MODELING OF HARD TURNING PROCESSES PARAMETERS BY EVOLUTIONARY TECHNIQUES USING MICRO-GENETIC ALGORITHM

AmaranadhReddy.M *

B. Sasidhar **

ABSTRACT

In this paper we focus on describing the Multi-Objective Optimization Problem considering a case study from real-world production processes like hard turning, for non-ferrous metals and abrasive non-metals, which demands high precession tools. A drawing of experiments has been made in order to determine the empirical non-linear relationship between the selected parameters from the process, the tool life and material removal rate, which are mutually dependent objectives needs to be optimized simultaneously. Micro-Genetic Algorithm procedures have been applied to solve the problem, which are characterized as one of the Evolutionary Algorithm conceptual tools. Further the relationships have been applied to develop the evolution simulation model for adapting the cutting parameters.

Keywords: *Hard turning process, Multi-objective Optimization, Evolutionary Algorithms, Micro-Genetic Algorithms.*

* Associate Professor, Department of Mechanical Engineering, MRCET, Secunderabad, Andhra Pradesh.

** Associate Professor, Department of Computer Science & Engineering, CM Engineering College, Secunderabad, Andhra Pradesh.

1. INTRODUCTION.

Most of the real world problems corresponding to search and optimization naturally involve multiple objectives and may produce trade-offs among different objectives. A solution that is extreme with respect to one objective requires a compromise in other objectives. This restricts in choosing a solution which is optimal with respect to only one objective. In today's competitive manufacturing environment, globally, simultaneously increasing productivity is highly appreciated, while improving and maintaining high quality surfaces in finish turning of hardened parts is required to gain a competent advantage for manufacturers. The hard turning processes demands the use of advanced tools with specially made cutting edges as referred and reported in the earlier literature. Koenig et al. [1] presented the advantages of using hard turning over grinding and its potential to control surface integrity by optimizing tool geometry and machining parameters. Matsumoto et al. [2] addressed the issue of obtaining favorable surface integrity in hard turning processes. It is also evident from a large number of other experimental works that the tool geometry and selected machining parameters have complex relations with cutting forces, tool life, surface roughness and integrity of the finished surfaces. Chou et al. [3] showed the influence of using various cubic boron nitrite (CBN) contents in the cutting tools on tool wear and surface integrity. Thiele et al. [4] have shown that cutting edge geometry has a great influence on the residual stresses induced by hard turning.

Multi-Objective Optimization (MOO) and Evolutionary Algorithms (EAs) were discussed with working illustrations like A concrete engineering design problem, A Car buying decision making problem (Quality & Cost) etc. also rise of Evolutionary Multi-Objective Optimization (EMOO) and innovations are introduced Deb.KalyanMoy in his the literature works.

2. MULTI-OBJECTIVE OPTIMIZATION.

Multi-objective optimization (MOO) is concerned with the simultaneous achievement of multiple objective functions that are subjected to a set of constraints.

Max./ Min. $f_m(x)$, m=1,2,3…M

Subject to $g_j(x) \geq 0$ j=1,2,3……J (Inequality constraints)

$h_k(x) = 0$ k=1,2,3……K (equality constraints)

$x_i^{(L)} \leq x_i \leq x_i^{(U)}$ i= 1,2,3……n (restraining decision variable)

Since the conventional techniques can deal with only one objective function, for solving MOO Problems EA techniques like Genetic Algorithms and Neural Networks are employed. EA's are stochastic search methods that mimic the metaphor of natural biological evolution and operate on a population of potential solutions applying the principle of survival of the fittest to produce better and better approximations to a solution. At each generation, a new set of approximations is created by the process of selecting individuals according to their level of fitness in the problem domain and breeding them together using natural genetic operators such as Selection, Recombination, Mutation, Migration, Locality and Neighborhood.

Advantages of EAs are

- The solution space is from a population of points in parallel, not just a single point.
- Do not require derivative information or other auxiliary knowledge; only the objective function and corresponding fitness levels influence the directions of search.
- Uses probabilistic transition rules, not deterministic ones.
- Generally more straightforward to apply, because no restrictions for the definition of the objective function exist.
- Provides a number of potential solutions to a given problem. The final choice is left to the user.

3. GENETIC ALGORITHM (GA).

GA is one of the meta-heuristic algorithms, which was introduced by Holland (1975). GA is a class of iterative procedures that simulate the evolution process of a population of structures subject to competitive forces prescribed in Darwin's 'survival of the fittest' principle. The process of evolution is random yet guided by a selection mechanism based on the fitness of individual structures. If carefully designed and properly implemented, a GA will exhibit behavior similar to that described in Darwin's evolution theory, and relatively high fitness structures have a greater chance to survive and to produce even higher fitness offspring. The result will be an increase in the overall fitness of a population in each new generation. Genetic algorithms are different from the traditional optimization and search techniques. Goldberg (1989) concluded the differences as:

1. GAs work with a coding of the parameter set, not the parameters themselves.
2. GAs search from a population of points, not a single point.
3. GAs use information of fitness function, not derivatives or other auxiliary knowledge.
4. GAs use probabilistic transition rules, not deterministic rules.

The GA paradigm has been proposed to solve a wide range of problems [6, 7, 8]. GA has been successfully applied to optimization problems in diverse fields and it is differs from other search techniques which depend on natural genetic evaluation process [15, 16]. GA starts with an initial set of solutions selected randomly called population. A suitable encoding for each solution in the population is used to allow computation of the fitness. The solution set in the population, called as chromosome or individual, represents a solution to the optimization problem. Each individual contains a number of genes. The individuals in the initial population are evaluated to measure its fitness. To create the next population, new individuals are formed by either merging two individuals from the current population using a crossover operator or modifying an individual solution using mutation operator. Based on the individuals' fitness, the individuals to be included in the next population are then probabilistically selected from the set of individuals in current population. The iteration, called a generation is continued until the fitness reaches its maximum value, with the hope that strong parent will create a fitter generation of the children. The best overall solution becomes the candidate solution to the problem. To create the next generation GA based on three operations: Selection, crossover and mutation. Out of many GA

3.1 Micro-GA.

Micro-genetic algorithms are similar to the standard genetic algorithm in sharing the same evolution parameters and similar considerations. The important distinction is, since new genetic material is introduced into the population every time the algorithm is restarted, there is really no need for either jump or creep mutation. Also, elitism is required, at least every time the population is restarted, and otherwise the algorithm would lose its exploitation capability. High dimensionality problem demands more precession in the process time consuming for all the model parameters to converge within a given margin of error. In particular, as the number of model parameters increases, so does the required population size. Recall that large population sizes imply large numbers of cost-function evaluations. An alternative is the use of micro-genetic algorithms [12], which evolve very small populations that are very efficient in locating promising areas of the search space. Obviously, the small populations are unable to maintain diversity for many generations, but the population can be restarted whenever diversity is lost, keeping only the very best fit individuals (usually we keep just the best one, that is, elitism of one individual). Restarting the population several times during the run of the genetic algorithm has the added benefit of preventing premature convergence due to the presence of a particularly fit individual, which poses the risk of preventing further exploration of the search space and so may make the program converge to

a local minimum. Also, since we are not evolving large populations, convergence can be achieved more quickly and less memory is required to store the population.

First, a random population is generated. This random population feeds the population memory, and this is divided in two parts: a replaceable and a non-replaceable portion. The non-replaceable portion of the population memory will never change during the entire run and is meant to provide the required diversity for the algorithm. On the other hand, the replaceable portion will experience changes after each cycle of the micro-GA. The population of the micro-GA at the beginning of each of its cycles is taken (with a certain probability) from both portions of the population memory so that a mixture of randomly generated individuals (non-replaceable portion) and evolved individuals (replaceable portion). During each cycle, the micro-GA undergoes conventional genetic operators (binary representation is used in our implementation): tournament selection, two-point crossover, uniform mutation and elitism (only one non-dominated vector is arbitrarily selected at each generation and copied intact to the following one). After the micro-GA finishes one cycle, we choose two non-dominated vectors from the final population and compare them with the contents of the external memory (this memory is initially empty). If either of them (or both) remains as non-dominated after comparing it against the vectors in this external memory, then they are included there (i.e., in the external memory).

4. APPLICATION ON PRODUCTION PROCESS.

The present work is focused on to formulate a procedure and solve optimization problem for MOO that exists in advanced Production processes like Hard turning, which demands the use of precession tools with a specially prepared cutting edges. The non- linear relations between the machining parameters including the tool geometry and the performance measure of interest can be obtained by EAs using the experimental data. As mentioned earlier, wear behavior of CBN tools are different than conventional tools; therefore, the tool life expressions developed for conventional tools does not apply to CBN tools [3]. In order to be competitive in today's market conditions, manufacturers should find the most economical way to manufacture their product, which can be possible by selecting the optimum machining conditions.

In this work, objective functions will be treated separately and the optimal cutting conditions will be calculated by GAs to solve optimization problems with multiple objective functions. Hence, the objective of this study is to obtain a group of optimal process parameters which maximize tool life while maximizing material removal rate. It is widely reported that the GA

is very easy to implement and has fewer parameters to adjust when compared to other evolutionary algorithms. The information sharing mechanism among chromosomes in GAs is significantly different from other. In GAs, the entire group moves toward an optimal solution area and facilitates to opt for the best solution and then tends to converge to the best solution.

4.1. Maximizing tool life and maximizing material removal rate.

In finish hard turning, Cubic Boron Nitride (CBN) cutting tools is highly dependent on cutting conditions such as cutting speed, feed, feed-rate, and depth of cut. Also Cutting speed and depth of cut have a particularly significant influence on tool life.

Selected optimal machining parameters are presented in the following equation

$$T = \frac{A}{V^3 + B + V^2 + C + V}$$

Where

T : Tool life

V : Cutting speed

A, B, C : Coefficients calculated under different cutting conditions

The earlier work by Mamalis et al. [5], a new tool life equation for CBN tools when machining 100Cr6 steel (62 HRC) has been proposed. This proposed tool life equation, which is valid in the whole cutting speed range for a given feed rate and depth of cut, has two extreme values. The coefficients A, B, and C of the equation were calculated under different cutting conditions and the results are in good agreement with experimental values [5]. It was also shown that different feed rate and depth of cut values changes the location of the two extreme values of the tool life curve.

f_1= max. (tool life)

f_2= max. (Material removal rate = V x f x d)

Subject to:

10m/min ≤ V ≤ 120 m/min; (Cutting speed)

0.025mm/rev ≤ f ≤ 0.125 mm/rev; (Feed rate)

0.05mm ≤ d ≤ 0.25 mm (Depth of cut)

5. CONCLUSIONS:

In this paper, we introduce a procedure to formulate and solve optimization problems for multiple and conflicting objectives that may exist in finish hard turning processes using Micro-GA modeling with. The representative multiple objectives for hard turning are defined to obtain a optimal process parameters, which maximizes both tool life and material removal

rate. The EMOO models integrated with the GA is proposed in order to obtain a family of solutions that provides useful information to the user during the selection of machining parameters. Therefore, Pareto optimal fronts are computed and presented with stem plots for representing machining parameters yielding to a certain merit of interest such as material removal rate, surface roughness, tool life etc. that can be selected by the user according to production requirements. The proposed approach is applied and its effectiveness is demonstrated. The results indicate that the Micro-GA approach for solving the MOO problem with conflicting objectives is both effective and efficient, and can provide intelligence in production planning for multi-parameter turning processes.

Feed rate	Depth of cut	Cutting speed	Tool life	MRR
(mm/ rev)	(mm)	(m/min.)	(min)	(mm^3/min)
0.025224	0.10723	10	350.4	0.027047695
0.025975	0.11823	11.1	349.8	0.034088369
0.034957	0.11823	12.754	299.2	0.05271185
0.040953	0.21028	49.333	216.47	0.424835907
0.072226	0.24096	40.831	194.85	0.710605451
0.095481	0.24985	40.396	178.26	0.963684061
0.102131	0.25003	41.452	179.61	1.058510559
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
0.125	0.25	120	5.5001	3.75

6. REFERENCES:

1. Koenig WA, Komanduri R, Toenshoff HK, Ackeshott G (1984) Machining of hard metals. CIRP Ann 33(2):417–427
2. Matsumoto Y, Hashimoto F, Lahoti G (1999) Surface integrity generated by precision hard turning. CIRP Ann 48(1):59–62
3. Chou YK, Evans CJ, Barash MM (2002) Experimental investigation on CBN turning of hardened AISI 52100 steel. J Mater Process Technol 124:274–283
4. Thiele JD, Melkote SN, Peascoe RA, Watkins TR (2000) Effect of cutting-edge geometry and workpiece hardness on surface residual stresses in finish hard turning of AISI 52100 steel. ASME J Manuf Sci Eng 122:642–649

5. Mamalis AG, Kundrak J, Horvath M (2005) On a novel tool life relation for precision cutting tools. J Manuf Sci Eng 127:328–332
6. S. Forrest, “Genetic Algorithms”, ACM Computing Surveys, Vol. 28 No. 1, 1996, pp. 77-83.
7. D. Lawrence, Handbook of Genetic Algorithms, Van No strand Reinhold, New York, 1991.
8. D. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, New York, 1989.
9. Goldberg, D. E., and Richardson, J., "Genetic Algorithms with Sharing for Multimodal Function Optimization," Genetic Algorithms and their Applications: Proceedings of the Second International Conference on Genetic Algorithms, 1987, pp. 41-49.
10. Goldberg, D. E., "A Note on Boltzmann Tournament Selection for Genetic Algorithms and Population-Oriented Simulated Annealing," in: Complex Systems, Vol. 4, Complex Systems Publications, Inc., 1990, pp. 445-460.
11. Goldberg, D. E., "Real-coded Genetic Algorithms, Virtual Alphabets, and Blocking," in: Complex Systems, Vol. 5, Complex Systems Publications, Inc., 1991, pp. 139-167.
12. Goldberg, D. E., and Deb, K., "A Comparitive Analysis of Selection Schemes Used in Genetic Algorithms," in: Foundations of Genetic Algorithms, ed. by Rawlins, G.J.E., Morgan Kaufmann Publishers, San Mateo, CA, pp. 69-93, 1991.
13. Goldberg, D. E., Deb, K., and Clark, J. H., "Genetic Algorithms, Noise, and the Sizing of Populations," in: Complex Systems, Vol. 6, Complex Systems Pub., Inc., 1992, pp. 333-362.
14. Krishnakumar, K., "Micro-Genetic Algorithms for Stationary and Non-Stationary Function Optimization," SPIE: Intelligent Control and Adaptive Systems, Vol. 1196, Philadelphia, PA, 1989.
15. Syswerda, G., "Uniform Crossover in Genetic Algorithms," in: Proceedings of the Third International Conference on Genetic Algorithms, Schaffer, J. (Ed.), Morgan Kaufmann Publishers, Los Altos, CA, pp. 2-9, 1989.
16. Beasley, D., Bull, D., and Martin, R., 1993, An overview of genetic algorithms: Part 2, research topics: University Computing, **15**, no. 4, 170–181.
17. Gen, M., and Cheng, R., 2000, Genetic algorithms and engineering optimization: John Wyley and sons.
18. Gill, P., and Murray, W., 1981, Practical optimization: Academic Press.