an adaptive search and optimization algorithm that mimics the principles of natural genetics. GA's are very different from traditional search and optimization methods used in engineering design problems. Because of their simplicity, easy of operations minimum requirements and global perspective, GA's has been successfully used in a wide variety of problem domains. GA work through three operators, namely reproduction, cross over and mutation. In this paper an attempt is made to use of genetic algorithm to minimize the surface roughness by optimizing the depth of cut, feed rate and cutting speeds.

3.1 Steps in the Genetic Algorithm Method

Step 1: Initialization

Randomly generate an initial population of N chromosomes and evaluate the fitness function to a function to be maximized for the encoded version) for each of the chromosomes.

Step 2: Parent Selection

Set if elitism strategy is not used; otherwise. Select with replacement parents from the full population (including the elitist elements). The parents are selected according to their fitness, with those chromosomes having higher fitness value being selected more often.

Step 3: Crossover

For each pair of parents identified in Step 1, perform crossover on the parents at a randomly (perhaps uniformly) chosen splice point (or points if using multi-point crossover) with probability. If no crossover takes place (probability), then form two offspring that are exact copies (clones) of the two parents.

Step 4: Replacement and Mutation

While retaining the best chromosomes from the previous generation, replace the remaining chromosomes with the current population of offspring from Step 2. For the bit-based implementations, mutate the individual bits with probability ; for real coded implementations, use an alternative form of "small" modification (in either case, one has the option of choosing whether to make the elitist chromosomes candidates for mutation).

Step 5: Fitness and End Test

Compute the fitness values for the new population of N chromosomes. Terminate the algorithm if the stopping criterion is met or if the budget of fitness function evaluations is exhausted; else return to Step 1.

Genetic Algorithm Parameters:

1. Population size: 20
2. Chromosome length: 30
3. Selection mode: rank order
4. Cross over: single-site cross over
5. Probability: 0.08
6. Mutation probability: 0.1
7. Fitness: minimum surface roughness

3.2 Working Principle

1. The decision variables X_i are coded in some string structure, binary coded string having zeros and ones are mostly used.
2. The length of the string is usually determined according to the desired solution accuracy. For example, the strings (0000) and (1111) represent the point ($X_1^{(L)}$, $X_2^{(L)}$) and ($X_1^{(u)}$, $X_2^{(u)}$), the sub string has the minimum and maximum decoded values.
3. The parameter values are calculated by using the following formula,

$$X = X_i^{(L)} + \frac{X_i^{(U)} - X_i^{(L)}}{2^n - 1}\ (\textit{Deocoded value})$$

(Or)

$$X = Min + \left(\frac{Max - Min}{2^n - 1}\right) * (\textit{Decoded value})$$

3.3 Fitness Function [4]

1. Genetic Algorithm mimics the survival of the fittest principle of nature to make search procedure
2. The fitness function F (x) is first derived from the objective function and used in successive genetic operation
3. For minimization problems, the fitness function is an equivalent maximization problem such that the optimum point remains unchanged.

$$F(X) = \frac{1}{1 + G(X)}$$

3.4 Operation of genetic Algorithm

Genetic Algorithm [4, 6, 7] begins with population of random strings representing design and decision variables thereafter each string is evaluated to find the fitness value.

1. The population is operated by three main operators
   a. Reproduction
   b. Crossover
   c. Mutation
2. The population formed is further evaluated and tested for termination. If the termination criteria is not met, the population is iteratively operated by the above three operators and evaluated.
3. This procedure is continued until the termination criteria are met.

3.5 Genetic Algorithm operators [4, 2]

Reproduction

Reproduction selects good strings in a population and forms a mating pool. The reproduction operator is also called a selection operator. In this work rank order selection is used. A lower ranked string will have a lower fitness value or a higher objective function and vice versa. the probability of selection for each string which is calculated, based on the following formula:

Expected value of probability,

$$= \frac{\text{Min} + \square\ (\text{max} - \text{min})\ (\text{rank} - 1)}{N - 1}$$

Where, N = 20
Min = 0.02
Max = 0.08

Crossover

In the crossover operator, exchanging information among strings of the mating pool creates new strings. In most crossover operators, two strings picked from the mating pool at random and some portion of the strings are exchanged between the strings.

Mutation

After a crossover is performed, mutation takes place. This is to prevent falling all solutions in population into a Local optimum of solved problem. Mutation changes randomly the new offspring. For binary encoding we can switch a few randomly chosen bits from 1 to 0 or from 0 to 1.

Mutation can then be following:

Before crossover
00011110110001110 | 1100010111011
01101010110011101 | 0100001000100

After crossover
00011110110001110 | 0100001000100
01101010110011101 | 1100010111011

Table 1. Output of parameter values

S.No	Decoded values			String 1	String 2	String 3	Actual speed	Actual feed	Actual depth of cut
1	321	131	244	0101000001	0101000001	0011110100	189.2229	0.138416	1.096285
2	445	231	123	0110111101	0110111101	0001111011	204.3744	0.167742	0.800587
3	323	421	456	0101000011	0101000011	0111001000	189.4673	0.22346	1.61437
4	123	122	268	0001111011	0001111011	0100001100	165.0293	0.135777	1.154936
5	433	432	287	0110110001	0110110001	0100011111	202.9081	0.226686	1.201369
6	66	95	367	0001000010	0001000010	0101101111	158.0645	0.127859	1.396872
7	499	76	287	0111110011	0111110011	0100011111	210.9726	0.122287	1.201369
8	123	118	187	0001111011	0001111011	0010111011	165.0293	0.134604	0.956989
9	403	95	156	0110010011	0110010011	0010011100	199.2424	0.127859	0.881232
10	196	75	99	0011000100	0011000100	0001100011	173.9492	0.121994	0.741935
11	348	372	271	0101011100	0101011100	0100001111	192.522	0.209091	1.162268
12	460	162	269	0111001100	0111001100	0100001101	206.2072	0.147507	1.15738
13	480	423	276	0111100000	0111100000	0100010100	208.651	0.224047	1.174487
14	82	23	313	0001010010	0001010010	0100111001	160.0196	0.106745	1.264907
15	445	345	319	0110111101	0110111101	0100111111	204.3744	0.201173	1.27957
16	234	323	260	0011101010	0011101010	0100000100	178.5924	0.194721	1.135386
17	456	123	342	0111001000	0111001000	0101010110	205.7185	0.13607	1.335777
18	182	456	499	0010110110	0010110110	0111110011	172.2385	0.233724	1.719453
19	427	234	68	0110101011	0110101011	0001000100	202.175	0.168622	0.666178
20	324	234	198	0101000100	0101000100	0011000110	189.5894	0.168622	0.983871

Table 2. After Cross Over

01111100110001001101 0111001000
01101111010011100011 0101010110
01111000000110100111 0100010100
01101111010101011001 0100111111
01110010000001111111 0001111011
01110011000010100010 0100001101
01111000000110100111 0100010100
01010000110110100100 0100011111
00010000100001011000 0111110011
00011110110001111010 0100001100
01111000000110100111 0100010100
00101101100111001111 0101101111
00011110110001110010 0001000100
01100100110001011011 0101010110
01101010110011101110 0010111011
01010000010010000110 0010111011
01010111000101110100 0100001111
00010000100001011111 0101101111
00011110110001110011 0011110100
01110010000001111111 0010011100

4. GA Procedure[4]

Step1:

Choose a coding to represent problem parameter, a selection operator, a crossover operator and a mutation operator. Choose population size N, crossover probability p_c, and mutation probability p_m. Initialize a random population of strings of size 10. Set iteration t=0.

Step 2: Evaluate each string in the population.

Step 3: If it > it_{max} (or) other termination criteria is satisfied, terminate.

Step 4: Perform reproduction on the population.

Step 5: Perform crossover on the random pairs of strings.

Step 6: Perform bit wise mutation.

Step 7: Evaluate strings in the new population. Set it = it +1 and go to step 3.

Table 3.Output of Genetic Algorithm

S.No	Surface roughness	fitness	Sorted fitness	Rank	Probability of selection	Cumulative probability	Random number	Selected rank
1	91.93534	0.01076	0.009076	1	0.02	0.02	0.237122	7
2	106.5099	0.009301	0.009301	2	0.023158	0.043158	0.055939	2
3	74.41097	0.013261	0.009581	3	0.026316	0.069474	0.515228	13
4	78.90687	0.012515	0.009888	4	0.029474	0.098947	0.678711	15
5	91.43897	0.010818	0.010105	5	0.032632	0.131579	0.783447	17
6	70.43693	0.013998	0.010422	6	0.035789	0.167368	0.491699	12
7	100.1326	0.009888	0.010535	7	0.038947	0.206316	0.561035	13
8	83.58795	0.011822	0.010545	8	0.042105	0.248421	0.083893	3
9	103.376	0.009581	0.010632	9	0.045263	0.293684	0.17981	6
10	93.92137	0.010535	0.01076	10	0.048421	0.342105	0.10199	4
11	88.61061	0.011159	0.010818	11	0.051579	0.393684	0.517456	13
12	97.96541	0.010105	0.010867	12	0.054737	0.448421	0.85709	18
13	94.94815	0.010422	0.011159	13	0.057895	0.506316	0.272369	8
14	75.07653	0.013145	0.011822	14	0.061053	0.567368	0.340149	9
15	91.02521	0.010867	0.011838	15	0.064211	0.631579	0.938477	19
16	83.47469	0.011838	0.012515	16	0.067368	0.698947	0.032898	1
17	93.05703	0.010632	0.013145	17	0.070526	0.769474	0.436371	11
18	64.68515	0.015224	0.013261	18	0.073684	0.843158	0.198944	6
19	109.185	0.009076	0.013998	19	0.076842	0.92	0.266113	8
20	93.82769	0.010545	0.0159	20	0.08	1	0.774445	17

Objective function:

The objective of this model is to minimize the Surface roughness.

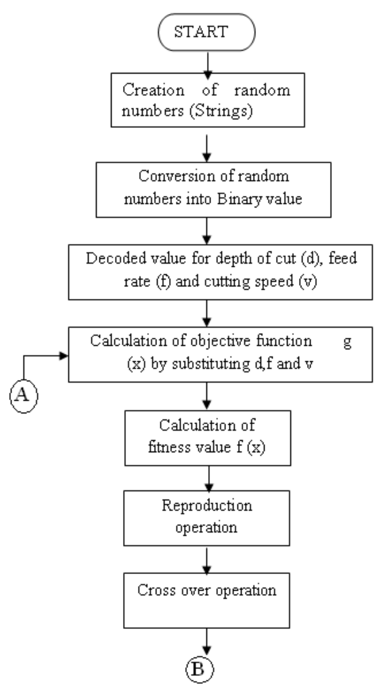
The formula for calculating the Surface roughness is as given by,

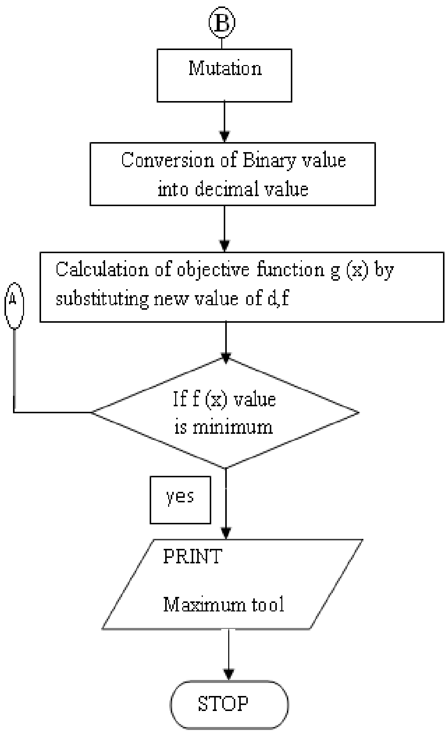
$$R_a = -0.309 + 0.675V_c + 0.870f + 0.175d - 0.234V_c.f - 0.002f.d - 0.143Vcd$$

Finally, by using the above mathematical processes, the Surface roughness is obtained.

Where,	V	= Cutting Speed	(m/min)
	f	= Feed Rate	(mm/rev)
	d	= Depth of Cut	(mm)
	Ra	= Surface roughness	(µm)

Fig 1. Flow chart





Graphical output of genetic algorithm

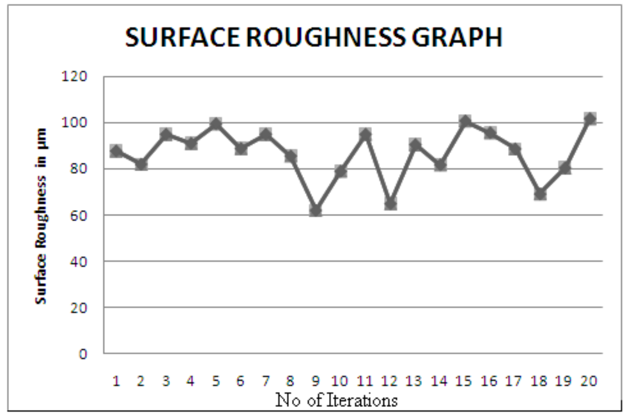


Table 4. Output after iteration

S.No	Decoded values			Children speed (m/min)	Children feed (mm/rev)	Children DOC (mm)	Surface roughness (µm)	Fitness
1	499	77	456	210.9726	0.122581	1.61437	87.73072	0.01127
2	445	227	470	204.3744	0.166569	1.648583	81.92994	0.012058
3	480	423	276	208.651	0.224047	1.174487	94.94815	0.010422
4	445	345	319	204.3744	0.201173	1.27957	91.02521	0.010867
5	456	127	251	205.7185	0.137243	1.113392	99.50479	0.00995
6	460	162	397	206.2072	0.147507	1.470186	88.79617	0.011136
7	480	423	276	208.651	0.224047	1.174487	94.94815	0.010422
8	323	420	287	189.4673	0.223167	1.201369	85.5414	0.011555
9	66	216	499	158.0645	0.164433	1.719453	61.92024	0.015893
10	123	122	268	165.0293	0.135777	1.154936	78.90687	0.012515
11	480	423	276	208.651	0.224047	1.174487	94.94815	0.010422
12	234	323	260	178.5924	0.194721	1.135386	83.47469	0.011838
13	123	114	68	165.0293	0.133431	0.666178	90.44433	0.010936
14	403	91	470	199.2424	0.126686	1.648583	81.70061	0.012092
15	427	238	187	202.175	0.169795	0.956989	100.7736	0.009826
16	321	134	187	189.2229	0.139296	0.956989	95.64201	0.010347
17	348	372	271	192.522	0.209091	1.162268	88.61061	0.011159
18	66	223	367	158.0645	0.165396	1.396872	69.08111	0.014269
19	123	115	244	165.0293	0.133724	1.096285	80.35823	0.012291
20	456	127	220	205.7185	0.137243	1.037634	101.7202	0.009735

5. RESULT AND DISCUSSION

The objective function is the minimization of surface roughness by varying feed, speed, depth of cut. In this work, the optimum surface roughness is obtained by using genetic algorithm at the 9th generation. The optimum value of surface roughness is **61.92024**µm. The corresponding speed is **158.0645** m/min, feed is **0.164433**mm/rev and depth of cut is **1.719453** mm. These are the best parameters obtained to achieve minimum surface roughness in machining PTFE tubes to enhance the airflow in the airplanes and aircraft air-conditioning systems.

6. CONCLUSION

GA's are derivative-free calculations and therefore, are neither bound to assumptions regarding continuity, nor limited by required prerequisites. As Goldberg stated, GAs are blind. They can handle any kind of objective function and any kind of constraints (e.g., linear or nonlinear) defined on discrete, continuous or mixed search spaces. In addition, as stated earlier, they are robust in producing near-optimal solutions, with a high degree of probability to obtain the global Optimum. A genetic algorithm was proposed for optimizing the machining parameters. The main advantage of this approach is that it can be used for any objective function, which was most clearly demonstrated in this example, where the objective function was the minimization of surface roughness. In this approach three constraints namely feed, speed and depth of cut are considered for minimizing surface roughness in PTFE tubes of air-conditioning systems. There are many other constraints that affect surface roughness, which can be solved by using multi objective genetic algorithm in the future.

References:

Journal Papers:

[1] Ramon Quiza Sardinas, Marcelino Rivas Santana and Eleno Alfonso Brindis, Genetic algorithm-based multi-objective optimization of cutting parameters in turning processes, Engineering Applications of Artificial Intelligence, 19, 2006, 127 – 133

[2] R. Saravanan, P. Asokan and K Vijayakumar, Machining Parameters Optimisation for Turning Cylindrical Stock into a Continuous Finished Profile Using Genetic Algorithm (GA) and Simulated Annealing (SA), *International Journal of Advanced Manufacturing technology,* l, 2003, 21:1–9

[3] M. C. Chen and C. T. Su, Optimisation of machining conditions for turning cylindrical stocks into continuous finished profiles, *International Journal of Production Research, 36(8), pp. 2115–2130*, 1998.

[4] Narayanasamy, R., Venkatesan, R. and Ponalagusamy, R., Extrusion Die Profile And Extrusion Pressure Optimization Using Genetic Algorithm, *International Journal of Institution Engineers, Singapore, Vol.45, pp 1-13*, 2005.

[5] B. Gopalakrishnan et al., Machine parameter selection for turning with constraints: an analytical approach based on geometric programming, *International Journal of Production Research, 29(9),pp. 1897–1908*, 1991

Books:

[6] Kalyanmoy Deb, *Optimizations for engineering design – Algorithm and example*s (Prentice-Hall of India, New Delhi, 1996).

[7] S.Rajasekaran,G.A.Vijayalakshmi pai, *Neural Networks, Fuzzy Logic and Genetic Algorithms Synthesis and Applications*(PHI Learning pvt ltd,new delhi, 2003)

[8] D.E. Goldberg, *Genetic Algorithm in search optimization and Machine Learning(* Addison Wesley, MA 1989).

Proceedings Papers:

[9] Maria E. Requena-Pérez, Antonio Albero-Ortiz, Juan Monzó-Cabrera, and Alejandro Díaz-Morcillo, Combined Use of Genetic Algorithms and Gradient Descent Optmization Methods for Accurate Inverse Permittivity Measurement, *IEEE transactions on Microwave Theory and Techniques, vol. 54, no. 2, February 2006.*