Implementation of Genetic Algorithm in Multi-Objective Optimization of Milling Toolpath

Munish Kumar, Dr. Pankaj Khatak

Department of Mechanical Engineering, GJUS&T, Hisar 125001, India.

Abstract

Genetic algorithm is a part of metaheuristic techniques based on the principles of evolution and survival of the fittest. GA's have been extensively used in optimization in a variety of computing domains including manufacturing and industrial realms. In this paper, the GA is implemented on multi-objective optimization of CNC milling operations, particularly in pocket machining. The milling area is discretized approximated using a square grid where a sequence of grid squares constitutes a milling toolpath. This paper discusses the modeling and implementation of various GA operators such as selection, crossover, mutation, etc. using a suitable fitness function.

Keywords: CNC milling, genetic algorithm, crossover, mutation, fitness.

1. Introduction

From the first machine tool to recent assembly-line robots, technology has played an essential role in production and manufacturing sector. Continuous advancements in technology has enabled a manufacturing industry to produce beyond its previous limits hence increasing its productivity and quality along with a positive impact on overall manufacturing and energy cost [1]– [4]. Nowadays, integration of a computer can be seen in almost every part of production whether it be material handling, quality control, packaging and export, testing & statistical analysis, and above all, manufacturing; where automated machine tools have transformed the process.

There are a number of elements of automation which dictate the degree of automation in any industries. These elements are: computer numerical control (CNC), direct numerical control (DNC), CAD/CAM, flexible manufacturing systems (FMS), automated material handling and retrieval system, assembly lines, industrial robots, etc. Among these, CNC technology is a major element in automation of any manufacturing industry. It dominates other elements of automation as most of them serves the sole purpose to enhance CNC capabilities and make CNC machine tools more productive.

Nomenclature

d = distance between two consecutive squares
d_s = distance of first square from tool starting point
d_f = distance of last square from tool end point
c = collinearity coefficient
t = tool-change coefficient
n = total number of squares in a sequence
n_t = total number of cutting tools used
i, j, k = squares in a sequence
D = cutting distance (non-dimensional form)
P = tool parking distance (non-dimensional form)
C = degree of collinearity of a sequence
T = degree of tool-change operations of sequence
f = fitness score
ρ = standard dimension of workpiece
τ = cutting tool used for machining a particular square
ϕ = penalty factor

For every manufacturing industry, production rate and quality are two important aspects of production. They continuously strive to enhance both the factors by introducing new techniques, processes and even machines. An ideal manufacturing process is the prime objective of the industries. It is generally characterized by combination of sets of ideal manufacturing variables such as machining parameters, process plans and toolpaths. A so-called 'near-ideal' or 'optimal' process can lead to significant increases in the levels of production rates and quality of manufacture. On the other hand, use of non-optimal data leads to huge limitations in the process and prove to be a serious liability to an industry. Selection of vague parameters can cause excessive tool wear, surface roughness, less material removal rate and high manufacturing costs and non-optimal toolpaths result in high tool travel, high machining time, and reduced tool life.

In literature, optimum toolpath planning is traditionally regarded as a "travelling salesman problem" (TSP) [11]. Main objective of this problem lies is finding a suitable for tool such that required material is removed. But this classical problem has huge complexities associated with it. Also, TSP is found to have large search spaces which is very difficult to solve. Researchers have tried a number of methods to solve TSP, such as nearest neighbour, cutting planes, branch and bound, to name a few [11]. Due to complexity of problem, a new strategy to obtain good results is using evolutionary approach. Mainly, genetic algorithms (GA) have been extensively used in literature to solve such toolpath optimization problems.

One such strategy was presented by A. Krimpenis and G.C. Vosniakos. They proposed an optimization technique based on GA to optimize toolpaths for roughing operations on sculptured surfaces. Machining time was considered to be minimized for the goal of optimization [12]. J.C Chen and T.X. Zhong proposed a hybrid genetic algorithm for the solution of such travelling salesman problem [11]. The so-called hybrid-coded genetic algorithm (HCGA) was used to optimize non-productive paths in CNC contour machining such as laser engraving and flame cutting. M. Lee and K. Kwon measured the performance of a proposed toolpath optimization based on genetic algorithms by relative effectiveness. The proposed method was similar to methods discussed above [14].

Genetic algorithms are an important and the most explored part of evolutionary algorithms introduced by J. Holland in 1975 [7]. Almost every field of engineering, mathematics, computer technology has been using genetic algorithms for a long time since its adaptation. The reason is the ability of GAs to solve a variety of simple and fairly complex problems. It possesses distinctive attributes from other conventional methods, for instance, it is not based on modelling data and its simulation. A data set in its simplest form is coded in an encoding scheme on which GA functions.

In early GAs, data was represented as a binary string consisting of ones and zeros only, known as binary representation of GA. Each data string was called a chromosome, an organism or an individual, all used interchangeably and a group of these is called a population which evolve through generations. The binary digits are called genes of that chromosome or individual and the value assigned to a gene is known as an allele. Two other important terms are genotype or genome and phenotype or phenome. A genotype is the characteristics inhibited by an individual, genes and alleles, their ordering and sequencing. A phenotype is the physical representation of the arrangement of genes, i.e. an individual's features exhibited in physical world. For example, in humans, structure and arrangements of genes related to eyes and ears are his genotype while colour of his eyes and shape of ears is the corresponding phenotype. The processes of variety generation such as crossover and mutation were applied on these strings to produce offspring. While in crossover two parents are required to produce an offspring, mutation require only one. These two GA operators enable the algorithm to search in a vast space of solutions for better and fitter ones. The so-called blind search of GAs [8] is directed to either direction by use of a fitness function. This is a typical function which determines the fitness of an individual or chromosome and is a representation of characteristics of a standard solution for given problem.

In general, evolutionary algorithms and particularly GAs have been implemented successfully in two major fields of research – optimization and study of complex systems and their adaptation. In computer numerical control (CNC), as discussed in previous sections there is a constant need to optimize machining environment for better accuracy, productivity and efficiency. There have been a number of implementations of evolutionary approach to optimize machining parameters, toolpaths, process planning and error control with the aim to maximize production rate, productivity, and minimize production cost, machining time, & surface roughness.

Most of the times a CNC optimization problem constitutes a multi-objective optimization problem which are solved through conventional means for long times with non-optimal results. Such multiobjective problems cannot be solved using conventional approaches as they are not designed keeping multiple objectives & solutions in mind [9]. However, evolutionary approaches tend to be well suited for multiobjective problems consisting of multiple variables & criteria [10]. Here, multiple individuals can search for multiple solutions to a problem in parallel in a search space. Hence it can prove to be an effective search & optimization tool when applied to machining environment.

In this paper, genetic algorithm is implemented for toolpath optimization. First, the machining area is discretized into finite squares and modelled in GA domain with a suitable representation scheme, various GA operators. Then, the optimization problem is discussed and fitness function for GA is established. The performance of GA and effect of various parameters are also discussed.

2. Modeling of machining area

In a typical machining process, a cutting tool is required to follow a sequence to machine given part. The part may consist of a number of design features such as pockets, contours, or holes. A tool-path is generated which covers all these design features at least once inside the machining environment. Different locations of machinable area are visited by a cutter in sequence for successful machining of part/component. Therefore, a cutting strategy or toolpath can be defined as the sequence of cutter locations of a design feature inside machinable area. In the present study, the machinable area is divided into a grid of finite squares, a strategy also known as discretization.

As shown in Fig. 1, the square grid covers the entire machining area. The squares marked in red color represent a particular

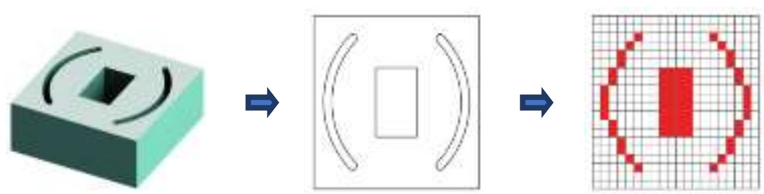


Fig. 1. Process of machining area discretization.

design feature and known as active squares while other white squares are inactive squares. A sequence of active squares constitutes a cutting tool path. Following constraints are posed on a valid sequence of squares: -

1. Every active square or R-active square must be included in a sequence and must not be visited more than once by a cutting tool.
2. Every square corresponding to one design element must be visited before shifting to square of other design element.
3. A design element is cut by specified cutting tool only and every given tool must be used in the process at least once.

3. Optimization problem

As discussed in previous section, a tool path constitutes of sequence of squares to machine the entire area. Depending upon the number of points or squares, there can be a number of such sequences constructing a toolpath. An optimal toolpath is selected from these toolpaths. The optimization problem is hereby formulated with respect to some aspects of machining process such as cutting distance, tool change time, jerk etc.

4.1 Minimization of cutting distance

A cutting distance is taken as the summation of distance between consecutive squares in a sequence. The objective function deals with minimization of the cutting distance *D* (non-dimensional form).

$$D = Min\sum_{i=1}^{n-1}\sum_{j=2}^{n}\frac{d_{ij}}{\rho}$$

4.2 Minimization of tool parking distance

Tool parking distance considered in this problem is the distance travelled by tool before cutting starts and after cutting is completed. This is taken as the sum of distance from tool starting point to the first square and tool final point to the last square of a sequence. The objective function minimizes the tool parking distance (non-dimensional form).

$$P = Min\sum\frac{ds + df}{\rho}$$

4.3 Minimization of jerk

To minimize the effects of jerk, a toolpath favoring fewer turns and keeping the tool moving in straight direction for longest possible durations is given priority. It can be ensured by checking the degree of collinearity in a sequence considering 3 squares at a time upto n.

$$c_{ijk} = \begin{cases}1, & i, j, k \text{ are collinear} \\ 0, & i, j, k \text{ are noncollinear}\end{cases}$$

$$C = Min\sum_{i=1}^{n-2}\sum_{j=2}^{n-1}\sum_{k=3}^{n}\frac{1}{c_{ijk}}$$

4.4 Minimization of tool change time

Frequent change of tools during a cutting operation leads to higher machining times. The goal of this objective function is to reduce number of un-necessary tool change operations and hence to lower the machining times.

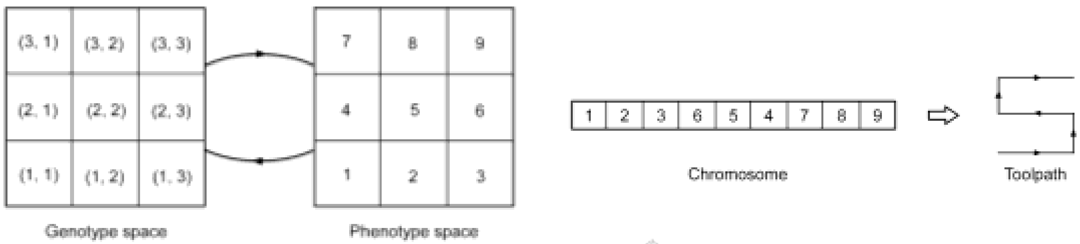
$$t_{ij} = \begin{cases}1, & \tau_i = \tau_j \\ 0, & \tau_i \neq \tau_j\end{cases}$$

$$T = Min\sum_{i=1}^{n-1}\sum_{j=2}^{n}\frac{t_{ij} + 1}{n_t}$$

4. GA representation scheme

As the optimization problem being dealt in this research is related to a sequencing problem. A sequencing of squares is sought to generate a toolpath with optimal characteristics. Keeping this in mind, permutation encoding is selected. The sequence of squares is represented through numbers as shown in Fig. 2(a). The permutation encoding scheme is best suited for such sequencing problems. It possesses the characteristics such as locality and heritability [8]. It is worth mentioning here that this sequencing of squares does not affect the optimization process by any means. An example of chromosome is shown in Fig. 2(b).

5. Diversity generation using GA operators



a. b.

Fig. 2. Permutation encoding scheme. (a) Genotype and phenotype mapping. (b) Example chromosome and corresponding toolpath.

The process of generating new offspring aims to obtain information from better solutions in the current populations and using it to search for even better areas of the search space. In this regard, selection and crossover tend to promote exploitation whereas mutation tends to promote exploration. Choices for the selection strategy, the design of mutation and crossover operators determine balance between exploitation and exploration [8].

5.1 Selection

The selection usually comprises of choice of the individuals from current population to be parents for the upcoming generation. The selected individuals are recombined to generate offspring and characteristics of both parents are transferred to them. During this evolution and reproduction process, it becomes necessary that the selected parents possess high fitness so that probability of high-performance offspring is increased. Therefore, a lot of emphasis is placed on better selection strategy that lead to best individuals. The design of a selection strategy puts emphasis on the competitive environment chromosomes has to go through. The two commonly used selection operators are – fitness proportionate selection and tournament selection.

In the present study, tournament selection is used which is based on competition within a subset of the population. A number of individuals, equal to the tournament size, are selected at random, and a selective competition takes place. The winners of the tournament are then selected for reproduction. The size of tournament can vary from a small to a large sized tournament depending upon the requirements of parents. In the smallest possible tournament, two individuals compete with each other to differentiate a winner and a loser. The tournament size allows to adjust the selection pressure. A small tournament size causes a low selection pressure, and vice-versa [5].

5.2 Crossover

The crossover operator combines the genetic material of two parents by swapping a part of one parent with a part of the other. The offspring contains the characteristics of both the parents involved in reproduction. The probability of performing crossover i.e. crossover rate determines when to perform crossover. The crossover rate lies between 0 and 1.

In the present study, order crossover is used which is based on the idea of preserving the order of the genes of a chromosome and is well-suited for sequencing problems. In this operator, two parents are mapped to each other and two crossover points are selected at random. The portion of first parent within the points is copied to child as it is. The remaining part of the child is filled with elements from the second parent. While copying from second parents the elements are copied in the same order as they are present in parent. This crossover used the sliding motion to fill the elements left by transferring the mapped positions as shown in Fig 3.

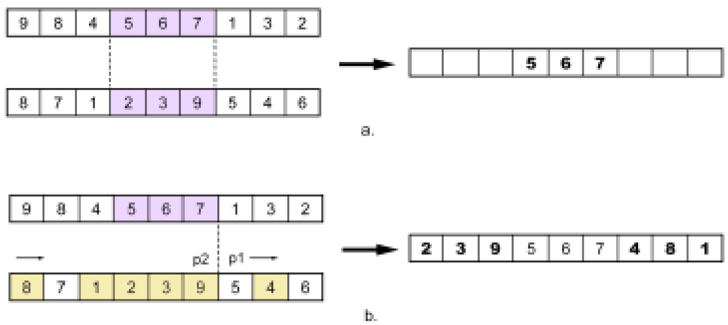


Fig. 3. Order crossover. (a) Gene transfer from first parent to offspring. (b) Gene transfer from second parent to offspring.

5.3 Mutation

The mutation operator plays a vital role in the GA. In each iteration of the algorithm, mutation can potentially uncover useful novelty. In contrast, crossover, if applied as a sole method of generating diversity, ceases to generate novelty once all members of the population converge to the same genotype. The principle of innovation or generating new genetic material serves as the driving force behind mutation. It ensures the degree of dispersion along with diversity. At low diversity levels, as the individuals of a population tend to possess similar characteristics it becomes difficult to generate variety. So, a high mutation rate is required to generate sufficient amount of new genetic material. In setting an appropriate mutation rate, the aim is to select a rate which helps generate useful novelty but which does not rapidly destroy good solutions before they can be exploited through selection and crossover.

In the present study, insert mutation is used. In this operator, two elements of a chromosomes are selected at random as shown in Fig. 4. Next, the second element is moved next to the first element by shifting rest of the elements to accommodate. Through this operator, most of the order and adjacency information is preserved as sliding of elements takes place.

Fig. 4. Insert mutation.

6. Fitness function

In each generation, the fitness of every individual in a population is evaluated. Based on the fitness levels of individuals, more fit individuals are differentiated from less fit individuals. These individuals are stochastically selected from the current population as they possess higher chances of survival. The process imitates natural phenomena of 'survival of the fittest'. In GA terms fitness measure is a derived form of the objective functions. On the basis of performance, a fitness score is given to the candidate, which represents the candidate's ability to satisfy the objective function. Penalties are awarded for every constraint violation by assigning penalty factors during fitness evaluation [8].

Based on objective functions defined in Section 3, a chromosome is assigned a fitness score as follows: -

$$f = \frac{1}{\phi_1 D + \phi_2 P + \phi_3 C + \phi_4 T}$$

7. Results and discussions

To test the performance of genetic algorithm, a number of experiments were performed on a variety of mechanical parts. One such experiment is presented here incorporating the clutch bell inspired component as shown in Fig. 5.

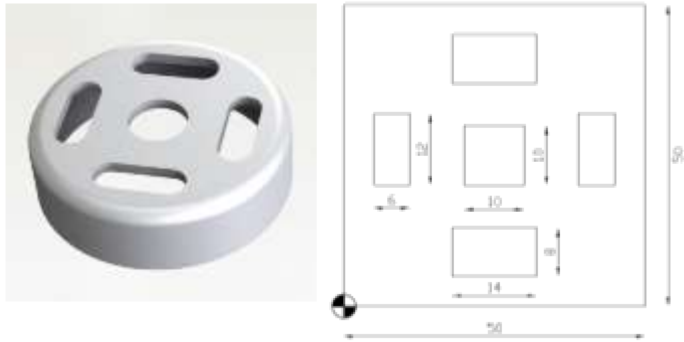


Fig. 5. A clutch bell inspired test component.

Tests were performed to check the diversity levels in population during evolution. A low level of diversity results in premature convergence and solutions obtained are local optima. On the other hand, although a high level of diversity provides a good exploration of search space, but if it is too high, population become heavily dispersed and cannot converge to a solution. It takes longer times to improve and converge. Therefore, a moderate level of diversity is acceptable in a population during evolution. For this purpose, maximum fitness score of a population and average fitness score are selected as performance parameters for this study.

For this experiment, the values of crossover rate are varied from 0.1 to 0.9 and mutation rate is varied from 0.005 to 0.3. For each value, a number of GA runs are executed and average values of fitness scores are recorded at every generation during evolution. The effects of crossover and mutation rates are shown in Fig. 6 and Fig. 7, respectively.

The operators of diversity generation can have two opposite effects on fitness of chromosomes. Either it is constructive in nature or destructive. While constructive crossover enhances the fitness of chromosomes, destructive crossover reduces the fitness that is why it is called destructive. The nature of crossover is decided on the basis of offspring fitness compared to its parent's fitness values. If the fitness of offspring lies within 2.5% above or below the fitness of its parent, it is neutral in nature. A crossover which results in an offspring fitness below 2.5% of parent is considered destructive in nature while that with above 2.5% is considered as constructive. The nature of crossover impacts the quality of solution and time taken to reach convergence. A constructive crossover is always desirable. On the other side, mutation is always considered as destructive in nature. This is due to the fact that chances of fitness improvement are very less, i.e. of the order of one in a thousand.

The overall effect of fitness of parent population and offspring population largely depends upon the rate of crossover which requires a careful selection of rate. A generally accepted range of crossover rate is 0.6 – 0.9 and that of mutation is 0.1 – 0.3. But the actual rate of operators is influenced by a number of factors such as design of operator, selection of operator, size of population, etc. Mutation rate is inversely proportional to the size of population. As a matter of fact, actual rates for a particular genetic algorithm run may vary beyond the general range. To determine the values of these rates in context with current problem and conditions, a number of experiments have been performed.

Two fitness measures for studying the effect of crossover and mutation rates in a GA run are – variation of maximum fitness in a population and variation of average fitness of entire population. The former is used to describe the fitness of solution and convergence. The latter describes the overall fitness of a population and its variation. The difference between the two lines describe the level of diversity within a population. A large difference denotes high level of diversity and vice-versa.

In nature, genes of an organism are present in DNA in a particular order. They are sequenced with respect to their properties and functionalities. While transferring the genes to an offspring, a set of genes of one parent combines with another set of genes of another parent. The combination also maintains the order of genes in which they must be located for proper development of an offspring. This analogy is followed in genetic algorithms too.

A chromosome contains genetic material or genes in an order. The ordered genes are propagated during evolution to further generations. Any two chromosomes in a population can be distinguished on the basis of ordered genes. These also describe the similarities between the two. A schema is a subset of these ordered genes that is present in chromosome and propagated. The schema which contribute to high fitness of a chromosome are maintained while those with lower fitness values are deleted. The

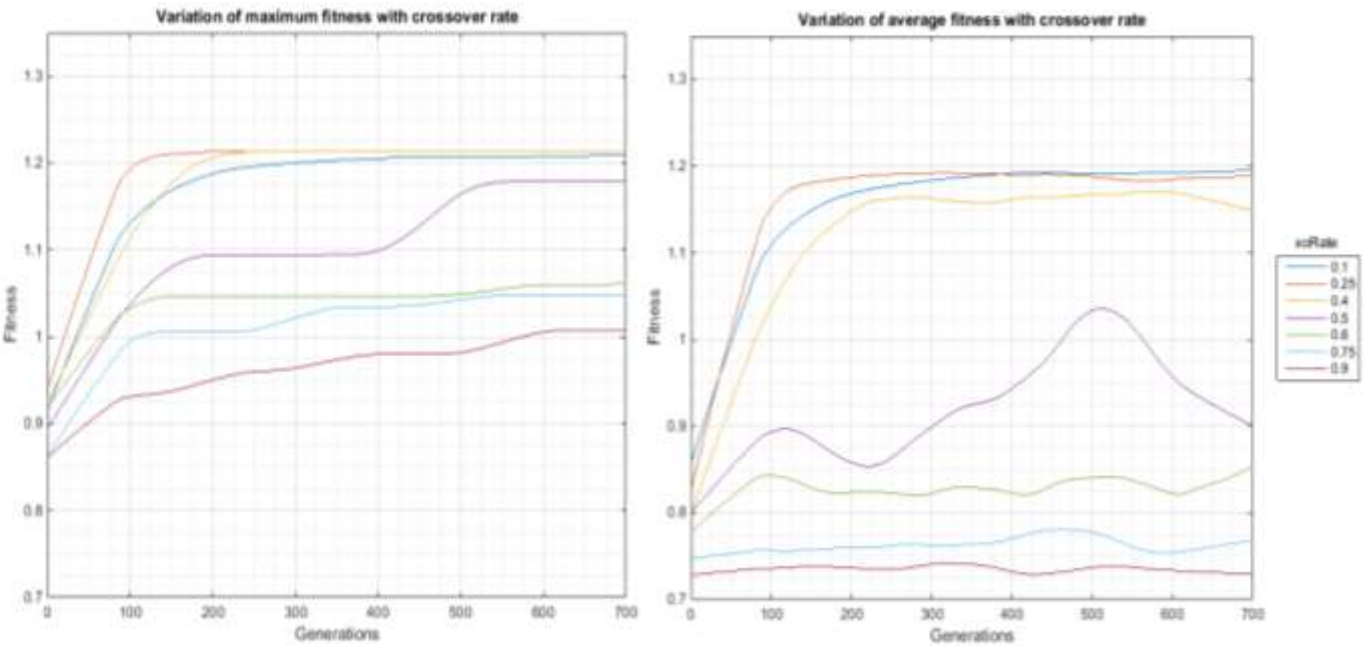


Fig. 6. Variation of fitness score with crossover rate.

schemata are created by genetic operators by rearranging the ordered genes.

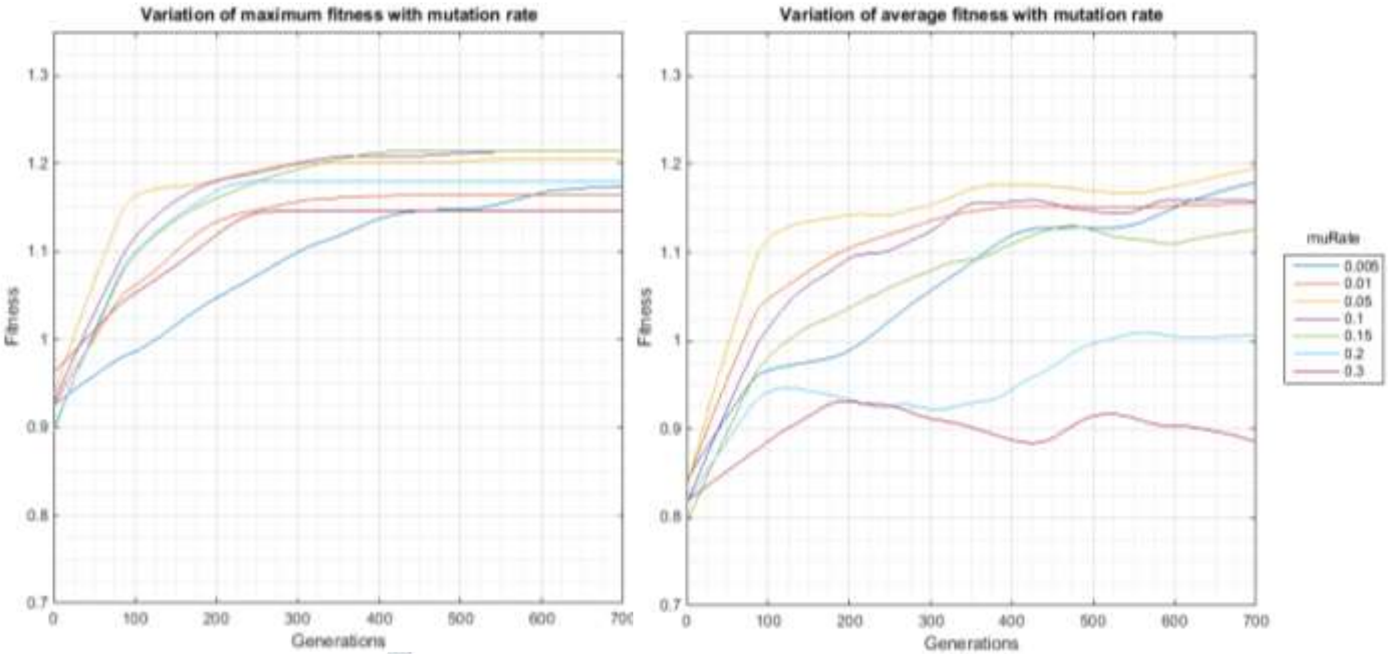


Fig. 7. Variation of fitness score with mutation rate.

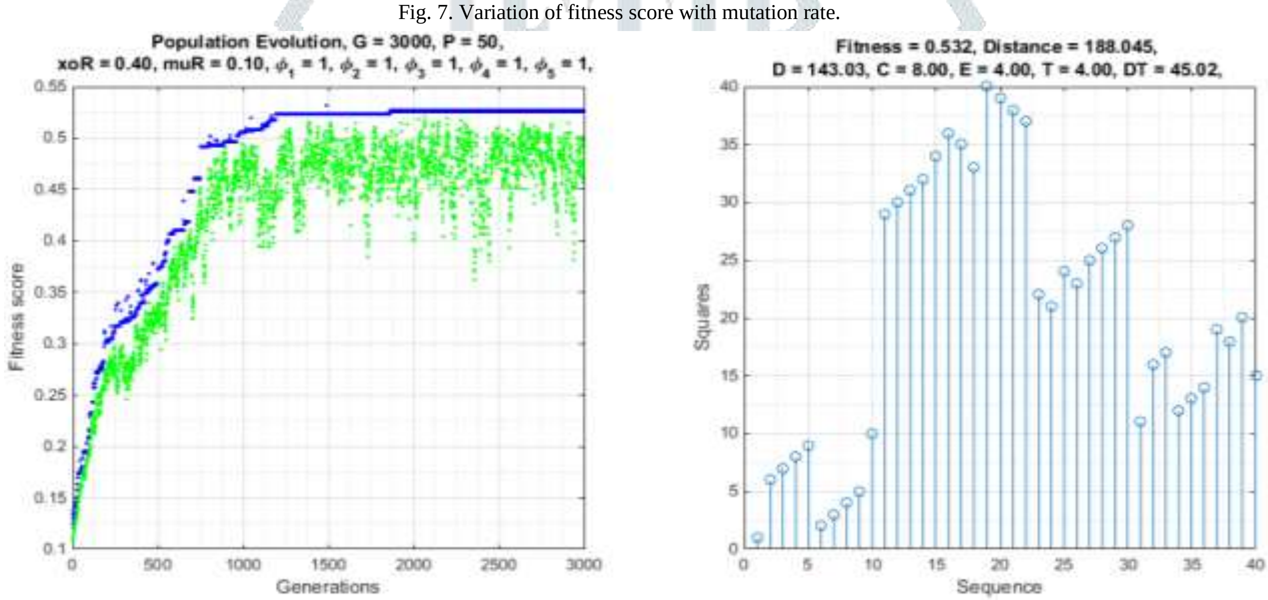


Fig. 8. Population evolution and convergence to optimal solution.

It has been considered that the schemata describe the fitness of chromosomes. As schemata grows with improving fitness, fitness of the chromosome also increases. But when a schemata is disrupted during diversity generation, fitness of both schemata and chromosome reduces. In an entire population, a number of schemata can grow in parallel leading to multiple solutions to the problem. It is worth mentioning that all schemata are not created equally in a population. Some of them have higher chances of creation and growth. The survival probability of a schema depends upon many factors. Most important factors are order of schemata, length of schemata and average fitness of the schemata.

For instance, let us assume a chromosome with length of 9 genes. A schemata is taken as 12***4***, where 1, 2, 4 are elements or genes of schemata and asterisk (*) represent other genes of chromosome. A number of chromosomes can possess this schemata, such as – 128364975, 127534698, etc. The order of schemata is taken as the number of fixed genes in the schemata. In above example, order is 3 as the fixed genes are 1, 2 and 4. The length of schemata is the distance between first and last fixed gene. So, above schemata have a length of 5 as position of first is gene is 1 and that of last gene is 6, i.e. 6 – 1 = 5.

The growth of schemata in a population is directly proportional to the ratio of average fitness of schemata to average fitness of population. The schema theorem by D. Goldberg considers that short length, low order, and above average schemata have higher probability to survive and grow in a population. Such type of schemata are recombined, resampled to form chromosomes of higher fitness. These play an important role in the action of GAs and therefore termed as 'building blocks'. The building blocks enables a GA to seek better solutions by combining such blocks of genes together. In fact, they combine to form optima or near optima. The fitness of building blocks largely affects the solution obtained through GA. Better quality blocks drives the search towards good solutions and convergence whereas misleading building blocks cause a problem to take higher times to converge and find near optimum solutions. The evolution of population in the GA run is shown in Fig. 8. It can be seen that the optimal solution is achieved after 2000 generations.

The random and unpredictable behavior of GAs can be addressed to some extent with these concepts of schemata and building blocks. It is very difficult to completely determine behavior mainly because of random nature of evolutionary processes. Randomness serves as the prime driver of such algorithms. This is the reason why genetic algorithms can find multiple solutions to a problem or in some cases no good solution is obtained. GA operators work on the probability values to modify population. The study of building blocks created during a GA run can reveal direction of the run and solution obtained.

8. Conclusion

In this paper, genetic algorithm was implemented on multi-objective optimization of milling toolpaths. The optimization problem was defined using multiple objective functions spanning over key parameters of milling such as tool travel, jerk, tool positioning, etc. GA operators such as selection, crossover, and mutation were modeled and a fitness function was defined including the objective functions. Tests were conducted to validate the implementation accuracy of GA method. It was observed that a careful selection of GA parameters is crucial in convergence to global optima. Also, the GA parameters are subjective to the constraints and fitness variables in a particular domain. The test results are helpful in improving the performance of GA in the optimization of milling toolpaths.

References

1. S. Gurel and M. S. Akturk, “Considering manufacturing cost and scheduling performance on a CNC turning machine,” *Eur. J. Oper. Res.*, vol. 177, no. 1, pp. 325–343, 2007.
2. C. Anderberg and S. Kara, “Energy and cost efficiency in CNC machining,” *7th CIRP Conf. Sustain.*, pp. 1–4, 2009.
3. E. Schoenberger, “Some dilemmas of automation: strategic and operational aspects of technological change in production,” *Econ. Geogr.*, vol. 65, no. 3, pp. 232–247, 1989.
4. M. G. Mehrabi, a G. Ulsoy, and Y. Koren, “Reconfigurable manufacturing systems : Key to future manufacturing,” *J. Intell. Manuf.*, vol. 11, pp. 403–419, 2000.
5. A. Brabazon, M. O’Neill, and S. McGarraghy, *Natural Computing Algorithms*. Springer, 2015.
6. R. Saravanan, P. Asokan, and M. Sachithanandam, “Comparative analysis of conventional and non-conventional optimisation techniques for CNC turning process,” *Int. J. Adv. Manuf. Technol.*, vol. 17, no. 7, pp. 471–476, 2001.
7. J. Holland, “Adaptation in Natural and Artificial Systems.” 1975.
8. D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Seventeent. Pearson Education, Inc., 2015.
9. C. M. Fonseca, S. Sheffield, and P. J. Fleming, “An Overview of Evolutionary Algorithms in Multiobjective Optimization,” *Evol. Comput.*, vol. 3, no. 1, pp. 1–15, 1995.
10. M. Kovacic and J. Balic, “Evolutionary programming of a CNC cutting machine,” *Int. J. Adv. Manuf. Technol.*, vol. 22, no. 1–2, pp. 118–124, 2003.
11. J. C. Chen and T. X. Zhong, “A Hybrid-Coded Genetic Algorithm Based Optimisation of Non- Productive Paths in CNC Machining,” *Int. J. Adv. Manuf. Technol.*, pp. 163–168, 2002.
12. A. Krimpenis and G. C. Vosniakos, “Optimisation of multiple tool CNC rough machining of a hemisphere as a genetic algorithm paradigm application,” *Int. J. Adv. Manuf. Technol.*, vol. 20, no. 10, pp. 727–734, 2002.
13. B. Vaupotic, M. Kovacic, M. Ficko, and J. Balic, “Concept of automatic programming of NC machine for metal plate cutting by genetic algorithm method,” *J. Achiev. Mater. Manuf. Eng.*, vol. 14, no. 1–2, pp. 131–139, 2006.
14. M. Lee and K. Kwon, “Cutting path optimization in CNC cutting processes using a two-step genetic algorithm,” *Int. J. Prod. Res.*, no. October 2014, pp. 37–41, 2007.