Research Paper Engineering

Prediction Of Tool Life In Turning Operation Using Genetic Algorithm

* R. Venkatesan ** V. Kaviarasan *** M. Sanjeev kumar

* Head, Department of Mechanical Engineering, Sona College of Technology, Salem

** Department of Mechanical Engineering, Sona College of Technology, Salem

*** Student, M.E. - Product Design and Development, Sona College of Technology, Salem

ABSTRACT

Surface quality is one of the specified customer requirements for machined parts. There are many parameters that have an effect on surface roughness, but those are difficult to quantify adequately. Surface quality and production cost is mainly depended on tool wear. In finish turning operation many parameters such as cutting speed, feed rate, and depth of cut are known to have a large impact on tool life. In order to enable manufacturers to maximize their gains from utilizing hard turning, an accurate model of the process must be constructed.

The present work deals with Turning Tool life Prediction Using Genetic Algorithm has been established, with depth of cut, cutting speed and feed rate as input parameters and tool life as an output parameter.

Key word : Tool Life, Speed, Feed, Depth of Cut, Genetic Algorithm

Introduction

Machining hardened steel parts has become more pronounce in manufacturing process, particularly in the mold and die industries and subsequently mostly contributed in making automotive and aerospace components. Due to the hardness of the material, abrasive processes such as grinding, polishing, etc. have been typically required, but advances in machine tool and cutting tool materials has allowed machining of hardened steels to become a realistic replacement for many grinding applications [3]. Prediction of Tool Life in turning of Hardened Steel Despite of having outstanding machinery, no one could not expect the failure of tool life for certain conditions in machining operation. It will become most apparent when machining hard materials such as hardened steel. Thus, how to find the best way to prolong the life of a tool subjected to hardened material cutting is the aim of this study. Tool wear/tool life is an important aspect commonly considered in evaluating the performance of a machining process. In addition, tool wear/tool life estimates and the corresponding economic analysis are among the most important topics in process planning and machining optimization. Plus tool life prediction is an important factor that has profound influence on the higher productivity in industrial activities. High metal removal rate is intended to reduce the manufacturing cost and operation time. The productivity in terms of a machining operation and machining cost, as well as quality assurance, and the quality of the work piece machined surface and its integrity are strongly depend on tool wear and consequently it depends on the life of the tool. Moreover, despite having the target of achieving optimum superficial finishing with the shortest possible time one must take

Into Account the consideration of tool life, so that the complete finishing operation can be carried out with just one tool, avoiding the intermediate stops in order to change the tool due to its wear. Eventually, sudden failure of cutting tools lead to loss of productivity, rejection of parts and consequential economic losses

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 lathe how long the tool last under variation of the parameters such as speed, feed, depth of cut in order to achieve maximum tool life.

Machining Model

The objective of this model is to maximize the tool life [4]. The formula [8] for calculating the above tool life is as given by,

$$T = \frac{5.48 X 109}{V3.46 X f0.696 x d0.46\backslash}$$

Finally, by using the above mathematical processes, the tool life is obtained.

Where,

V = Cutting Speed (m/min),

f = Feed Rate (mm/rev),

d = Depth of Cut (mm),

T = Tool Life (min)

Genetic Algorithm

Genetic algorithm [5, 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 maximize the tool life by optimizing the depth of cut, feed rate and cutting speeds.

Solution methodology

- The decision variables X_i are coded in some string structure, binary coded string having zeros and one's are mostly used.
- 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.
- The parameter values are calculated by using the following formula,

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

(Or)

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

Fitness Function

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

$$F(x) = \frac{1}{1 + g(x)}$$

Operation of genetic Algorithm

Genetic Algorithm begins with population of random strings representing design and decision variables thereafter each string is evaluated to find the fitness value, shown in table1.

- The population is operated by three main operators
  - a. Reproduction
  - b. Crossover
  - c. Mutation
- 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.
- This procedure is continued until the termination criteria are met.

Genetic Algorithm operators

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

Table 1: Output of genetic algorithm

DOC	String 1	Feed	String 2	Speed	String 3	Speed	Depth	Feed
450	0111000010	164	0010100100	256	0100000000	207.5268817	2.759531	0.148094
220	0011011100	364	0101101100	121	0001111001	193.0107527	1.860215	0.206745
125	0001111101	483	0111100011	135	0010000111	194.516129	1.488759	0.241642
105	0001101001	206	0011001110	346	0101011010	217.2043011	1.410557	0.160411
154	0010011010	417	0110100001	214	0011010110	203.0107527	1.602151	0.222287
408	0110011000	387	0110000011	154	0010011010	196.5591398	2.595308	0.21349
211	0011010011	443	0110111011	309	0100110101	213.2258065	1.825024	0.229912
374	0101110110	173	0010101101	325	0101000101	214.9462366	2.462366	0.150733
174	0010101110	458	0111001010	411	0110011011	224.1935484	1.680352	0.234311
415	0110011111	166	0010100110	235	0011101011	205.2688172	2.622678	0.14868
479	0111011111	244	0011110100	445	0110111101	227.8494624	2.872923	0.171554
201	0011001001	420	0110100100	98	0001100010	190.5376344	1.785924	0.223167
263	**0100000111**	**138**	**0010001010**	**111**	**0001101111**	**191.9354839**	**2.028348**	**0.140469**
175	0010101111	263	0100000111	265	0100001001	208.4946237	1.684262	0.177126
121	0001111001	122	0001111010	365	0101101101	219.2473118	1.473118	0.135777
398	0110001110	119	0001110111	465	0111010001	230	2.556207	0.134897
109	0001101101	349	0101011101	215	0011010111	203.1182796	1.426197	0.202346
337	0101010001	290	0100100010	313	0100111001	213.655914	2.317693	0.185044
406	0110010110	428	0110101100	498	0111110010	233.5483871	2.587488	0.225513
496	0111110000	327	0101000111	108	0001101100	191.6129032	2.939394	0.195894

which is calculated, based on the following formula:

Expected value of probability,

$$= \frac{Min + (max\ \ min)(rank\,(i,t) - 1)}{N - 1}$$

Where,

N = 30

min = 0.02

max = 0.08

Table 2: Output of reproduction method

Tool Life	Fitness	Sort	Rank	Probability of Selection	Cumulative Probability	Random Number	Selected Rank
124.8031	0.007949	0.00509	1	0.02	0.02	0.237122	8
152.4498	0.006517	0.006451	2	0.023158	0.043158	0.055939	3
147.505	0.006734	0.006517	3	0.026316	0.069474	0.515228	14
137.2948	0.007231	0.006734	4	0.029474	0.098947	0.678711	16
130.3612	0.007613	0.006777	5	0.032632	0.131579	0.783447	18
120.0938	0.008258	0.006788	6	**0.035789**	**0.167368**	**0.491699**	**13**
101.1963	0.009785	0.007231	7	0.038947	0.206316	0.561035	14
115.044	0.008617	0.007296	8	0.042105	0.248421	0.083893	4
87.20974	0.011337	0.007501	9	0.045263	0.293684	0.17981	7
132.3219	0.007501	0.007547	10	0.048421	0.342105	0.10199	5
80.04717	0.012338	0.007613	11	0.051579	0.393684	0.517456	14
154.0081	0.006451	0.007949	12	0.054737	0.448421	0.76709	17
195.4615	**0.00509**	**0.008258**	13	0.057895	0.506316	0.272369	9
136.069	0.007296	0.008617	14	0.061053	0.567368	0.340149	10
146.3226	0.006788	0.009459	15	0.064211	0.631579	0.938477	20
96.6553	0.01024	0.009785	16	0.067368	0.698947	0.032898	2
146.5563	0.006777	0.01024	17	0.070526	0.769474	0.436371	12
104.7172	0.009459	0.011337	18	0.073684	0.843158	0.198944	7
63.74775	0.015445	0.012338	19	0.076842	0.92	0.266113	9
131.5058	0.007547	0.015445	20	0.08	1	0.774445	18

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 shown in table3.

Before crossover	After crossover
00 | 11	00 | 00
11 | 11	11 | 00

Table3: Crossover operation

REPRODUCTION	CROSS OVER
0101110110001010110101 0 1000101	0101110110001010110101 0 0000111
0001111101011110001100 1 0000111	0001111101011110001100 1 1000101
0010101111010000011101 0 0001001	0010101111010000011101 0 1010001
0110001110000111011101 1 1010001	0110001110000111011101 1 0001001
0101010001010010001001 0 0111001	0101010001010010001001 0 1101111
0100000111001000101000 0 1101111	**0100000111001000101000 0 1101111**
0010101111010000011101 0 0001001	0010101111010000011101 0 1011010
0001101001001100111001 0 1011010	0001101001001100111001 0 0001001
0011010011011011101101 0 0110101	0011010011011011101101 0 1010110
0010011010011010000100 1 1010110	0010011010011010000100 1 0110101
0010101111010000011101 0 0001001	0010101111010000011101 0 1010111
0001101101010101110100 1 1010111	0001101101010101110100 1 0001001
0010101110011100101001 1 0011011	0010101110011100101001 1 1101011
0110011111001010011000 1 1101011	0110011111001010011000 1 0011011
0111110000010100011100 0 1101100	0111110000010100011100 0 1111001
0011011100010110110000 0 1111001	0011011100010110110000 0 1101100
0011001001011010010000 0 1100010	0011001001011010010000 0 0110101
0011010011011011101101 0 0110101	0011010011011011101101 0 1100010
0010101110011100101001 1 00110 11	0010101110011100101001 1 0111001
0101010001010010001001 0 0111001	0101010001010010001001 0 0011011

Mutation

- Mutation operator changes one's to zeros and zeros to one's with a mutation probability Pm.
- The widthwise mutation is performed bit by bit by flipping a point with a probability Pm If at any width the outcome is true then the bit is altered otherwise the bit is kept unchanged.
- Mutation probability Pm=0.08
- The need for mutation is to create a point in the neighborhood of the current point. The change after mutation process shown in table4.

There by achieving the local search around the current solution

For example,

0110 -> 0111

1001 -> 1011

Table 4: Mutation operation is carried out

MUTATION
0101110110001010 1101010 1000111
0001111101011110001 1001 0000101
0010101111010000011 1010 0010001
0110001110000111011 1011 1001001
0101010001010010001 0010 0101111
0100000111001000101 0000 0101111
0010101111010000011 1010 0011010
0001101001001100111 0010 1001001
0011010011011011101 1010 0010110
0010011010011010000 1001 1110101
0010101111010000011 1010 0010111
0001101101010101110 1001 1001001
0010101110011100101 0011 0101011
0110011111001010011 0001 1011011
0111110000010100011 1000 0111001
0011011100010110110 0000 0101100
0011001001011010010 0000 1110101
0011010011011011101 1010 0100010
0010101110011100101 0011 1111001
0101010001010010001 0010 1011011

Ga Procedure

Step1:

Choose a coding to represent problem parameter, a selection operator, a crossover operator and a mutation operator.

Choose population size N, crossover probability pc, and mutation probability pm. 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 > itmax (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 5: Output after iteration

DOC child	Feed Child	Speed Child	DOC mm	Feed mm/rev	Speed m/min	Final Tool life min	Fitness
187	173	327	1.7311828	0.160034	215.1613	129.314	0.00767
125	483	133	1.4887586	0.26761	194.3011	137.916	0.0072
175	263	273	1.684262	0.191266	209.3548	127.162	0.0078
398	119	457	2.5562072	0.141295	229.1398	94.8091	0.01044
337	290	303	2.3176931	0.200635	212.5806	100.727	0.00983
263	**138**	**47**	**2.028348**	**0.147889**	**185.0538**	**213.978**	**0.00465**
175	263	282	1.684262	0.191266	210.3226	125.149	0.00793
105	206	329	1.4105572	0.171486	215.3763	134.95	0.00736
211	443	278	1.8250244	0.253729	209.8925	99.7806	0.00992
154	417	245	1.6021505	0.244707	206.3441	115.247	0.0086
175	263	279	1.684262	0.191266	210	125.815	0.00789
109	349	201	1.4261975	0.221109	201.6129	141.377	0.00702
174	458	427	1.6803519	0.258935	225.914	79.227	0.01246
415	166	219	2.6226784	0.157605	203.5484	130.815	0.00759
496	327	57	2.9393939	0.213475	186.129	136.962	0.00725
220	364	44	1.8602151	0.226315	184.7312	166.599	0.00597
201	420	117	1.7859238	0.245748	192.5806	138.798	0.00715
211	443	290	1.8250244	0.253729	211.1828	97.687	0.01013
174	458	505	1.6803519	0.258935	234.3011	69.8389	0.01412
337	290	347	2.3176931	0.200635	217.3118	93.3406	0.0106

Objective function

The objective of this model is to maximize the tool life. The formula for calculating the above tool life is as given by,

$$T = \frac{5.48 \times 10^{9}}{V^{3.46} \times f^{0.696} \times d^{0.46}}$$

Finally, by using the above mathematical processes, the tool life is obtained. Fig.1 shows the graphical output of genetic algorithm.

Where,

V = Cutting Speed (m/min)

f = Feed Rate (mm/rev)

d = Depth of Cut (mm)

T = Tool Life (min)

Graphical output of genetic algorithm

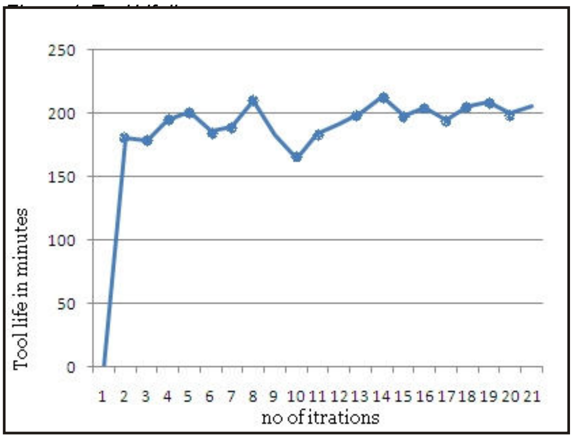

Figure 1 show the tool life obtained in each iteration of GA. The graph shows that the GA produces smooth fitness at the initial iteration and varying tool life in subsequent iterations.

Result And Discussion

The objective function is the maximization of tool life by varying feed, speed, depth of cut. In this work, the optimum tool life is obtained by using genetic algorithm at the 6th generation. The optimum value of tool life is 213.978 minutes. The corresponding speed is 185.0538 m/min, feed is 0.147889 mm/rev and depth of cut is 2.028348 mm refer table 5.The graph no 1 shows that varying tool life in different iterations.

Conclusion

A genetic algorithm was proposed for predicting tool life for a turning tool. 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 maximization of tool life. In this approach three constraints namely feed, speed and depth of cut are considered for maximizing the tool life. There are many other constraints that affect tool life, which can be solved by using multi objective genetic algorithm in the future. 22

REFERENCES

D.E. Goldberg, “Genetic Algorithm in search optimization and Machine Learning”, Addison Wesley, MA 1989. | 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) Int J Adv Manuf Technol (2003) 21:19 | Machining of Hard Materials J. Paulo Davim, PhD Department of Mechanical Engineering, University of Aveiro, Portugal , Springer, London, 2011, ISBN: 978-1-84996-449-4 | Turnad L. Ginta, A.K.M. Nurul Amin, H.C.D. Mohd Radzi, Mohd Amri Lajis, Tool Life Prediction by Response Surface Methodology in End Milling Titanium Alloy Ti-6Al-4V. Using Uncoated WC-Co Inserts European Journal of Scientific Research ISSN 1450-216X Vol.28 No.4 (2009), pp.533-541 | Kalyanmoy Deb, “Optimizations for engineering design Algorithm and examples”, Prentice-Hall of India, New Delhi, 1996. | Ramon Quiza Sardinas*, Marcelino Rivas Santana, Eleno Alfonso Brindis Genetic algorithm-based multi-objective optimization of cutting parameters in turning processes. Published in Engineering Applications of Artificial Intelligence 19 (2006) 127 133 | R. Venkatesan, S.Sridevi ,R.Narayanasamy “Extrusion Die Profile And Extrusion Pressure Optimization Using Genetic Algorithm”, published in the International Journal of Institution Engineers, Singapore, Vol.45, pp 1-13, 2005. | Hajra Choudhary SK and AK, Workshop Technology, Vol. II, Media Promoters & Publishers Pvt. Ltd., Mumbai,1989. | D. S. Ermer, “Optimization of constrained machining economics problem by geometric programming”, ASME Journal of Engineeringfor Industry, 93, pp. 10671072, 1971. | 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. 18971908, 1991 | 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. 21152130, 1998. | D. S. Ermer, “Optimization of constrained machining economics problem by geometric programming”, ASME Journal of Engineering for Industry, 93, pp. 10671072, 1971. | MG Teri Lynn Joslin, 'Tool Life', Volume 6/Issue 8, August 2007 | W. W. Gilbert, “Economics of Machining, Machining Theory and Practice”, American Society of Metals, 1950.