A virus-evolutionary multi-objective intelligent tool path optimization methodology for 5-axis sculptured surface CNC machining

Abstract

Sculptured surface machining is a material removal operation essentially adopted to manufacture complex products. Computing optimal tool paths with reference to ideally designed CAD models is indispensable to be able to suggest machining improvements in terms of high quality and productivity. The present paper proposes a new methodology based on a virus-evolutionary genetic algorithm for enhancing sculptured surface tool path planning through an automated selection of values for standard 5-axis end milling strategies' machining parameters to be decided upon in the context of a simulation-based; software-integrated, multi-objective optimization problem. The problem involves surface machining error as the first quality objective represented via the mean value of chordal deviations that tool path interpolation yields and effective radius of inclined tools that affects scallop. Machining time is the second quality objective entering the problem to assess productivity; whereas the number of cutter location points created for each tool path evaluation is also considered. Tool type, tool axis inclination angles as well as longitudinal and transversal steps are considered as the independent parameters in the case of 5-axis machining. Results obtained by conducting evaluation experiments and simulation tests accompanied by an actual machining process provided significant insight concerning the methodology's efficiency and ability of suggesting practically viable results.

Keywords: Sculptured surface machining, virus-evolutionary genetic algorithm, tool path generation, multi-objective optimization

1 Introduction

As cutting-edge products designed and ultimately consumed are getting smarter, more versatile, and more complex, "intelligently" formulated tool paths are now needed so as to meet pressing demands in the field of sculptured surface machining. Current investigations for planning precise tool paths has brought to light noticeable contributions concerning the proper application of already existed tool paths; the development of tool paths considering technological parameters and mechanics of processes, tool path generation considering isoparametric schemes via simulation-based studies [1] and the generation of tool paths by taking into account data points been measured [2] with reference to the geometry and the features of 3D CAD models [3]. Tool paths currently suggested and implemented in literature involve several techniques being used for their development such as surface-error-based segmentation in the feed forward direction of the tool; subdivision surface; twice differentiable continuous parametric surfaces; constant scallop-height operations and curve fitting using point cloud data.

Current manufacturing technology involves the usage of parametric surfaces when preparing complex sculptured surfaces for machining with the aid of computer-aided manufacturing (CAM) environment. The use of parametric surfaces allows the representation of complex design properties with less data. Such properties involve C^1, C^2 and C^n continuous surfaces like Bezier, B-spline and NURBS being represented by knots, weights and control points. Aiming towards the ability of getting as close as possible to the design surface by maintaining tolerance requirements, commercially available CAM systems currently embed functions to approximate such surfaces using advanced interpolators like NURBS, yet; enough evidence have assembled by several researchers to support that same precision cutting may be achieved by implementing linear interpolation as well; provided that huge NC files must be stored in NC units to enable the accurate representation of varying slope and local curvatures [4]. Nevertheless, this is not of major concern given the current state of high-tech CNC controllers which have become more sophisticated and “adept” while coping with large NC data, under fast processing rates. In addition, high frequency servo loop functions integrated to CNC systems, allow smoother machining operations whilst maintaining good transition from one move to the next, in terms of feed rate [5]. What is more, is that most of the CNC units found today in industry, are yet to integrate functions to support NURBS interpolation implying that they are still in their infancy. Such aspects have already led to the reconsideration of employing common interpolators when it comes to high-precision machining [6].

Conventional interpolation enables tool paths to discretize free-form surfaces to series of straight lines owing to the inherent nature of tool motions and surface properties mentioned above. Discretization involves sampling the theoretical surface with a certain step towards the feed direction. Given the surface geometry denser segmentation is desired when it comes to steep regions and larger in the case of flat areas. This spacing between successive tool positions along the tool path yields chordal deviations given the local curvature and the chord connecting two consecutive tool contact points [7, 8]. Therefore it is the job of the tool path planner to harness the advantages offered by existent CAM software properties and decide machining parameters' settings by experience or trial-and-error efforts ensuring that the error will remain within preset machining tolerance given the target surface as the tool moves along cutter location points by selecting suitable machining parameter values under a given cutting strategy. Crucial machining parameters to determine the tool path are tool type, tool axis mode referring to inclination angles, radial depth of tool engagement to the work piece (step-over) and maximum discretization step. Each of these attributes affects differently the tool path style and thus, the number and topology of cutter contact points. Thereby, the true challenge at this phase is how to satisfy the necessity of producing a number of tool center locations such that low chord error is yielded while simultaneously facilitating sampling time, thus; productivity.

Optimization problems introduced in manufacturing and allied engineering areas are considered to be non-linear followed by high levels of computational complexity. Consecutively, stochastic optimization strategies found in areas such as soft computing and artificial intelligence ought to be implemented. Representative systems are genetic-evolutionary algorithms (GAs-EAs), artificial neural networks (ANNs), simulated annealing (SA) algorithms and other methods that mimic physical behavior of species and natural mechanisms. Concerning optimum tool path generation noticeable contributions have been devoted to propose approaches to efficiently address the problem of machining surface error as a combined effect of chord error and scallop height [9]. Kersting and Zabel in [10] employed a multi-population, genetic algorithm to provide optimum solutions to 5-axis milling optimization problems. Ulker et al. in [11] implemented an artificial immune system to control tool path planning for sculptured surfaces with less cutter location points and reduced machining time. Zeroudi et al. in [12] implemented an intelligent system to obtain optimum tool paths in terms of cutting forces directly from CAM environment. Li et al. in [13] proposed a back propagation neural network-based multi-objective optimization strategy to optimize cutting parameters for sculptured surfaces. In their work a mathematical model is first developed with speed, feed depth of cut and step-over to be treated as independent parameters whilst machining time, energy consumption and surface roughness were the optimization objectives. Manav et al. in [14] formulated a triple-bounded surface machining optimization problem to be solved by an intelligent generalized solution involving physical relationship between mean resultant cutting forces, cycle times and scallop height magnitudes. Suggesting that these criteria are conflicting to each other, a pareto-front was produced to illustrate all possible solutions given the different weights of criteria. Fountas et al. in [15] implemented an enhanced genetic algorithm to optimize machining modeling for 3- and 5-axis multi-objective complex surface milling operations. Having defined their optimization objectives related to both productivity and part quality, they achieved automatic interaction between manufacturing software and a binary-encoded genetic algorithm for the necessary argument passing towards the convergence to optimal solutions in terms of machining time and surface deviation. Talon et al. in [16] captured the idea of implementing a non-deterministic technique with a controlled tolerance margin capable of reducing data processing times but in favor of high part accuracy. In [17] Zain et al., employed a genetic algorithm to find optimal cutting conditions for attaining minimum surface roughness in end milling. Their research involves experimental design using three different coatings for cutting tools. Three regression models were developed to predict the response and a statistical analysis indicated the suitability for one of them to be utilized as the fitness function for the GA employed. It was found that the application of GA to such problems is beneficial and capable of attaining lower values in terms of the response criteria than those obtained by experimental results; stand-alone regression modeling and response surface methodology. Face milling is another important machining operation and it was upon such a process that Raja et al.,in [18] succeeded on; by solving a multi-objective optimization problem with the use of PSO algorithm. Their work involved experimental runs using cutting conditions as the independent variables and their effect on surface quality as well as machining time was studied. A generic process model was used as the fitness equation for the iterative PSO evaluations whilst final confirmation experiments shown the validity of this algorithm. Liu et al., [19] addressed a complex process planning problem considering the selection of attributes such as production resources; operational sequence and manufacturing constraints towards the goal of optimizing the cost through a mathematical model handled by an ant colony optimization (ACO) algorithm. The process planning operation was mapped to a weighted graph and converted to a constraint traveling salesman problem. For the same optimization problem the authors implemented tabu search (TS) and simulated annealing (SA) so as to compare their results to those obtained by these algorithms; validating thus their proposed approach. Aydin et al., [20] proposed the application of an artificial immune algorithm to optimize the attributes of a support vector machine. As a classification strategy it is; support vector is optimized in their study via the multi-objective problem formulation approach considering the system's kernel and penalty parameters. In [21], Kim implemented a simulated annealing (SA) algorithm to optimize the process of rough machining operation for planar tool path creation. That is; the material volume is sliced to a discrete number of Z level passes and the tool trajectory direction is decided upon the context of minimizing cycle time. To reduce manufacturing cost regression modeling and artificial intelligence algorithms have applied also to software so as to come up with optimal tool paths that drive computer numerical control (CNC) machine tools for complex sculptured parts thus; examining the outcome prior to actual machining using software.

The non-linear characteristic that the majority of machining optimization problems suggest; may be accompanied with either discrete or continuous representations in terms of the solution space. That is; the problem may be designed to receive particular values for evaluation or may occupy any value, yet; over a continuous domain. By considering the above literature it is indicated that the generation of regression models to by utilized as fitness functions / solution ranges is an often-used strategy whilst ends up with introducing continuity. Therefore any intelligent algorithm may easily handle the problem and rapidly converge to a solution that is deemed optimal. This approach establishes an entirely different philosophy of building multi-objective manufacturing optimization problems by directly assessing outputs of discrete values reflecting quality and productivity criteria resulting from software-executed simulations. Hence, for an intelligent algorithm to be hosted and operated "in-house" by manufacturing software; it is mandatory that both discrete and combinatorial attributes are to be processed. Unfortunately not all algorithms support both problem types and this is a reason why researchers develop hybrid systems. On the other hand hybridization for tackling optimization problems suggests merits taken from individual systems constructing the hybrid under the imminent risk for an individual module to overcome to other's deficiency. Another important issue referring to the applicability of an intelligent algorithm to this work's problem would be the structure type that ought to allow bidirectional; multiple argument passing between inputs (candidate solutions) and outputs (values for quality objectives).

The heterogeneous nature of the parameter range found to most of the tool paths and the lack of explicit or even implicit criteria expressions impose the existence of an intelligent and customizable framework capable of formulating tool paths with evaluation cost and computational time. In this study an enhanced virus-evolutionary genetic algorithm is employed to compute the optimal tool trajectory for 5-axis machining operations in terms of chordal deviation and scallop height. The aim is to obtain the optimum values for machining parameters including inclination angles, tool type, discretization step and tool path interval by considering sculptured surface properties. In essence, the number of CL data is to be minimized without deteriorating the pre-determined cutting tolerance, thus; ending up with tool paths with low, uniform errors and of short cutting time. Computational operations involving surface curvature analysis and vector algebra has been fully automated through the development of visual basic macros and functions by taking advantage of the application program interface (API) that CAM environment offers. As a result, computational time and cost are dramatically reduced to a considerable fraction. It contributes to the field of complex surface machining by proposing a novel philosophy of CNC programming with automation and optimization benefits gained in a common intelligent machining modeling platform capable of processing heterogeneous parameters with good exploitation and exploration rates via the viral operators of the new enhanced virus-evolutionary genetic algorithm. This algorithm generates robust artificial schemata representing candidate solutions for machining parameters and it has been developed according the main structure suggested in [22]. The notion is extended further to maintain an acceptable processing ratio between the two populations the algorithm handles; sustaining this way low computational time. The algorithm integrated the programming development platform (API) of a cutting-edge CAD/CAM/CAE system (*Dassault systèmes* CATIA® V5) to formulate a common manufacturing modeling framework for tool path planning and NC programming and it may also exist as a stand-alone intelligent computational system to facilitate problem solving in engineering areas similar to the one investigated in this study.

2 Research problem and contribution

Tool path optimal planning is a well-established field for CNC sculptured surface machining, for which numerous effective algorithms have been developed whilst some of them are even implemented in commercial software systems. Most of the models and algorithms available for tool path planning are heavily based on deterministically formulated criteria evaluated by process models and predictive mathematical equations. Even highly sophisticated such a research could be considered; a partial problem-solving danger appears as well as queries concerning their optimal solutions and direct implementation to actual industrial operations. For the majority of studies involving tool path planning it is not clear to what extend their key attributes provide integrated systems capable enough to support manufacturing optimization and by no means suggest intelligence; only technical computing approaches to facilitate algorithmic evaluations.

The proposed work presented in this paper differentiates from others related to the field, not only in the part of global investigation via the introduction of all influential machining parameters avoiding marginal merits and partial problem investigation, but proposes an integrated simulation-based machining optimization platform providing a direct "ad-hoc" between computer-aided manufacturing (CAM) software and advanced evolutionary algorithms as well. The way of examining the solution domain both entirely and efficiently is facilitated by the intelligent algorithm proposed that undertakes the heterogeneous machining parameters processing. Thus, new insight is revealed and novel investigation directions could initialized with the proposal of this study as a reference; to facilitate the state-of-the-art research.

3 Problem definition and related attributes

The problem in this study is defined as the effort to produce tool paths capable of maintaining surface machining error to acceptable preset tolerance levels via a stochastic fashion with the least possible number of cutting data (CL points) to reduce processing time. Surface machining error is mainly a consequence of chordal deviation (or chord error) and scallop height. The former refers to the maximum distance between a chord whose connecting points lie on the original curve and a point on this curve (Fig. 1a) whereas the latter is the well-known remaining material magnitude among successive tool pass intervals. The chords that a tool's center locations produced represent the interpolated contour regarding the predetermined tolerance (Fig. 1b).

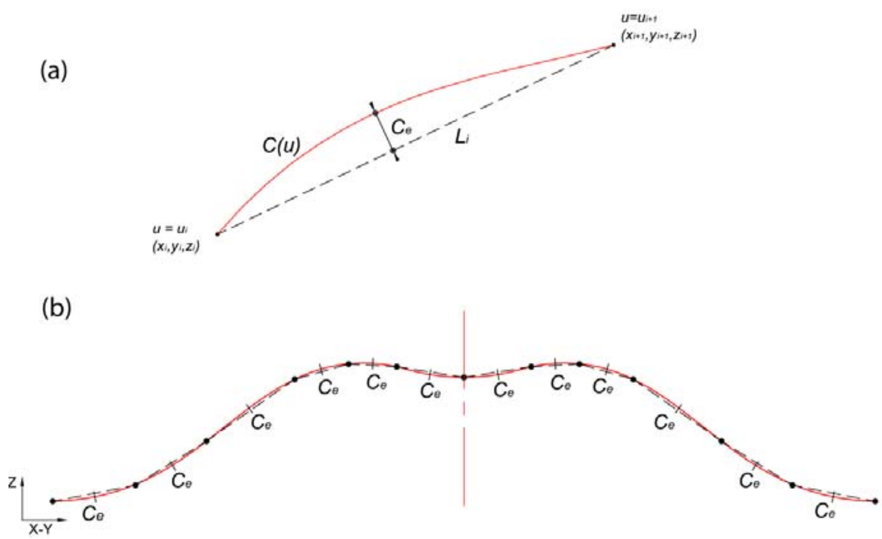


Fig. 1 (a) Chord error illustration given the step; (b) discretization steps according to tolerance

5-axis milling technology facilitates surface machining operations owing to the ability of varying the tool axis orientation and enhanced part quality results. When it comes to 5-axis machining, chord error and scallop are greatly affected by essential process parameters such as the tool type and its corresponding inclination angles (lead and tilt) as well as the two steps controlling surface segmentation and scallop. Three tool types were used for the proposed methodology's experiments, flat-end, ball-end and filleted-end mills. The decision about investigating these three major tool types was made in order to consider the variations of CL-points computation by taking into account their topologies in terms of the studied surface related to the tool's center position that changes regarding its geometry. Such variations would suggest different outputs for both chord error and scallop [8].

Inclination angles referring to lead and tilt were investigated so that collisions between the machining setup entities are avoided whilst producing sufficient cutting data at the same time. Hence, a specific interval was determined for their variations according the suggestion made in Chen et al. [23] and this interval is [-45*deg* to 45*deg*]. The rationale behind their inclusion to the study is their 'indirect" influence on scallop formulation through the inclined tool's effective radius. Indeed, the projection of an inclined tool's bottom-end on the machining surface is an elliptical silhouette which depends on the inclination angles a tool holds for a given CC-point and as a result the effective radius corresponded to the angles controls the magnitude of the ellipse's minor axis affecting the scallop. Fig. 2a depicts the relation between effective radii and resulting elliptical postures and Fig. 2b shows how scallop magnitude varies under different effective radii for inclined tools.

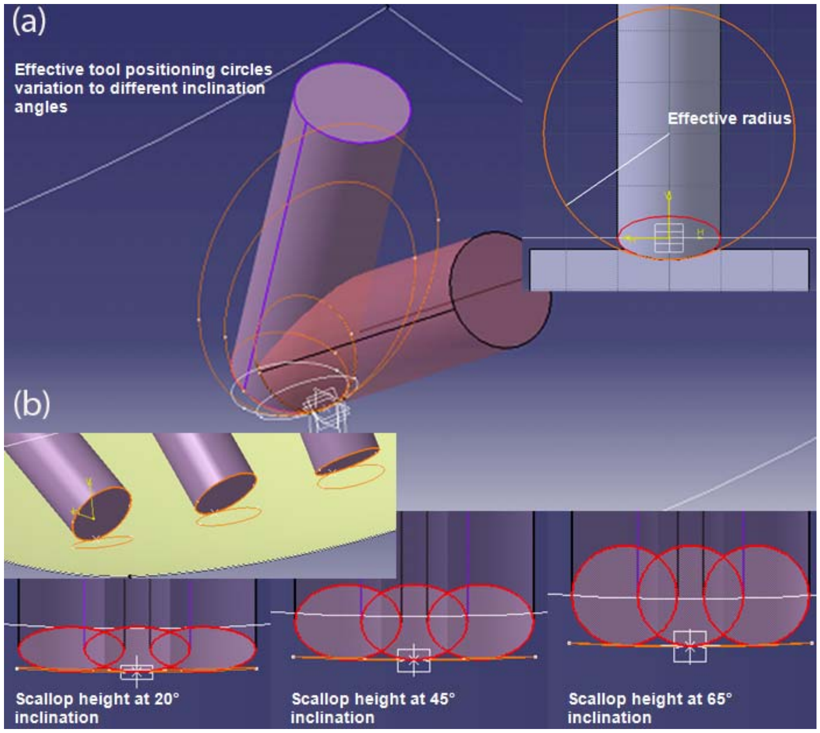


Fig. 2 (a) Elliptical projections under different effective radii from tool inclination; (b) Scallop dependence with inclination angle

At an early stage, the focus of the proposed methodology is given to the computational model environment preparation. The computational model environment involves the automated solving of certain equations to repetitively extract values for the objectives presented above. Material removal simulations supported by commercial CAM systems do not provide properties to directly analyze the deviation between the original and the machined part; only point information regarding the local region of the studied surface and a point's distance in the Cartesian or parametric space. As far as scallop among successive pass intervals is concerned, it is not expressed via the approach adopting effective radius and its direct computation by CAM interface is not supported by current software tool palettes. A strong benefit of computing effective radii for different tool inclinations, is the manipulation of control axis mode so that, not only collision-free tool paths will be generated, but paving the way also for stochastic specifications concerning tool pass intervals to facilitate surface machining error. Chord error computation imposes the need of implementing polynomials for curve fitting with reference to the coordinates of a tool path's CC-points. Local curvatures with respect to the feed direction need to be computed further by employing vector algebra for estimating the angle between pairs of lines (vectors) normal to local surface regions by retrieving their dot products. An important prerequisite for chord error computation is the length of each chord connecting two consecutive CC points. Chord lengths were computed by utilizing all coordinates of CC-points in a sequential manner with reference to the machining axis system given in the Cartesian space. Eq. 1 was adopted to compute the sequential chords and respective chord errors were computed via the mathematical formula given in Eq. 2.

$$L_i = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 + (z_{i+1} - z_i)^2} \qquad (1)$$

$$C_e = \rho_i - \sqrt{\rho_i^{\,2} - \left(\frac{L_i}{2}\right)^2} \qquad (2)$$

Effective radius was computed by employing an equation which is a product of analytical geometric approximations presented in Redonnet et al. [24] and tested in several numerical experiments presented earlier in [25]. This equation was thus employed to support the proposed computational model environment and is as follows:

$$R_{eff} = \frac{(R-r)\cos^2(a_T)}{\sin(a_L)\left(1-\sin^2(a_T)\sin^2(a_L)\right)} + r \qquad (3)$$

where:

R: the cutter's radius;
r: the cutter's corner radius (for flat-end mills: r = 0);
a_L: the cutter's lead inclination angle in degrees;
a_T: the cutter's tilt inclination angle in degrees.

Note that Eq. 3 is applicable for both filleted-end and flat-end mills, yet; it is not applied in the case of ball-end mills since their location positions of their center points are independent on tool axis orientations owing to their spherical geometry. In this case CAM software computes the scallop via its relation to the given step-over value (radial depth of cut). To avoid time-consuming and lengthy computations among mathematical expressions, the proposed intelligent algorithm was adopted to handle component object modeling attributes exploited during the investigation of the application program interface.

The physical consequence of iteratively handling the above presented manufacturing entities via the proposed intelligent system, is the stochastic control of parameters and gradual convergence towards a set of values, such; that will optimize the machining surface error as a combined effect of chord error and scallop height.

4 Virus-evolutionary genetic algorithm: Structure, development and implementation

A large variety of heuristics and intelligent computational models have already proposed for manufacturing engineering optimization. The reason that forces researchers to suggest them is the problem of premature convergence [26] owing to the inherent domination of outstanding solutions that take over the population of candidates. Premature convergence is the outcome of poor diversity that results to local trapping. The proposition of a virus-evolutionary genetic algorithm to be implemented for sculptured surface tool path optimization stems from the need to overcome the premature convergence problem by storing solution patterns during the search for global optimum. The ability of the virus-evolutionary genetic algorithm to sustain the optimization process lays on its two simultaneously-processed populations; that of hosts and that of viruses. Hosts represent the candidate solutions to one or more objectives response as it is suggested in most of heuristics and artificial intelligent algorithms, whereas virus population - which is actually produced by host population – illustrates the effective solution schemata being preserved during the search. What really gives this algorithm's its edge is the application of viral operators capable of simulating two major natural mechanisms found in the evolutionary procedure of viruses [27].

Viral operators: Transduction and Reverse transcription

When a generation of hosts is created, viruses as effective patterns offer information to produce better offspring. The algorithm performs the first viral operator to some of the best fitted individuals after evaluating their fitness function whereas the rest of the viruses are created by randomly selecting from the rest of the hosts. The respective operator simulates the ability of natural viruses to modify or even multiply their genomes after having infected their hosts; known as "*transduction*" procedure. The authors utilized such natural intelligence to artificially produce a certain number of viruses as a fraction of the host population number and specifically 1/10. This technically implies that the number of solution patterns to store, depends on the number of candidate solutions. Therefore a good probability of tracking and preserving outstanding seed to favor next generations exists, yet; with no bias owing to its small magnitude. What is depicted in Fig. 3 is the process of transducing segments of a set of individuals' main strings from a population of 50 hosts to create a population of 5 viruses as an example of the proposed approach. Substrings been cut, represent a viruses' chromosomes whose length is denoted as $VirusString$. Locus 1 is the starting point from which $VirusString$ length will be determined and Locus 2 is the ending point. These two limits are randomly selected and are constrained to the original host's chromosome length. Note that viruses have been programmed to be created by selecting some of the elite individuals and some randomly selected from the rest ones to infect.

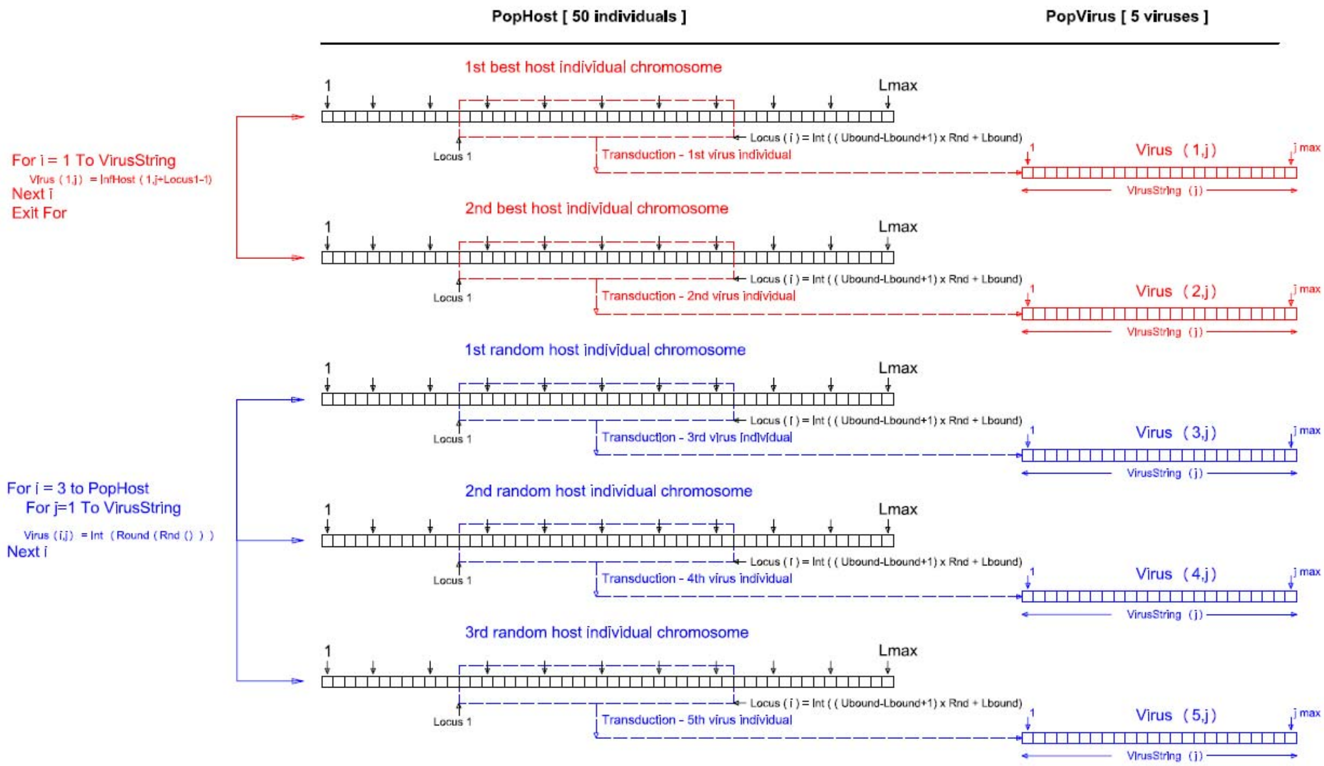


Fig. 3 Virus population generation using transduction

Once the two populations have been initialized, viruses attack to infect host individuals, which is the main task of the second viral operator named as "*reverse transcription*" operator. In computational terms this implies the functionality the specific operator enables for overwriting a virus' string to a host's string and is programmatically achieved by sequentially typing each virus' binary string digit to the chosen host's binary string digit according to the determined locus for replacement. To do so, the indices of both the virus and the host together with their replacement loci for the strings are declared in advance. As a result a new host, (infected host), is generated.

Viral operators are dynamically collaborated under the major goal of increasing more effective individual representation schemata that will rapidly lead to the global optimum solution for a given problem. Fig. 4 depicts the operational sequence for *reverse transcription* and *transduction* operators towards their virus infection procedure.

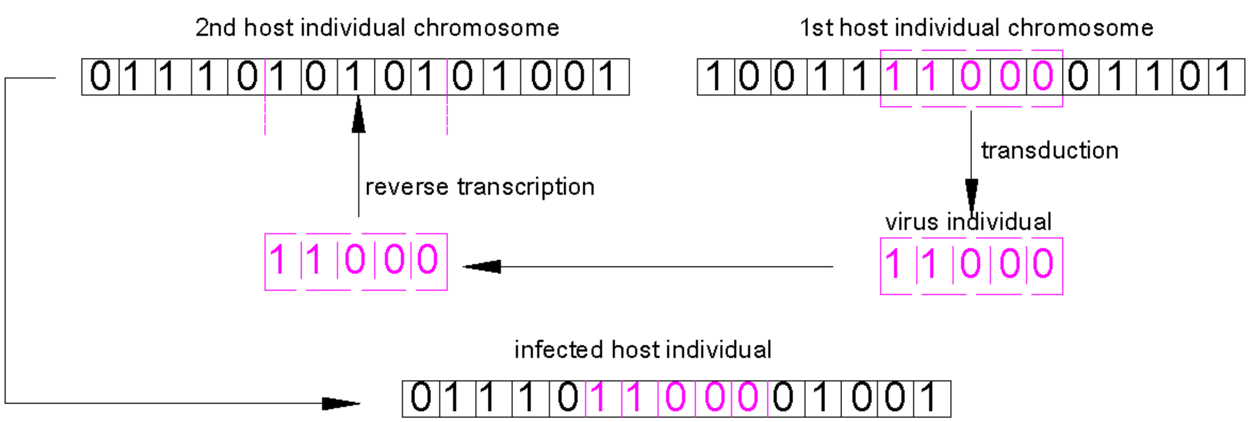


Fig. 4 Generation of a new host individual through virus infection operators

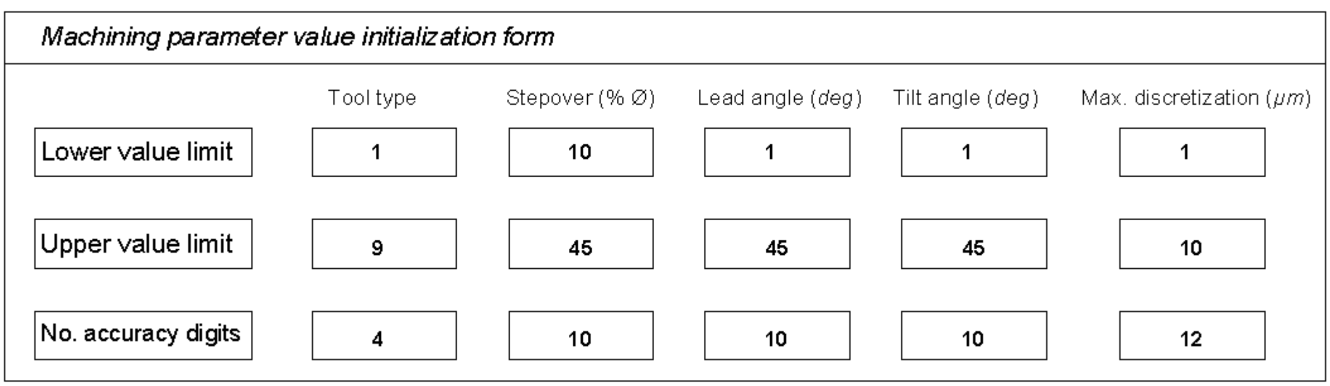
Several assessment metrics have also been programmed so as to judge infectivity and its positive impact to the optimization problem's response. These metrics involve indices for assessing virus fitness, infection rate and infection capability. Virus fitness is a magnitude that every virus holds and reflects its efficiency. Infection rate controls the number of infections a specific virus individual is capable to perform. Infection rate is related to virus fitness. Should a virus enhance its fitness in terms of infectivity; its infection rate is increased. On the contrary if virus fitness is degraded the corresponding infection rate is reduced. Information about the representation of these assessment metrics and their programming philosophy can be found in [22], [28] as well as in [29]. The rest of the algorithm's structure involves the chromosome representation, the conventional genetic operators (selection, crossover, mutation) and fitness function evaluation. Workflow and functionality is given to the following section.

Algorithm structure and computation workflow

The evolutionary algorithm proposed in this study, starts with the construction of the initial chromosome represented adopting binary encoding scheme. For the chromosome three matrices have been developed to depict the number of chromosomes; their number of digits and their locations [30]. To achieve decision variables assignment to CAM software, chromosomes were converted to their phenotype representation. Thus; considering a given variable $Var(i)$, its studying value domain $d_i = [u_i, l_i]$ and the corresponding chromosome's length $length(i)$ of binary digits, Eq. 4 was adopted to perform such conversion.

$$Var(i) = l_i + function(binStr) * \frac{u_i - l_i}{2^{length(i)} - 1} \tag{4}$$

Note that $function(binStr)$ is a function developed to return the corresponding decimal value for a given binary-encoded value considering the variables' accuracy requirements [30]. By taking into account the 5-axis machining parameters involved to the optimization problem, an interaction form was built to provide the initialization interface. Thereby, this form prompts users to select the range of machining parameters to be examined and is depicted in Fig. 5.


Machining parameter value initialization form

	Tool type	Stepover (% Ø)	Lead angle (*deg*)	Tilt angle (*deg*)	Max. discretization (*μm*)
Lower value limit	1	10	1	1	1
Upper value limit	9	45	45	45	10
No. accuracy digits	4	10	10	10	12

Fig. 5 Initialization form for machining parameters' chromosome representation

The values range can be taken from an ordinary cutting tool catalogue as it is done in conventional tool path planning approach. Accuracy digits is decided according the precision of a parameter in terms of its type (integer, double, etc) and its units. In the example of Fig. 5, the tool type which is an integer value does not require so much as high precision therefore it is set to a lower value comparing the rest of the inputs in the initialization form.

A function was formulated to measure the fitness of initial individuals (candidate solutions) so as to decide upon their ability to survive and reproduce in next generations. The function's exponential expression separates best candidates from others less fitted, even if small differences in terms of their values emerge. The fitness expression involves the summation of ranked objective values of individuals, dictated as $FitSum$, whilst $ObjVal(i)$ refers to the objective value of i_{th} individual. Ranking is performed prior to the fitness function computation to rearrange individuals in an ascending order based on their objective values in the problem. The formulated fitness expression is given in Eq. 5.

$$Fitted(i) = FitSum * [\exp(-ObjVal(i)^2)] \qquad (5)$$

The proposed system utilized “stochastic sampling with partial replacement” [30] as the selection scheme to select individuals for mating. This particular scheme suggests that selected individuals are pointed out by a random number generator. Each time a certain individual is selected, the probability for its re-selection reduces so that bests do not always dominate, thus; maintaining good elitistic behavior and avoiding local trapping. The expression to programmatically prompt to this action is given in Eq. 6.

$$fitted(j) = fitted(j) - [FitSum(i) / TotalNumOfIndivls] \qquad (6)$$

Crossover is achieved between two individuals towards the production of two offspring with 100% probability. As a consequence a new population results as twice the size of the previous one. Mutation is also performed in the proposed system for a randomly selected number of individuals by adjusting its rate to be limited to 10% so as to operate under efficient exploration [31].

The main structure along with the viral operators constitutes the virus-evolutionary algorithm proposed in this work for tool path optimization. Its operational workflow is illustrated in Fig. 6.

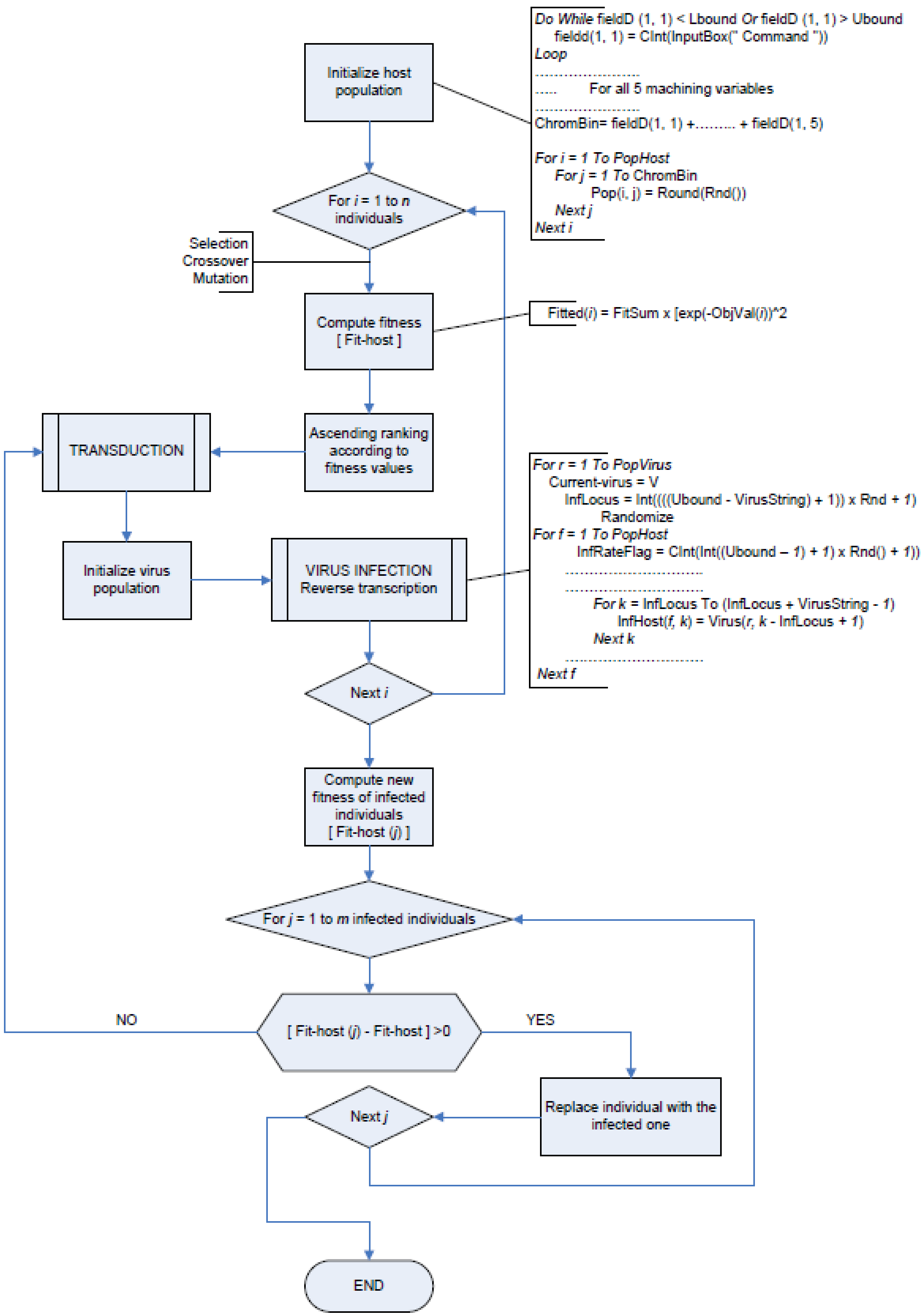


Fig. 6 Virus-evolutionary genetic algorithm computational workflow

Manufacturing software integration and system implementation

The proposed algorithm operates as a module embedded to CAM software application program interface (API). Its evolutionary function determines the total number of generations for individuals which is also the termination criterion. A “log” file either in **.txt* or **.dat* format is utilized to store individuals (candidate solutions) logged along with their objective values during their subsequent evaluations. The evolutionary function of the proposed algorithm named as “*Evolution*” interacts with CAM via the main manufacturing automation module handling programmed functions responsible for machining parameter retrieval, tool path computation / automatic program tool (APT) code creation and objective value computation. The declaration name of the main manufacturing automation module is “*Evaluate_5xSSM*” referring to the process of 5-axis sculptured surface machining (SSM) evaluation.

The algorithm is initialized via the machining parameters’ user form and passes the phenotype values of generated candidate solutions to the already opened machining modeling session of CAM. A secondary function also handled by “*Evolution*” checks whether the values selected fall in the variables’ applicable range. In the case that the pair of selections does not so much as to obtain applicable outputs, a penalized value is returned as the objective value for the corresponding candidate solution which is stored to the “log” file. Should the same pair of values is selected again by change; computations are not performed owing to the algorithm’s scouting mechanism to “log” file for previously assessed parameter values. Thus computational time and cost are dramatically reduced.

At this point, the manufacturing automation module undertakes to scan the respective machining project tree, retrieve the specific tool path strategy the user has applied and access its parameters referring to tool type, step-over, lead and tilt angles and maximum discretization step. Once the inputs for these parameters have automatically set by the algorithm, CAM software executes the tool path computation for that scenario, generates the APT source file (CL data) and computes the coordinates of cutter contact points (CC-points). The coordinates of these points are imported to the original sculptured surface and outputs concerning chord error computation and effective radius for tools are stored to formulate the objective function for minimization.

The proposed surface tool path optimization methodology is fully automated following the hierarchy and organization of machining project documents. Optimality of solutions is considered according preset constraints involving technological resources. As a result different outputs may be obtained considering power demand for given machine tools, cutting tool properties, work piece material and tool path time restrictions. The proposed system embodies such constraints to reflect actual industrial operations and span practical validity. An overview of the operational process between the virus-evolutionary GA and CAM system is given in Fig. 7.

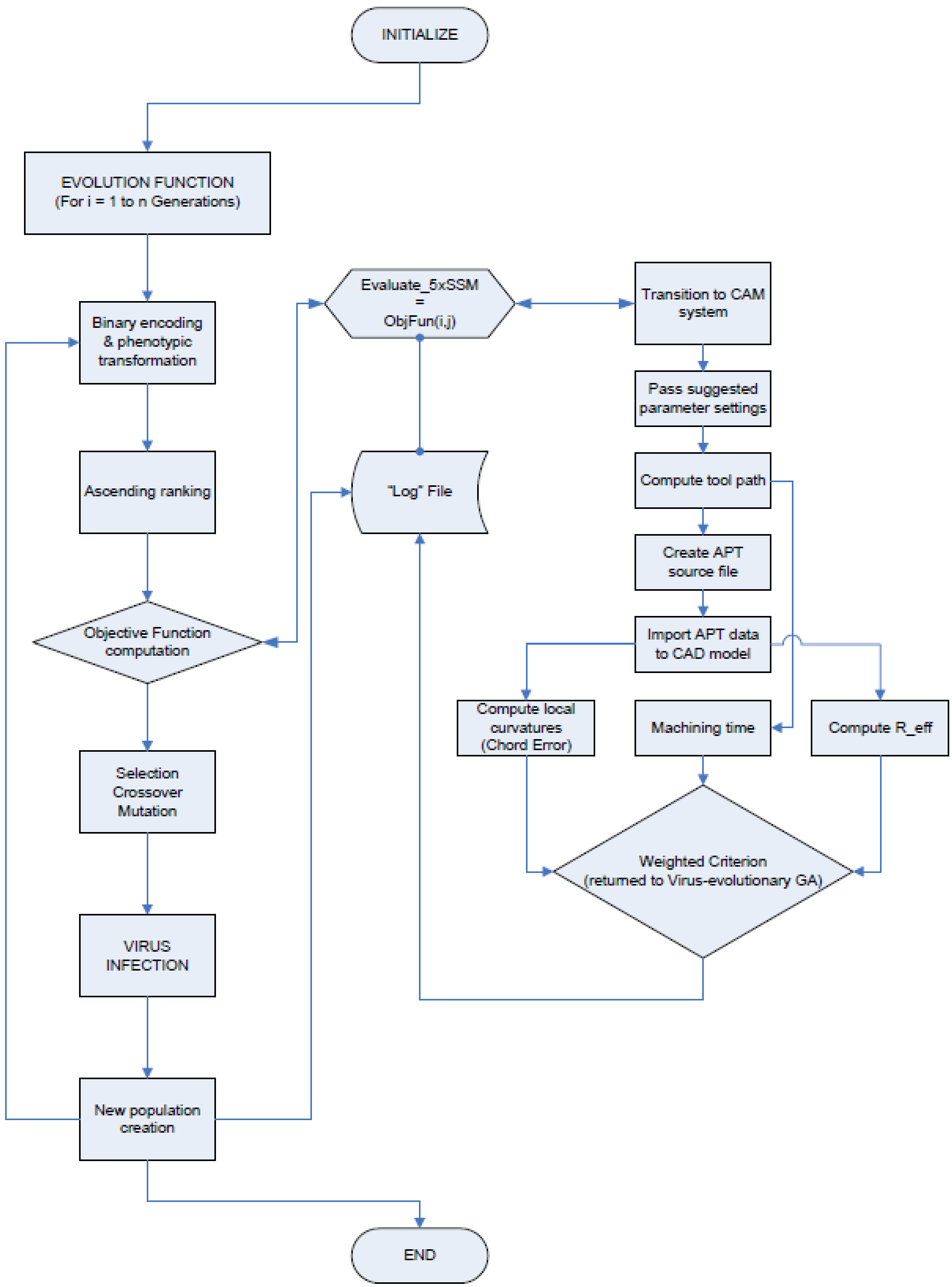


Fig. 7 Operational procedure between CAM software and virus-evolutionary genetic algorithm

The aforementioned methodology is deployed during the process planning stage towards the formulation of final CNC programs that will drive the machine tools. Prerequisites for the methodology to be used in actual industrial conditions involve the existence of the manufacturing process document that ought to be prepared in advance. This document should attach the machining strategy as well as indicative settings for cutting parameters. Thereby the methodology can be initialized using the user form been mentioned above; dealing with the initialization for the parameters involved to the optimization process. Minimum and maximum levels shout be determined by the end-user so at to formulate the solution domain; the space of which is to be exploited by the intelligent virus-evolutionary algorithm on its way to converge to the optimal solution.

5 Experimental results and discussion

The presented optimization methodology was implemented to a series of machining experiments adopting a standard 5-axis swept cutting tool path for a test model (Fig. 8) with a NURBS sculptured surface. The model was designed using the "*free-style*" workbench of the commercial CAD/CAM system *Dassault Systèmes CATIA® V5* whereas tool path planning was conducted in the same system utilizing the "*advanced machining*" workbench. To validate the proposed approach and show that is prominent, half of the experiments were conducted by applying a conventional GA and respective results compared with those obtained by the virus-evolutionary GA.

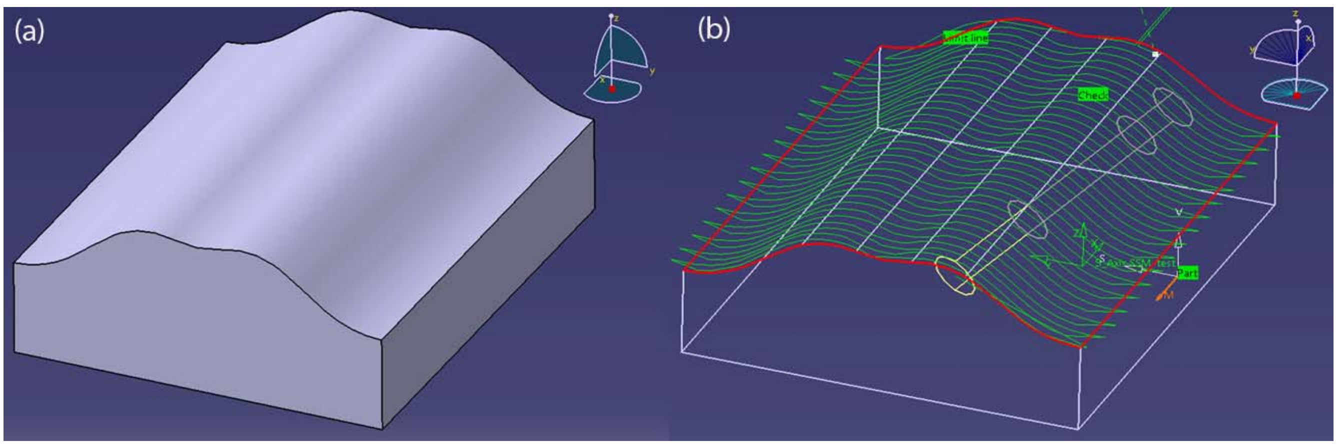


Fig. 8 Experimental work piece (a) the model with the NURBS surface; (b) 5-axis swept cutting style

Both series of experiments were defined with reference to five different machining tolerance restrictions being set from 0.01mm (min) to 0.05mm (max) and a tool database containing nine tools was established so as to subject the two competitive algorithms to their selections towards finding the optimum ones given the evaluation scenarios. The tool database categorizes the selected tools to three groups from which the first involves flat-end mills; the second filleted-end mills and the third ball-end mills. The tool database is illustrated in table 1.

Table 1 A categorized tool database according the type and geometrical configurations

Tool ID	*Name*	*Type*	*Cutting Diameter Ø (mm)*	*Corner radius Rc(mm)*
#1	Mfg_FlatEndMill_06	Flat-end mill	6	0
#2	Mfg_FlatEndMill_08	Flat-end mill	8	0
#3	Mfg_FlatEndMill_10	Flat-end mill	10	0
#4	Mfg_FilletedEndMill_06	Filleted-end mill	6	1.5
#5	Mfg_FilletedEndMill_08	Filleted-end mill	8	2
#6	Mfg_FilletedEndMill _10	Filleted-end mill	10	3
#7	Mfg_BallEndMill_06	Ball-end mill	10	3
#8	Mfg_BallEndMill_08	Ball-end mill	10	4
#9	Mfg_BallEndMill_10	Ball-end mill	10	5

Operational ranges for the machining parameters referring to all experimental runs were -45*deg* (min.) to 45*deg* (max.) for inclination angles under variable tool axis mode and 10% of tool diameter percentage (min.) to 45% of tool diameter percentage (max.) for step-over parameter. The lower range limit for maximum discretization step parameter was fixed to $1x10^{-3}$ *mm* and the maximum one varied in ascending order according to the preset machining tolerance (from 0.01*mm* to 0.05*mm*), thus it was programmed to take values from $1x10^{-2}$ *mm* to $5x10^{-2}$ *mm*.

To introduce a competitive trade-off among quality and productivity reflecting thus actual industrial status, machining time from CAM system was also considered as a minimization target together with mean chord error value and effective radius. Note that machining time (*minutes*) has different magnitude to that of chord error and effective radius for tools (*mm*) hence, proper normalization was done following trivial techniques [32]. Thereby, a double-bounded problem having the contradictory targets of machining surface error and machining time was addressed. A weighting algorithm was applied at this stage to allow criteria to be compensated through weights prioritizing their influence with the simultaneous simplification to a single objective expression which is depicted as follows:

$$f\left(\operatorname*{var}_{i=1}^{x}\right) = \min Objfun\left(Mach\_Err \times w_k, t_m \times w_{k+1}\right) \tag{6}$$

where,

$f\left(\operatorname*{var}_{i=1}^{x}\right)$: the individual objective function for each independent variable;

$Objfun$: the total objective function corresponding to the criteria of machining surface error $Mach\_Err$ and machining time t_m

w_k , w_{k+1}: the weights for criteria optimization importance

The total number of CL points was also examined as a performance metric for the problem. Results obtained from experiments of both optimization algorithms along with the rest of the aforementioned attributes are given in table 2.

Table 2 Series of evaluation experiments and results obtained to corresponding performance metrics for virus-evolutionary GA and conventional GA

Virus-evolutionary GA		Machining parameters					Performance metrics			
No. Exp.	Cut tolerance	Tool ID	a_e (%Ø)	Lead an. (*deg*)	Tilt an. (*deg*)	Max. discr. step ($x10^{-3}$)	Mean C_e (*mm*)	R_eff (*mm*)	No.CLs	T_m (*min*)
1	0.01	#6	34.307	3.208	7.079	8.339	0.00116	38.199	2203	1.37
2	0.02	#3	36.724	4.68	11.749	4.928	0.00366	58.757	1418	1.28
3	0.03	#5	26.714	3.484	6.555	8.153	0.00351	34.484	1893	2.09
4	0.04	#1	20.635	7.375	23.266	4.794	0.00383	19.776	1618	3.53
5	0.05	#3	22.148	3.133	22.703	5.103	0.00409	77.892	814	1.98
Genetic algorithm		Machining parameters					Performance metrics			
No. Exp.	Cut tolerance	Tool ID	a_e (%Ø)	Lead an. (*deg*)	Tilt an. (*deg*)	Max. discr. step ($x10^{-3}$)	Mean C_e (*mm*)	R_eff (*mm*)	No.CLs	T_m (*min*)
1	0.01	#2	36.968	14.594	5.383	3.711	0.00116	15.744	2766	1.78
2	0.02	#3	29.814	8.834	20.992	4.928	0.00367	28.466	1736	1.57
3	0.03	#5	26.714	15.168	8.529	3.713	0.00369	9.487	1988	2.20
4	0.04	#1	21.025	9.32	6.18	5.709	0.00437	18.315	1683	3.52
5	0.05	#2	22.148	3.133	22.703	5.103	0.00443	62.313	936	2.10

Based on the results obtained both algorithms attained values for machining parameters to optimize the quality objectives with the virus-evolutionary GA to outperform the conventional GA in terms of the majority of selections. Significant differences appear to those parameters that mostly affect the specified objectives such as inclination angles for the tool axis mode and maximum discretization step for the curve segmentation control. Most of the selections made for tools by both algorithms, were identical whereas values attained for the control of radial cut via step over parameter were of minor difference.

For the series of experimental runs conducted according the tolerance specification, the proposed algorithm managed to further reduce the fitness function in comparison to the conventional GA. As far as the tighter tolerance specifications determined (0.01*mm* and 0.02*mm*) both systems reached the same value for mean chord error, yet; by differentiating to the rest of the selections virus-evolutionary GA reduced the number of CL data and machining time for the same tool path. Of particular interest concerning the first experiments with 0.01 mm tolerance, was the different selection of tools between virus-evolutionary GA and conventional GA. The former system which is the one proposed, deemed the Ø10*mm* filleted-end mill (tool ID #6) as the optimum selection whereas the latter ended up with selecting the Ø8mm flat-end mill (tool ID #2) from the tool database. It can be seen in advance that a trade-off among filleted-end mills and flat-end mills exists and this can explain to a noticeable extend the argument among several researchers in favor of proposing either filleted or flat-end mills for 5-axis precision cutting. The effective radius according the tool geometry and inclination angles for 0.01*mm* tolerance experiments was found far more beneficial when employing virus-evolutionary GA against the application of conventional GA.

The rest of the results obtained from the series of experiments can be similarly interpreted, suggesting that the proposed virus-evolutionary GA can rapidly exploit and explore local regions and the entire search domain respectively for optimal solutions. This implies that the proposed system not only reaches lower fitness values but it can also self-adaptively change its searching ratio to promote local and global search depending on the problem's particular character. It is worth mentioning that, with the last experiments (0.05*mm* tolerance requirement) as a reference, a large group of proposed solution sets for both algorithms indicated tool ID #2 (Ø8 flat-end mill) as the best tool whilst varying the rest parameter values. This was found by scanning through the respective log files. Yet; for the proposed algorithm the last solution set indicated tool ID #3 which is the Ø10 flat-end mill. By considering the fact that large cutting diameters shorten machining time, reduce tool center location data and produce lower scallop heights among pass intervals as the literature indicates, it can be supported that the proposed algorithm manages to escape such near-optimal solutions by finally reaching the global optimum one. Fig. 9a to Fig. 9d illustrate the general tendency of the proposed algorithm to further optimize the quality objectives compared to the conventional algorithm's capabilities.

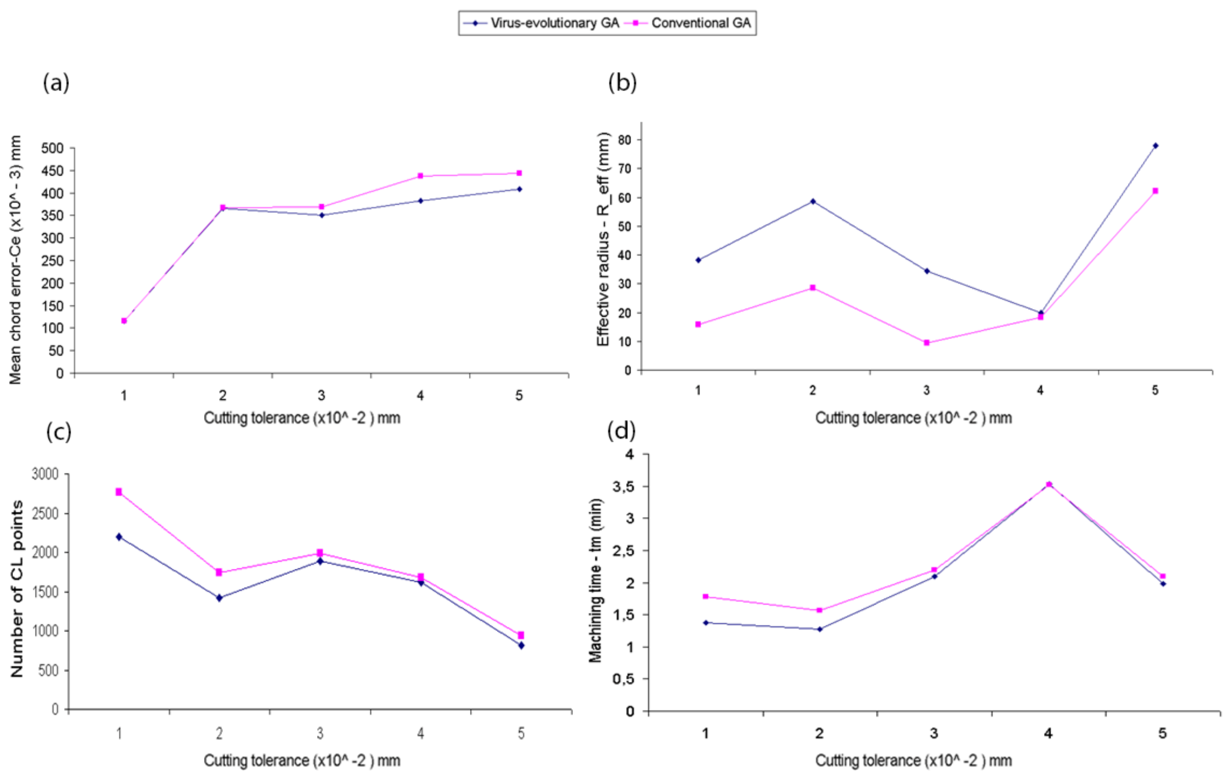


Fig. 9 Experimental analysis of results in terms of quality objectives optimization by applying both virus-evolutionary GA and conventional GA: (a) mean chord error reached with respect to cut tolerance; (b) effective radius computed for tools with respect to cut tolerance; (c) Number of CL data generated with respect to cut tolerance; (d) machining time for tool path with respect to cut tolerance

As the special emphasis was given to first experiments having 0.01*mm* cut tolerance requirements, the settings proposed by both algorithms were utilized to formulate the corresponding tool paths and NC codes. NC codes were generated via post-processing and verified for their consistency using the commercially available *CG-Tech Vericut*® *7.3* NC verification software. The same environment was also employed to examine the contouring accuracy. Surface analysis was conducted by utilizing *Auto-DIFF*® module of *CG-Tech Vericut*®. Contouring profiles for assessing the two NC codes were represented by an indicator based on remaining material after finishing known as *excess error* and machining time previously extracted from CAM software was also validated with the time reported from NC verification and found identical. Reports created after the NC code analysis indicated that virus-evolutionary GA simulated a cutting operation attaining 4.59 units of maximum excess with a total of 585 excesses. Conventional GA gave a simulated solid with 7.38 excess error units whereas the number of these excesses was equal to 1329. Fig. 10a shows the part with its stock defined; Fig. 10b the virtual setup for machining simulation; Fig. 10c the simulated cut solid that virus-evolutionary GA produced; Fig. 10d the excess error region for the model that virus-evolutionary GA produced; Fig. 10e the simulated cut solid that conventional GA produced and finally Fig. 10f the excess error region for the model that conventional GA produced.

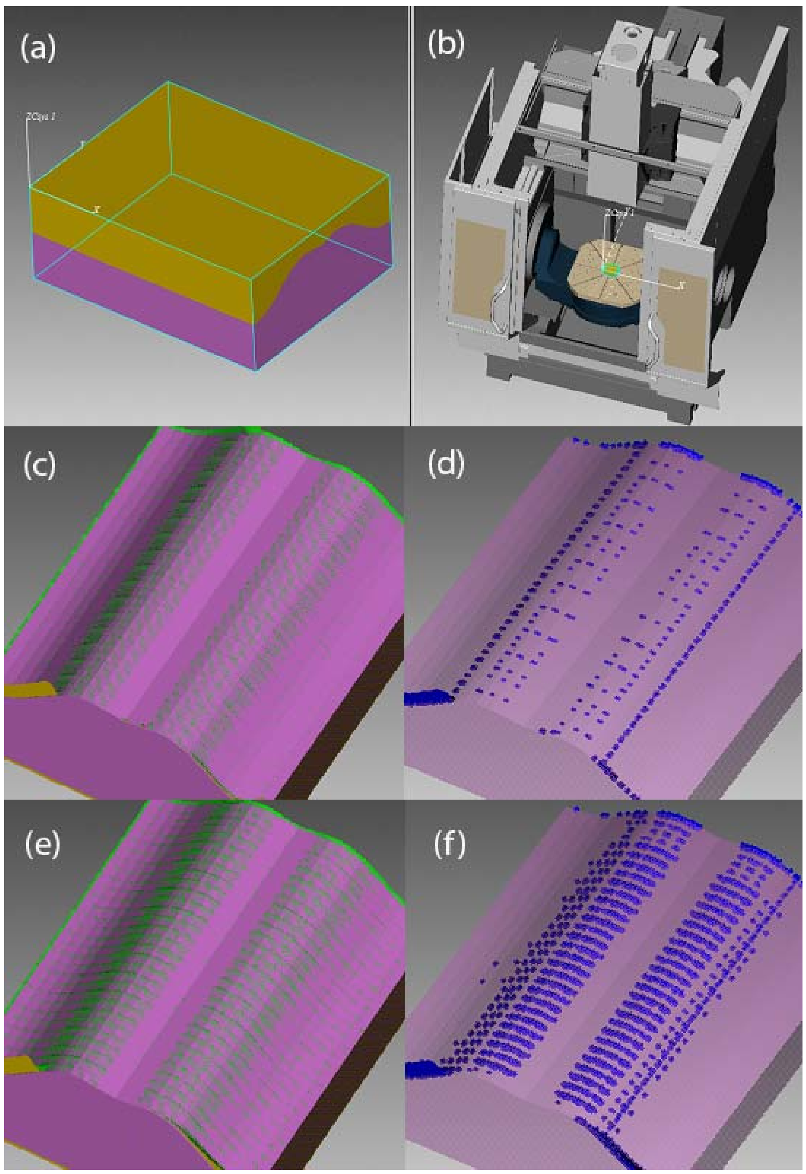


Fig. 10 NC verification conducted in *CG-Tech Vericut*®: (a) the part with its stock defined; (b) virtual machining setup; (c) simulated cut solid of virus-evolutionary GA; (d) excess error region for the virus-evolutionary GA model; (e) simulated cut solid of conventional GA; (f) excess error region for the conventional GA model

The part was machined in a *Deckel Maho MH-600C* 5-axis machine tool equipped with a *TNC-320 Heidenhain*® NC unit. A *45x79.5x100* (*mm*) *Al-5052* stock was selected to cut the NURBS test surface used for the series of experiments. A typical rough-cut tool path was first applied to prepare the part, following by the swept-style tool path presented above for finishing. The tool path applied was generated by adopting the optimum machining parameter set of the first experiment (0.01*mm* cut tolerance) that the proposed algorithm suggested. Feed rate - V_f was set to 2400 *mm/min*, rotational speed - *S* was set to 5000 *rpm.* The tool used was a solid carbide Ø10 filleted-end mill from *Jabro*®- *HPM* series of *SECO*®. Fig. 11 shows the machined part by implementing the proposed parameter values.

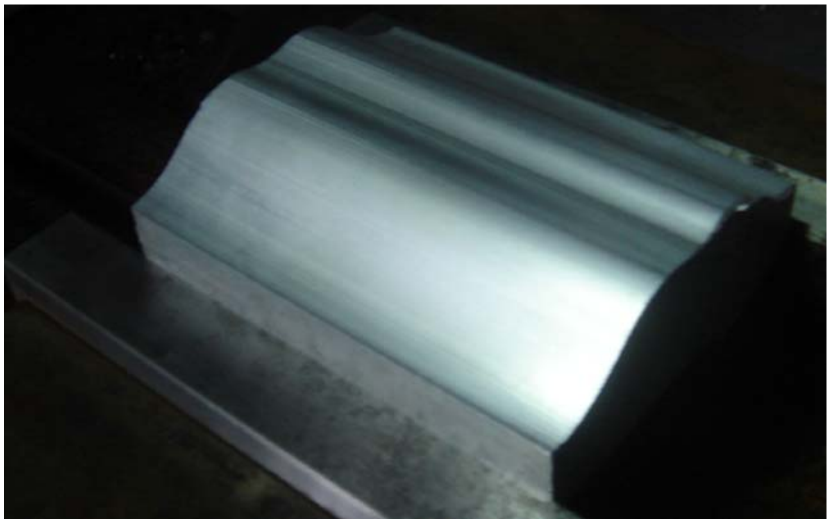


Fig. 11 The test part with the NURBS surface machined using the proposed parameter values

It can be concluded that the proposed intelligent optimization infrastructure embedded in CAM is capable of attaining optimum machining parameter values that will satisfy trade-offs between quality and productivity performance metrics. The proposed intelligent algorithm can thus handle control of tool axis mode, tool selection, proper setting for step size and pass intervals eliminating the pronounced "faceted result". Consecutively the method for optimizing surface machining tool paths proposed in this work can indeed contribute to industrial praxis and the improvements warranted are significant enough to further enhance the advanced machining modeling process.

6 Computational cost and analysis

The series of experiments were run on an *AMD Athlon*$^{(tm)}$ 64 processor, 3200+, 2.00 GHz and 1.00 GB RAM supported by *WinXP Pro*., *SP3* 2002 version. Even though automation functions which support CAM add to the computational cost, graphics card type and specs are of minor importance owing to the algorithm's "ghost" mode in terms of tool path generation. Computation time of virus-evolutionary GA spans between 4 to 6 hours depending on the cutting tolerance requirements that determine the applicable range for machining parameter settings. 87-94% of objective function computational time depends on CAM system's material removal simulation needing approximately 20-25*min* CPU time for each candidate solution. The aforementioned attributes can significantly vary through modifications in terms of the algorithm's settings referring to the adjustment of viral operators. These issues referring to the time and cost needed for the methodology to deliver results; is acknowledged as the major limitation of the research proposed.

To encourage industrials as well as preventing criticism, it is mentioned that the proposed methodology implemented on a low-tech PC. It is believed by the authors that, implementing the proposed approach in a cutting-edge system would reduce computational time about 4 to 5 times, whereas it could be further cut down similarly owing to the progress of new CPU technology. Nevertheless a running duration of 4 to 6 hours (less than one shift -8*h*-) is currently realistic; and a fraction of that is considered both appealing and feasible in the near future.

7 Conclusions and future research

In the present work the development and the application of a novel virus-evolutionary genetic algorithm is tested for obtaining optimum tool paths for sculptured surface machining. The proposed algorithm handles quality objectives that reflect both surface accuracy and process productivity. Surface accuracy is studied via chord error and effective radius for inclined tools which enters the problem as a new criterion. Process productivity is represented via machining time. The algorithm is embedded to a cutting-edge CAD/CAM system and formulates a user-friendly, fully automated tool path planning infrastructure. The proposed methodology can facilitate CNC machine tool industry and CAM software developers by providing a straightforward methodology of customizing intelligent manufacturing software environments to create complex CNC machining part programs. Significant differences of the proposed approach against other strategies already proposed, lie on the optimization concept through non-Darwinian, non-conventional intelligence as well as on the implementation manner introducing technological criteria with parameter initialization from scratch, rather that process enhancement based on already selected parameters.

Looking further ahead, feed adaptation is to be incorporated to the existing algorithmic functions since local curvature analysis has already been conducted. This is to be achieved by considering sampling times for several NC control units found in industry. The consideration of physical properties and their successful relation to geometrical attributes is also one of the authors' major future perspectives and a full validation project will be established to test the proposed methodology against others employing several artificial intelligence modules.

The methodology proposed may be further enhanced to attain and share results under the ultimate goal of optimizing advanced manufacturing technology as a whole. The methodology may be controlled remotely via networks or team viewing environments to one or more computing systems. Moreover optimized CNC programs from intelligently formulated tool-paths can find their way towards cyber-physical manufacturing to exchange information and support highly flexible; individualized and resource-friendly production practices. Owing to its generic problem-solving nature; the presented algorithm may be adopted to similar manufacturing frameworks so as to automate and optimize processes such as STEP-NC prior to their employment; or even become a participant in “Internet of Things”; connected in a comprehensive network for supporting 4th generation’s intelligent technologies as well as “Industry 4.0”; where, the opportunity to further support production across the entire life-cycle introduces a new research field.

References

1. Sarkar S, Pratim PD (2015) Tolerance constraint CNC tool path modeling for discretely parameterized trimmed surfaces. Eng with Comput 31:763–773
2. Li DY, Peng YH, Yin ZW (2006) Interference detection for direct tool path generation from measured data points. Eng with Comput 22:25–31
3. Elkott DF, Veldhuis SC (2007) CAD-based sampling for CMM inspection of models with sculptured features. Eng with Comput 23:187–206
4. Chu CH, Wu PH, Lei, WT (2012) Tool path planning for 5-axis flank milling of ruled surfaces considering CNC linear interpolation. J Intell Manuf 23:471-480
5. Yang J, Altintas Y (2015) A generalized on-line estimation and control of five-axis contouring errors of CNC machine tools. Int J Mach Tools Manuf 88:9-23
6. Lin Z, Fu J, Shen H, Gan W (2014) An accurate surface error optimization for five-axis machining of freeform surfaces. Int J Adv Manuf Technol 71(5-8):1175-1185
7. Warkentin A, Ismail F, Bedi S (2000) Multi-point tool positioning strategy for 5-axis machining of sculptured surfaces. Comput Aided Geom Des 17(1):83–100
8. Tournier C, Duc E (2002) A surface based approach for constant scallop height tool path generation. Int J Adv Manuf Technol 19(5):318-324
9. Masood A, Siddiqui R, Pinto M, Rehman H, Khan MA (2015) Tool Path Generation, for Complex Surface Machining, Using Point Cloud Data. Proc CIRP 26:397 – 402
10. Kersting P, Zabel A (2009) Optimizing NC-tool paths for simultaneous five-axis milling based on multi-population multi-objective evolutionary algorithms. Adv En Soft 40(6):452–463
11. Ulker E, Turanalp ME, Halkaci HS (2009) An artificial immune system approach to CNC toolpath generation. J Intell Manuf 20:67-77
12. Zeroudi N, Fontaine M, Necib K (2012) Prediction of cutting forces in 3-axes milling of sculptured surfaces directly from CAM tool path. J Intell Manuf 23(5):1573–1587

13. Li L, Liu F, Chen B, Li BC (2015) Multi-objective optimization of cutting parameters in sculptured parts machining based on neural network. J Intell Manuf 26(5):891-898
14. Manav C, Bank HS, Lazoglu I (2013) Intelligent tool path selection via multi-criteria optimization in complex sculptured surface milling. J Intell Manuf 24:349–355
15. Fountas NA, Vaxevanidis NM, Stergiou CI, Benhadj-Djilali R (2014) Development of a software-automated intelligent sculptured surface machining optimization environment. Int J Adv Manuf Technol 75(5-8):909-931
16. Talon HJL, Garcia-Hernandez C, Berges-Muro L, Gella-Marin R (2014) Obtaining a spiral path for machining STL surfaces using non-deterministic techniques and spherical tool. Comput Aided Des 50: 41–50
17. Zain AM, Haron H, Sharif S (2010) Application of GA to optimize cutting conditions for minimizing surface roughness in end milling machining process. Exp Sys Appl 37: 4650–4659
18. Raja SB, Baskar N (2012) Application of particle swarm optimization technique for achieving desired milled surface roughness in minimum machining time. Exp Sys Appl 39(5): 5982–5989
19. Liu X-J, Yi H, Ni, Z-H (2013) Application of ant colony optimization algorithm in process planning optimization. J Intell Manuf 24(1): 1-13
20. Aydin I, Karakose M, Akin E (2011) A multi-objective artificial immune algorithm for parameter optimization in support vector machine. Appl Soft Comput 11(1): 120-129
21. Kim T (2015) Simulated annealing approach with an exhaustive greedy search for the optimal machining region decomposition in CNC roughing tool path generation. Int J Prec Eng Manuf 16(2): 399-440
22. Kubota N, Fukuda T, Shimojima K (1996) Virus-evolutionary algorithm for a self-organising manufacturing system. Comput Ind Eng 30(4):1015–1026
23. Chen X, Zhao J, Zhang W (2015) Influence of milling modes and tool postures on the milled surface for multi-axis finish ball-end milling. Int J Adv Manuf Technol 77(9-12):2035-2050
24. Redonnet JM, Djebali S, Segonds S, Senatore J, Rubio W (2013) Study of the effective cutter radius for end milling of free-form surfaces using a torus milling cutter. Comput Aided Des 45:951-962
25. Lavernhe S, Tournier C, Lartigue C (2008) Optimization of 5-axis high-speed machining using a surface based approach. Comput Aided Des 40(10-11):1015-1023
26. Malik S, Wadhwa S (2014) Preventing Premature Convergence in Genetic Algorithm Using DGCA and Elitist Technique. Int J Adv Res Comput Sci Soft Eng 4(6):410-418
27. Anderson N (1970) Evolutionary Significance of Virus Infection. Nature 227:1346-1347
28. Jian C, Li F (2013) An Improved Virus Evolutionary Genetic Algorithm for Workflow Mining. J Theor Appl Info Technol 47(1):406-411
29. Fountas NA, Vaxevanidis NM, Stergiou CI, Benhadj-Djilali R (2015) Intelligent CNC Tool Path Optimization for Sculptured Surface Machining Through a Virus-Evolutionary Genetic Algorithm. In: Davim JP (ed) Modern Manufacturing Engineering: Materials Forming, Machining and Tribology, Springer, Switzerland, pp 143-177
30. Fonseca CM, Correia MB (2005) Developing redundant binary representations for genetic search. Proc IEEE Congress on Evol Comput 2:372–379, Edinburgh, New York
31. Fountas NA, Kimpenis AA, Vaxevanidis NM, Davim JP (2012) Single and multi-objective optimization methodologies in CNC machining. In: Davim JP (ed) Statistical and computational methods in manufacturing, Springer, Heidelberg, pp 187-218
32. Aich U, Banerjee S (2014) A Simple Procedure for Searching Pareto Optimal Front in Machining Process: Electric Discharge Machining. Modell Simul Eng Article ID 594054, doi: 10.1155/2014/594054