OPEN ACCESS INTERNATIONAL JOURNAL OF SCIENCE & ENGINEERING

DISCRETE MODELING FRAMEWORK FOR WORK AREA APPROXIMATION IN POCKET MILLING

Munish Kumar[1], Dr. Pankaj Khatak[2]*

Mechanical Engineering Department, Guru Jambheshwar University of Science & Technology, Hisar 125001, India
**Corresponding Author Email: pankajkhatak@gmail.com*

***ABSTRACT:* 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. In any such environment, selection of optimal data is very crucial for the industry. This data includes machining parameters and the toolpath. The distance travelled by a tool during machining is a key component of the optimization problem as it directly affects the machining time and tool life. This paper presents a modeling framework based on discretized squares for approximation of work area. Results indicate the effectiveness of the presented framework in reduction of search space and convergence time of the optimization problem.**

Keywords: *CNC milling, discretization, reduction, search space, convergence time*

I INTRODUCTION

CNC technology has developed to greater levels of flexibility and productivity. In addition, new machine design concepts have made it possible to integrate different types of machining processes into one process on a single machine. It is continuously transforming machines to be economical for low volume work. In addition, its increasing power has allowed the development of high production equipment with enhanced capabilities. The machines can now be used with user-friendly software applications which can be run of a typical personal computer (PC). A CNC machine can be used for cutting, milling, drilling, and roughing of almost any material. In general, it can be used for any project that requires accurate, repeatable, and efficient mechanical movements for extended periods of time. The applications of CNC will not necessarily create new objects but rather automate processes that once had been done manually.

A typical CNC part program can be prepared by a number of methods. Programming can be done using a CAD/CAM software, or by an automatic programming system, or manually by a programmer[1]. In a CAD/CAM software, programming is accomplished through three prime phases starting from a graphical representation of the part. In the second phase, based on part geometry, appropriate tool path is defined along with other machining parameters such as feedrate, spindle speed, depth of cut, tool used, and tolerance levels required. A part program is generated in the final phase in any software specific file format (.nc,.mpt,.mpf, etc.). Some popular CAD/CAM software are MasterCAM, HSM, Siemens NX, and PowerMILL.

For every manufacturing industry, production rate and quality are two important aspects of production[2]. 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[3].Optimization of machining process provide a number of advantages over conventional practices. To achieve higher standards, it becomes obligatory to use best possible conditions. Depending upon the needs of an industry, the processis directed to achieve different objectives[4]. Generally, objectives directly affecting the quality, surface finish, machining times and costs are more emphasized[5]. An optimization problem is formulated to achieve them. It comprises of some objective functions subjected to some constraints. Either a function is maximized or minimized depending upon nature of the problem, e.g. minimization of machining time, minimization of machining and tooling cost, maximization of production rate, minimization of surface roughness,etc.

In conventional systems, where part programmers are assigned CNC programming, the machining data is obtained through experience or handbooks (or a guess) which may be conservative and uneconomic leading to production limitations. K. Park and S. Kim [4] studied that the above selected machining data can prove to be incapable of eliminating inordinate amount of matching errors from tool failures such as tool deflection, wear, breakage, etc. Consequently, these conservative and non-optimal machining parameters results in low metal removal rate. In such cases, optimal data is needed to be described to consider economic, technological and geometric limitations for recommended machining conditions [6].

The process of optimization of a cutting process is carried out in two stages namely, modelling of parameters and determination of optimal or near-optimal conditions. In the first stage, a relationship is formed between various parameters involved in a machining process. These parameters can be input – output parameters on in-process parameters. Some important parameters in machining process are, cutting speed, feedrate and depth of cut [7], [8]. The relationship between these parameters was first depicted by Taylor’s tool life equation. Some researchers have used the same equation in their works whereas some of them modified Taylor’s equation according to their assumptions and objectives. Some important modelling techniques as described by I. Mukherjee and P.K. Ray are, Statistical regression technique, Artificial neural network and Fuzzy set-theory based modelling techniques [9].

The distance travelled by a tool is a key component in machining to look upon. It contributes predominantly to machining time of any product. A vague toolpath in which tool has to travel distance more than required, claims increased machining time lowered productivity of a plant. Therefore, a particular attention is given to optimization of toolpaths in machining by a number of researchers. Optimization of a toolpath is directed to reduce the distance travelled by a tool to a cut a particular part or contour. Another parameter which is optimized is material removal rate. An optimal toolpath involving optimization of boththese parameters increase the productivity of a plant, lowers machining time, and reduces tool travel thus tool wear.

M. Kovacic et al. proposed an evolutionary concept of CNC toolpath optimization and programming for both machining and turning operations. Their concept was based on discretization of machinable area. A machinable area is considered to comprise of tool motions, which are discretized into squares in case of turning and boxes in case of milling[10]. A similar approach to above presented work is proposed by J. Barclay, V. Dhokia and A. Nassehi. In their study [11], they used a simplified model to reduce the search space of optimization problem. They divided the workpiece into layers, and each layer was discretized into squares. A. Nassehi, W. Essink and J. Barclay proposed a discretization framework for generation and optimization of milling tool-paths [12]. The framework was based on evolutionary approach that allows various properties to be optimized without changing the algorithm. In order to generate a discretized model, a grid of equidistant points was superimposed over the geometry of a part beingmachined.

The process of discretization provides a good method of approximation of a workpiece. Using a discretized set of points or squares, a part can be easily approximated. Hence a small number of points is necessary to find an optimal sequence rather than using a large set of points. This reduces the search space for an optimization method along with a decrease in computational complexity. Due to these reasons a discretization framework is proposed in this research based on the concept of design elements and squares grid.

A prepared drawing of part marks various dimensions and features of that part such as pockets, holes, fillets, contours, etc. These design features are needed to be formed on a workpiece through machining. Efforts are made to ensure that proposed design features are cut with efficiency and quality. It is the assignment of a planner or programmer to efficiently machine those features onto the workpiece using adequate cutters. A process plan is prepared for the same consisting of several cutter locations along with machining parameters associated with them. After selecting an appropriate cutter for current operation, cutter location data (CL data) is used to create toolpaths. It connects a number of cutter locations on workpiece in sequence which cutting tool has to follow to machine a particular design feature. A number of optimal size cutters and relative toolpaths are needed for creation of an efficient process plan. Machining parameters, on one hand, influence the selection of cutters and toolpaths, but design features play a dominant role in efficient preparation of toolpaths and processplans.

In this paper, a discretization approach is presented for approximation of work area. The presented framework provides an effective method of toolpath generation and optimization. The paper is formatted as follows – in Section 2 the discretization framework is presented. Section 3 discusses the tool offset and discrete square reduction. In Section 4, mapping of the framework in computational domain is presented. Section 5 provides the results of experiments.

II DISCRETIZATION FRAMEWORK

The problem of large space search spaces is evident in CNC milling. It restricts the use of optimization techniques with efficiency. To address this problem, a discretization framework is hereby proposed. The proposed framework reduces the search space by lowering the number of cutter locations required for machining. In this context, actual cutter locations or points are approximated through discrete elements (not to be confused with design elements) of finite dimensions.

The framework is based on the concept of design elements as discussed above. It is conceptualized that a part is a mere grouping of design elements on a workpiece. Individual elements are dealt separately in a process plan. In certain cases, different tools may be employed to cut separate elements. The proposed framework considers a workpiece to be divided into a grid of identical squares. This grid covers entire area on surface of a workpiece (i.e. in x-y plane). Size of squares can be chosen arbitrarily depending upon the size of workpiece. Here, irrespective of size of a workpiece, squares of 1mm are used. The area of workpiece where a design element is present, squares becomes active, therefore termed as active squares. On the other hand, in non-machinable area, the squares are termed as inactive squares.

To illustrate the proposed framework, let us consider a random 3D part as shown in Figure 1a. This part contains three design elements which are, a rectangular pocket, and two circular arcs. Both arcs are to be cut with tools of same size, but in case of pocket, tool size can be varied. Workpiece in this case is a rectangular block of size, let’s say 20 x20 mm and height of block is 10 mm. A grid of size of workpiece i.e. 20 x 20 mm is chosen. In the grid, squares of 1 mm size are selected to approximate the workpiecearea.

The discretization framework applied to given 3-dimensional part is shown in Figure 1. Various features of part are shown in Figure 1(a). It can be noted that the framework deals with only top-most layer of the workpiece, ignoring thickness of part. Figure 1(b) shows a discretized workpiece with grid representation of various areas. In the above part, 400 squares are used to approximate the given workpiece of 20x20 mm size as the size of each square is 1 mm. The squares marked in red are active squares which represent elements of the part. Other empty squares, market in white, are inactive squares. It is worth mentioning that the centre of a square is used to make calculations and measure distances, not the square itself. Only active squares contribute to a cutting strategy. A cutting tool has to visit every active square in order to completely cut a workpiece and produce given part. The sequence in which these squares are visited is of major concern in ourproblem.

In this way, a workpiece is approximated through square grid. It results in reduced search space, as of now, rather than searching for every point in Cartesian space, active squares can be trialed. In this reduced search space, optimization techniques can be easily implemented with effective results. Similar discretization frameworks are proposedandtestedbyM.Kovacicetal.[9],J.Barclayetal. [10] and A. Nassehi et al. [11]. In these works, square elements are selected for approximation, except forA.Nassehi et al who used grid of equidistance points.

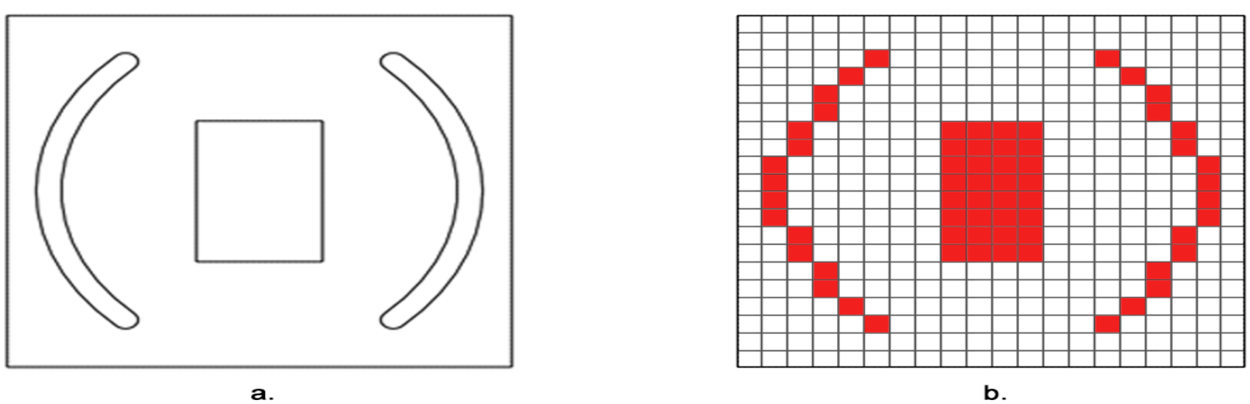


Figure 1. Discretization framework showing, a. Features of part, b. Approximation of given part by square grid.

1. Tool Offsets and Reduction ofSquares

A grid of squares has active and inactive squares separating machinable areas and blanks of workpiece. The cutting tool is required to pass through machinable area by visiting active squares. To machine completely a design element,ithastovisitalltheactivesquaresinsidethat element. Cutting tool is placed on the center of an active square. In such cases when cutting tool diameter is greater than size of squares in grid, it is considered to occupy a group of squares rather than a single square. Center of cutting tool coincides with center of the middle square (central square) with all adjacent squares falling in vicinity of the tool as shown in Figure 2. A cutting tool of diameter 3 mm is placed on grid of squares 1 mm is size. Therefore, cutting tool will occupy portions of nine squares, central square and eight adjacent squares shown in red color.

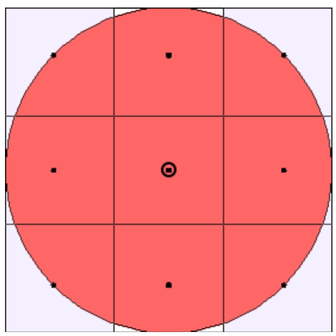


Figure 2. Cutter occupying group of squares.

It is evident from above figure that a cutting tool of size larger than size of squares need not visit each square individually. Rather if it is placed at one square, all squares falling under its vicinity are considered occupied. In this way, the central square can be retained and other adjacent active squares can be eliminated. This property can be better visualized with the help of illustrative example. Consider an illustration shown below in Figure 3. A number of active squares are highlighted on a grid of squares, shown in Figure 3a.

There are 21 active squares on the grid. Conventionally, all 21 squares need to be visited if a tool of 1 mm is considered starting from square 1. Let us now consider a cutting tool having diameter 3 mm. In this case, if this tool is placed at square 1, it will extend beyond the boundaries of machinable area as the diameter is greater than 1. Similar will be the case when placed at all boundary squares, such as 2, 3, 4, 7, 8, 9, and so on. After considering tool offset value, the first active square where tool has to be placed will be square 5, as it will occupy above mentioned squares too. This is because these squares come in the vicinity of central square 5. Therefore, it will automatically occupy all its adjacent squares numbered 1 to 9, as shown is Figure 3b.

Now, to move the cutting tool towards right, the subsequent squares to be selected must lie outside the vicinity of cutting tool. This condition implies no square inside vicinity of cutting tool need to be selected, therefore, the square right to square 6 will be selected subsequently. But first, cutting tool will be placed at the right flank of grid. The algorithm works simultaneously from all possible flanks or sides of machinable area, as shown in Figure 3b. In the left side, similar conditions apply and the central square where tool is placed is selected. To move towards left, the closest square outside the vicinity of cutting tool is selected subsequently.

In this way, a system of 21 active squares is reduced to only 3 active squares, termed as "reduced active squares or R-active squares". The R-active squares are shown in Figure 3c, highlighted in dark blue. Similarly, reduction of active squares in previous problem is shown in Figure4.

For this system of active squares, the decrease in number of squares might seem small. But when the framework is appliedtoasystemhavinghundredsofactivesquares,itcan

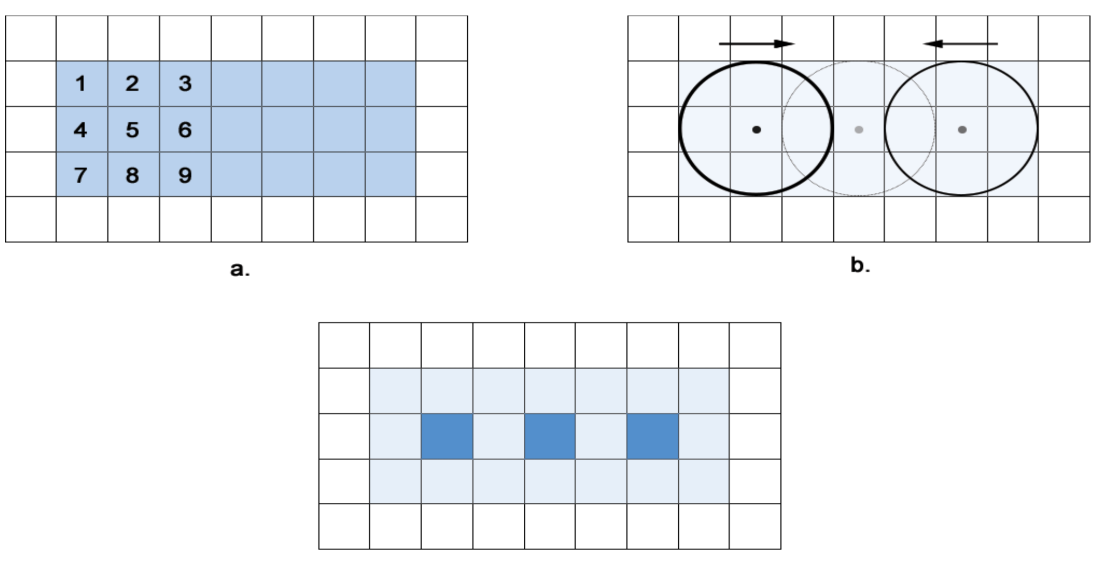


Figure 3. Reduction of active squares, a. Active squares on a grid, b. considering tool offsets, c. Reduced active squares after offsetting.

drastically reduce the number to only a few. This is evident through experiments that depending upon tool size, a drastic reduction in search space is observed. This reduction of search space turns a complex problem into a simpler one and significantly affects the functionality of optimization algorithms. It facilitates efficient application of algorithms and secures better solutions.

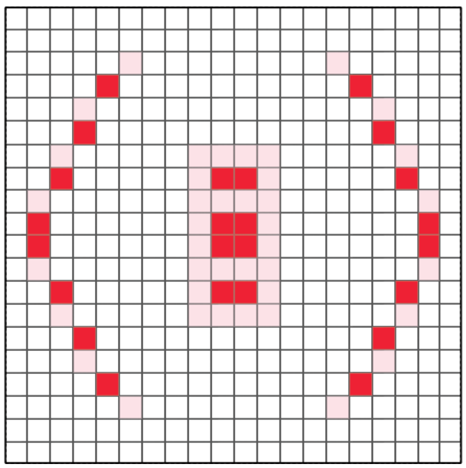


Figure 4. R-active squares in previous problem.

Also, the reduction of active squares is performed on the basis of tool size. It is irrespective of size of machinable area, number of squares in grid, or size of squares. Therefore, the algorithm is able adapt to changes in tool sizes. A tool of any size, which can satisfy constraints of surface finish and material removal rate, can be used to machine the area. This eliminates the need to change or modify the algorithm for different tool sizes. The algorithm is developed to be self-adaptable to any such changes. Other researchers have considered a single size of tool to apply the discretization framework [10]-[12]. No multiple tools can be used in a single optimization problem. The proposed framework is able to facilitate use of multiple tools in a single problem. Different design elements of a part can be machined with different toolsizes.

2. Mapping of DiscretizationFramework

The concept of discretization using grid of active and inactive squares is presented above. To implement such framework in a programming language, it is important to map the framework and grid structure to a digital form. MATLAB programming language is used to implement the proposed framework. It is a high-level language having vast capabilities and functionalities. It readily supports a number of data structures including, string, cell, matrix, structure, table, etc.

For the mapping of proposed framework, matrix data structure is used. A matrix is a typical form of array storing a number of elements of same type (integers, floats, etc.). As we have considered a two-dimensional grid having active and inactive squares, a two-dimensional matrix would serve the purpose. It can be noted that in a typical two-dimensional matrix, elements are given a value and are stacked into rows and columns. The number of rows and columns gives the order of the matrix. This 2D matrix is imposed over the square grid. Order of matrix is kept same as the size of grid. As stated earlier, size of each square in grid is taken as 1 mm. Number of rows of the matrix is equal to the length of grid, or number of squares counted horizontally. Similarly, number of columns of matrix is equal to width of grid, or number of squares counted vertically. In this system of matrix, each element of matrix represents a particular square of the grid on which it is imposed. The value of a particular element of matrix tells the type of square it is representing while the index values of element (i and j) denote x and y-coordinates of center of that square. It implies that matrix is a numeric representation of the grid and an element is a numeric representation of asquare.

Only binary values are allowed to be assigned to the matrix. Any particular element of matrix can have only two values namely, 0 and 1. This value describes the type of square represented by that element. A value of 1 denotes an active square where 0 is assigned for inactive squares, as shown in Figure 5. This also holds true after performing reduction of active squares. After reduction, the R-active squares are assigned a value 1 while other are assigned 0. It is worth mentioning that in the proposed framework, origin of workpiece and hence the grid, is fixed at the bottom left corner of the same. This origin cannot be shifted to other locationsontotheworkpiece.Allcoordinatesanddistances

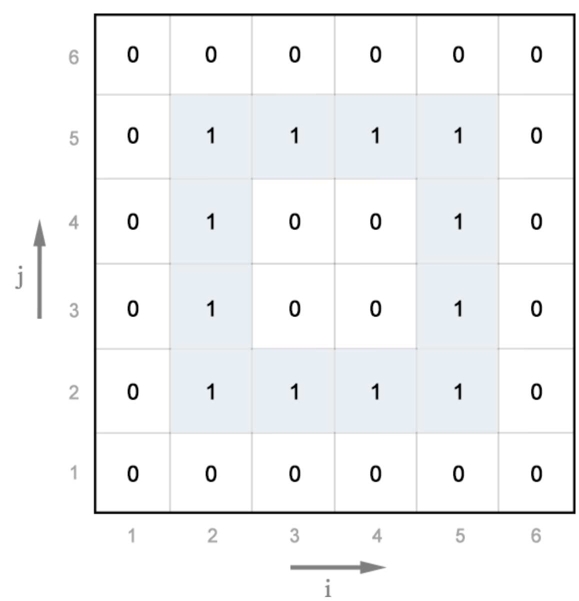


Figure 5. Matrix mapping of a square grid.

are taken with reference from this point.

Typically, first element of a matrix starts from upper left corner of the matrix. This is a general rule that every programming language follows. As the matrix is considered to be imposed over the grid, the bottom left element of matrix must be same the origin or bottom left square. For this purpose, the matrix is inverted upside down and then values are assigned to elements. In this way, first element of matrix shifts from upper left corner to bottom left corner, keeping ordering of elements similar. It is shown in Figure 3.6, numbering of elements starting from bottom left corner of grid (or matrix). It should be noted that numbering of elements, or the index value, starts from 1 and not from 0. Hence, the first square at bottom left corner has an index value [1,1].

After mapping of the system, coordinates of all the active squares are stored in a different data array. In the data array, sequence of coordinates of active squares is taken at random. No specific ordering of squares takes place at this point. We have discussed that correct sequence of squares visited by a tool constitutes an optimal toolpath. Genetic algorithm is applied for this purpose. The data array containing coordinate values of all squares to be visited is supplied to GA. But first it needs to be encoded into a particular representation scheme of genetic algorithms. GA calculates the correct sequence of squares to be visited by cutting tool through evolution. This sequence of squares then constitutes atoolpath.

III RESULTS AND DISCUSSION

The proposed framework provides a method of approximation of parts using a square grid. The approximation is needed as a large number of points makesit very complex and computationally intensive to generate and optimize toolpaths. The proposed framework uses the concept of design features and design elements to represent a part onto the grid with active and inactive squares. Activation of squares is performed by interpolator itself after calculating the data points of a given design element. To further reduce the number of squares, the reduction algorithm is used which reduces the squares by considering tool offsets. In this way, search space of the optimization problem has been reduced to large extent by a small number of squares. The problem comprises of generating toolpaths and optimize them for multiple objective functions and constraints. A toolpath is a sequence of squares visited by a cutting tool. Genetic algorithm (GA) is used for optimizing the generated sequences to obtain a toolpath, or an optimal solution to the problem.

A number of parts were modeled using the presented framework. One such part is shown in Figure 6. The part contains three design features or specifically rectangular pockets of different sizes. The interpolated and reduced discrete squares are shown in Figure 6a and 6b, respectively. It was observed that the presented framework is able to model various design features effectively. In this succession, a series of parts were modeled and the number of squares was recorded. After that, optimization method was initialized and convergence time was noted down. The graph of the observations is shown in Figure 7 and Figure8.

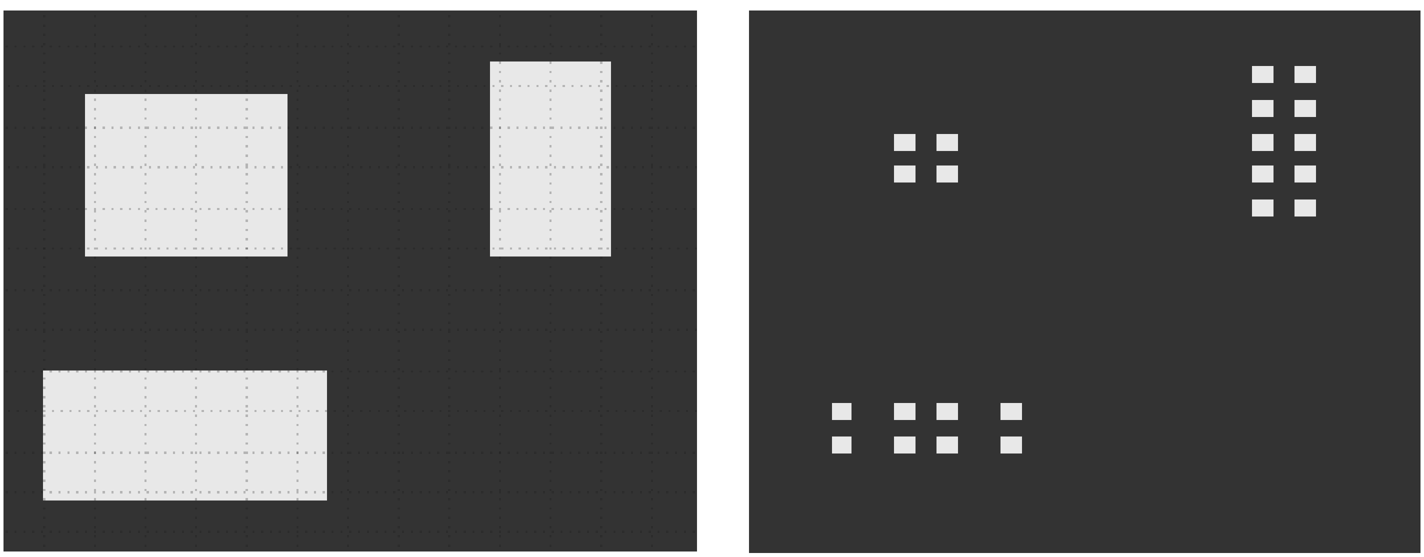
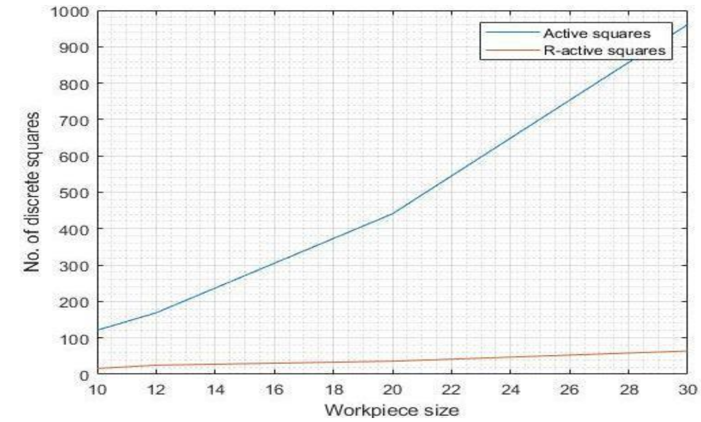


Figure 6. (a) Interpolated squares of test part, (b) Reduced squares of test part.

It is evident from Figure 7 that number of active squares increases exponentially with size of the workpiece. High number of squares makes it difficult for the optimization method to converge to global optima in less time as shown in

Figure 8. However, using the reduction approach, the number of squares in the search space reduces significantly, as high as 90% depending upon the size of workpiece. The corresponding convergence time is very low for the reduced squares**.**



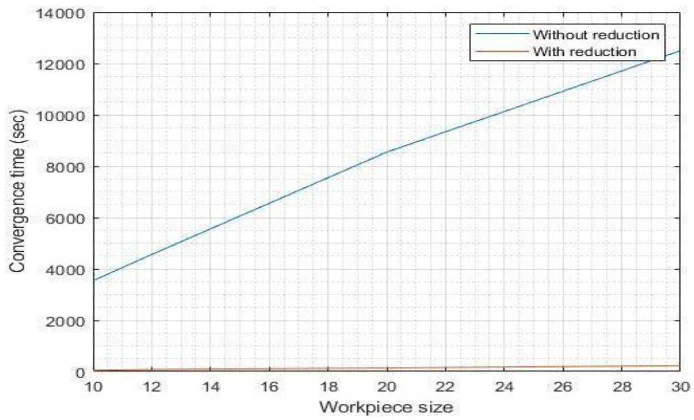


Figure 8. Variation of convergence time with workpiece size.

IV CONCLUSION

The proposed framework consists of discretization of a workpiece into grid of finite squares. A workpiece is added with design features that are cut by a cutting tool to shape the workpiece into a part or component. These design features constitute of simpler design elements such as lines and arcs. The approximation through square grid considerably decreases the search space of an optimization problem. A further reduction is achieved by applying a reduction algorithm. The reduction process deactivates non-required active squares by calculating tool offset values. The presented approach was validated through modeling of different parts. The number of squares representing the search space of an optimization problem, and the convergence time were analyzed. It is thus obvious from the results that the presented framework provides an effective work area approximation of milling surfaces, particularly for pocket milling operations. The reduction approach shows the potential to greatly reduce the convergence time of the optimization problem in milling toolpaths.

REFERENCES

1. J. Sauer, B. Zimolong, and S. Ingendoh, “Maintaining production quality in advanced mechanical engineering organizations: The compensatory function of the CNC-machine operator,” *Hum. Factors Ergon. Manuf.*, vol. 10, no. 4, pp. 465–480, 2000.

2. E. Poutsma and A. Zwaard, “Programming CNC Equipment - The Effects of Automation in Small Scale Industrial Enterprises,” *Int. Small Bus. J.*, vol. 56, no. 3, pp. 460–474,1996.

3. M. J. Kwon and P. Stoneman, “The Impact Of Technology Adoption On Firm Productivity *,” *Econ. Innov. New Technol.*, vol. 3, pp. 219–234,1995.

4. Munish Kumar and Pankaj Khatak “ An Investigation of Conventional and Non-Conventional Optimization Techniques in CNCMachining”

5. K. S. Park and S. H. Kim, “Artificial intelligence approaches to determination of CNC machining parametersinmanufacturing:areview,”*Artif.Intell.Eng.*, vol. 12, pp. 127–134,1998.

6. E. H. T. Wee, A. Y. C. Nee, and T. N. Goh, “Off-line adaptive optimization of CNC lathe turning.,” *Int. J. Prod. Res.*, vol. 28, no. 12, p. 2157,1990.

7. B. Malakooti and J. Deviprasad, “An Interactive Multiple Criteria Approach for Parameter Selection in Metal Cutting,” *Oper. Res.*, vol. 37, no. 5, pp. 805–818,1989.

8. 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.

9. I. Mukherjee and P. K. Ray, “A review of optimization techniques in metal cutting processes,” *Comput. Ind. Eng.*, vol. 50, pp. 15–34,2006.

10. M. Kovacic, M. Brezocnik, I. Pahole, J. Balic, and B. Kecelj, “Evolutionary programming of CNC machines,” *J. Mater. Process. Technol.*, vol. 165, pp. 1379–1387,2005.

11. J. Barclay, V. Dhokia, and A. Nassehi, “Generating Milling Tool Paths for Prismatic Parts Using Genetic Programming,” *Procedia CIRP*, vol. 33, pp. 491–496, 2015.

12. A. Nassehi, W. Essink, and J. Barclay, “Evolutionary algorithms for generation and optimization of tool paths,” *CIRP Ann. - Manuf. Technol.*, vol. 64, no. 1, pp. 455–458,2015.