

An integrated framework for optimizing sculptured surface CNC tool paths based on direct software object evaluation and viral intelligence

N. A. Fountas^{1,2}, R. Benhadj-Djilali², C. I. Stergiou³, N. M. Vaxevanidis^{1*}

¹Laboratory of Manufacturing Processes and Machine Tools (LMProMaT), Department of Mechanical Engineering Educators, School of Pedagogical and Technological Education (ASPETE), ASPETE Campus, N. Heraklion, GR 14121 Athens, Greece

²Faculty of Science, Engineering and Computing, Kingston University, Roehampton Vale Campus, Friars Avenue, Kingston upon Thames, London SW15 3DW, UK

³Department of Mechanical Engineering, Piraeus University of Applied Sciences, GR 122 44 Egaleo, Greece

* corresponding author; vaxev@aspete.gr

Abstract

Two critical objectives in sculptured surface tool path optimization are machining accuracy and process efficiency. The former objective is characterized as the combined effect of chord error and scallop height, known as surface machining error, whilst the latter may be reflected by the number of cutting tool locations constituting the tool path. These objectives are entirely depended on the values to be selected for computing tool paths under a given cutting strategy and preset tolerance. In order to determine optimal tool path parameters that will simultaneously satisfy the trade-off incurred between these objectives, a novel; generic and unbiased environment integrated with a virus-evolutionary heuristic for intelligent tool path optimization is presented. The proposed environment has been developed using the open architecture of a cutting edge CAM system whilst it deploys a set of interactive functions to straightforwardly assess criteria without the formal knowledge of any objective function; but directly from computer-aided manufacturing attributes; fully responsible to formulate efficient tool paths. A utility based on weighted summation for multi-objective optimization has been introduced to capture the direct output of globally optimized tool paths avoiding this way problem oversimplification and statistical errors that mathematical relations involve. Results have been rigorously validated both computationally and experimentally with the aid of a benchmark sculptured part that has been previously tested by several noticeable research contributions. Based on the quality of research outputs it is shown that the proposed framework for optimizing sculptured surface CNC tool paths may gain a prominent role for further extending the capabilities of current industrial strategies and be the flagship of allied; not-yet industrially interfaced approaches for deploying similar software integration tools to transfer significant results to production.

Keywords

Sculptured surface machining; tool path optimization; viral intelligence; intelligent algorithms

Introduction

As products need to be functional, aesthetic and more versatile, a tendency to employ modern methods in design and manufacturing has grown imposing high quality requirements in lower production times. CAD/CAM systems play key role to the support of such modern methods. New algorithms in design and manufacturing allow the development of free form features to enhance products from different industrial environments such as automotive; aerospace; electronics and mold/die. Selection for tool path parameters to machine products comprising sculptured surfaces in CAM systems is one of the most critical tasks. In general, latest proposals for addressing the sculptured surface machining optimization problem span a number of significant contributions that follow two main directions; tool positioning for sculptured surface CNC machining and intelligent optimization for either standard or newly developed tool paths. The major goal of tool positioning strategies is to generate feasible tool locations on a given part's surface considering important geometrical attributes such as principal curvature and cutting tool geometry. Redonnet et al. (2016) proposed a tool positioning strategy for 3-axis sculptured surface CNC machining with the use of a toroidal cutter. Their study takes advantage of "parallel-planes" milling and the effective radius of filleted-end mills to produce larger cut trajectories and simultaneously maintain scallop height at low levels. Gan et al. (2016) managed to answer to the research question concerning the optimal direction a cutter should rotate around to

achieve wider machining strip widths, by implementing the “mechanical equilibrium” theory when generating isoparametric tool paths. Rufeng et al. (2010) developed a module that integrates a commercially available CAM system to produce successive cutter contact points for sculptured surface tool paths when the cutter is to be controlled by two drive-curves. Their methodology first generates tool points for the first curve and according to the tolerance a second set of points is sequentially created with reference to the second curve to maximize machining strip width. Other older, yet well-established approaches dealt with the problem are Multi-point machining (Duvedi et al. 2014; Warkentin et al. 2000), Curvature matching (Lin et al. 2009), Arc-intersect method (Paul et al. 2005), rolling-ball approach (Gray et al. 2004), principal axis method (Rao et al. 1997), etc.

Tool path generation methods based on computational intelligence have also been proposed to address the sculptured surface machining problem. Li et al. (2015) implemented a back propagation neural network to optimize machining time, energy consumption as well as surface roughness by investigating spindle speed, feed rate, depth of cut and path spacing as process parameters for sculptured surface tool path generation. Their network is trained using known results obtained by design of experiments. Fountas et al. (2014) presented a software-based environment dedicated for optimizing sculptured surface tool paths for two-stages; roughing and finishing. Their approach optimizes rough machining tool paths based on the concept of attaining a uniformly distributed volume left to be finish-machined; whilst finish-machining is evaluated by iteratively assessing the surface deviation between the theoretical surface and the virtually machined one. The whole procedure is supported by a conventional genetic algorithm. Manav et al. (2013) proposed an intelligent approach of generating 3-axis optimal surface machining tool paths in terms of cutting force, cycle time and scallop height by adopting the weighted summation algorithm. Their approach is heavily based on a mathematical relationship representing the force model involved to the weighted criterion. Ulker et al. (2009) took advantage of “Clonal-G” artificial immune algorithm to find the most convenient step size in u and v parametric directions using known equations. Their approach refers to 3-axis surface machining with ball-end cutter whereas each algorithmic iteration results to a pair of curves intersected to give a surface point. This method is repeated until all necessary surface points are obtained to constitute the tool path. Krimpenis and Vosniakos (2009) proposed a novel strategy for sculptured surface rough machining optimization based on game theory. In their approach the goal is to optimize tool path parameters handled by two intelligent algorithms treated as players in a Stackelberg game. The goal is to optimize machining parameters for the benefit of both players and their environment (*Nash equilibrium*). Agrawal et al. (2006) proposed the implementation of a genetic algorithm capable of defining the optimal orientation of master cutter paths to produce an isoscallop surface topography and have machining time minimized as a side-effect. Their method is carried out in two steps. The first step involves the generation of the adjacent scallop curve from a known tool path trajectory (tool path interval computation) whereas the second step deals with the sequential determination of all cutter contact points under a preset tolerance constraint (forward step computation). Lazoglu et al. (2009) proposed a two-phase optimization approach for producing force-minimal sculptured surface tool paths. The first phase implements the “Minimum Spanning Tree-MST” algorithm to obtain optimal tool path points whereas the second-phase implements the “Traveling Salesman” algorithm for connecting them so that productivity is maximized. Castelino et al. (2003) presented an approach for properly ranking discrete tool path segments through the solution of a generalized “Traveling Salesman” problem to minimize air time during machining.

Several other intelligent systems have been employed to optimize general advanced machining tasks allied to the one discussed in this paper (Mellal and Williams 2016; Zainal et al. 2016; Yusup et al. 2014; Chu and Hsieh 2012; Zhang et al. 2006; Yildiz 2009a; 2009b; Yildiz 2012; Yildiz 2013a-c). For the essential case of sculptured surface tool path optimization, one may distinguish two research limitations, yet to be overcome. The first limitation refers to tool positioning strategies for sculptured surface tool paths, based on the fact that, independently dealing with tool contact points may not always guarantee tool path continuity and smoothness. In addition, the imminent abrupt tool path changes owing to large local curvature variations may deteriorate surface finish. To ensure tool path continuity as well as process efficiency, a systematic approach is needed to dynamically evaluate patterns of tool path points referring to the entire surface, rather than creating them one-by-one given a cut tolerance to build the tool path. The second limitation refers to the philosophy underpinning the employment of intelligent heuristics as optimization tools. The approach of deterministic expression for criteria as a function of tool path parameters through mathematical relations could turn prominent approaches to less efficient ones, despite their advanced attributes. That is; mathematical models playing the role of objective functions to intelligent algorithms may lead to partial problem solving owing to statistical errors, questioning thus optimality. By contrast, the direct link between an intelligent module and a CAM system’s interface (via interacting functions and a utility for measuring objective outputs), could represent the problem’s domain itself, resulting thus; to an unbiased; generic optimization framework for sculptured surface CNC machining tool paths. Besides CAM environment is in practice the most suitable resource for tool path generation whilst it provides the necessary properties for experimenting on different machining modeling scenarios and methods (Brecher and Lohse 2013; Zeroudi et al. 2012).

To avoid problem oversimplification and further push the envelope of unbiased tool path optimization to the required level of efficiency, generality and consistency, this paper aims to contribute to the field by presenting a novel

CAM-assisted optimization environment for sculptured surface tool paths; seamlessly supported by a new virus-evolutionary genetic algorithm to enable the simultaneous, software-automated assessment of heterogeneous tool path parameters. This allows for a rigorous process optimization by directly extracting objectives instead of repetitively evaluating objective functions that relate them to the parameters involved. To achieve seamless functionality, the environment handles CAM software automation routines developed for the dynamic retrieval of key components constituting the tool path (i.e. cutter location data). The proposed approach has been validated by conducting comparative tests concerning process simulations as well as actual 5-axis CNC machining, by examining a benchmark sculptured part for which results are available from several previous tool path generation/optimization approaches addressing the same problem.

Proposed dynamic framework for intelligent tool path optimization

The proposed optimization framework for generating intelligent tool paths is based on the concept of dynamically handling manufacturing software programming components and the development of interactive exploitation routines capable of linking tool-path parameters to the arguments of an intelligent system's algorithmic functions. Thereby, precise information about performance metrics may be obtained directly from the geometrical properties of a given sculptured surface, as opposed to deploying case-oriented mathematical solutions aiming at relating independent variables to discrete quality criteria.

As a first step, the proposed system initializes the environment by accepting values from the end-user, for a given set of tool path parameters being undoubtedly crucial in terms of their effect upon quality criteria such as the machining surface error as well as the number of cutter locations (Cai and Zhao 2014). The former criterion may reflect surface finish requirements as a combined effect of chordal deviation and scallop height, whilst the latter may represent productivity under the assumption that less cutter location points throughout the tool-path ought to minimize machining time (Kayal 2007; Ulker et al. 2009; Lin et al. 2014). A graphical illustration of the machining surface error as a combination of chordal deviation and scallop height is depicted in Fig. 1.

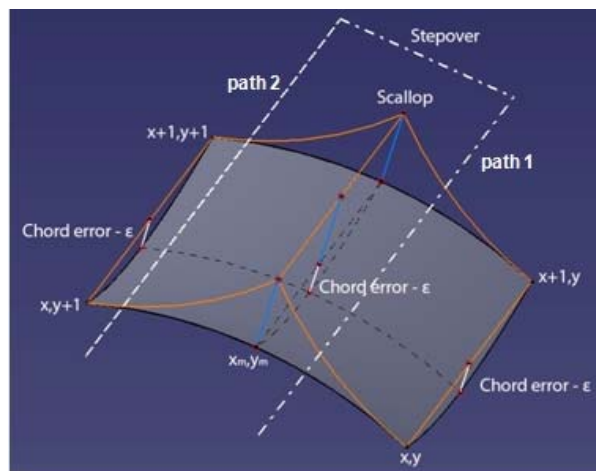


Fig. 1 Machining surface error as a combined effect of chordal deviation and scallop at an intermediate surface point

The parameters affecting the aforementioned criteria are the cutting tool geometry; the step-over; the maximum discretization step (distance between two consequent points on a part's surface; as it is interpolated by the tool during machining) as well as the two inclination angles; lead and tilt. Identification of surface tool path parameters is based upon their effect on surface tomography when it comes to material removal simulations (Tunk 2016; Tunk and Budak 2009). The next step is the automatic argument passing of inputs determined during the initialization stage, to the tool path's parameters for computing the aforementioned criteria. Note that determination of tool path parameters is conducted by considering an applicable parameter value range; as well as fundamental constrains (Fountas et al. 2014). Tool path evaluation with recommended values for process parameters against the criteria of machining error and cutter location number, is dynamically conducted utilizing the information contained in cutter location (CL) file; known also as "APT source" file. Once the tool path is automatically computed, a module dedicated for CL file parsing undertakes to distinguish key attributes for NC surface machining such as tool axis vector and tool tip coordinates. These properties facilitate geometrical analysis in CAD environment where cutter contact points are imported to the sculptured surface towards the formulation of surface normal vectors which are not available by the CL file (Tunk and Budak 2009). These instances are distinguished in the application's project tree as individual geometrical sets, named as "APT points" and "Normal Vectors". With reference to the topology of cutter contact points existing on a given sculptured surface, an interactive routine undertakes to sequentially compute local

curvatures according to the feed direction by employing vector algebra and retrieving dot products of normal vectors utilizing the angle between them. Thus, if two unit normals \vec{n}_1 and \vec{n}_2 separated by a 3D distance L_i on the sculptured surface; having also an angular difference θ , where $\theta = \arccos(\vec{n}_1 \circ \vec{n}_2)$ their local curvature at that local section may be computed by implementing Eq. 1 (Fisher 1989).

$$\rho_i = 2 \times \sin(\theta / 2) / L_i \quad (1)$$

3D distances L_i corresponded to the lengths of chords connecting the pairs of consecutive CC points by taking into account the machining axis system in Cartesian space, are computer in the same “For-Next” loop as the one implemented for local curvatures. Eq. 2 was applied to compute the sequential chords for pairs of cutter contact points; whereas Eq. 3 was used for the computation of corresponding chordal deviations. The routine developed for finding local curvatures as well as chordal deviations was verified by comparing its outputs to those obtained by manual processing via the “Surfacic curvature analysis” tool of Dassault Systemes CATIA® V5 R18 for the same sculptured surface and other experimental 3D free-form models.

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

$$C_e = \rho_i - \sqrt{\rho_i^2 - \left(\frac{L_i}{2}\right)^2} \quad (3)$$

where,

L_i : Chord length, or 3D point distance between two points in forward direction (mm)

x_i, y_i, z_i : Cutting tool point coordinates

C_e : Chord error (mm)

ρ_i : Local curvatures between normal vectors of points in forward direction (mm^{-1})

During the sequential evaluations executed by the intelligent part of the proposed environment scallop heights evolve for different kinds of cutting tool geometries; different step-over values as well as different inclination angles. As a result, scallop heights may not be accurately represented by retrieving automation objects for scallop height computation, available to CAM systems. A solution to the problem is given by considering not only the distance between the pairs of adjacent tool trajectories but also the effective cutting profiles for the three basic types of end mill geometries, flat-end; ball-end and filleted-end (Bedi et al. 1997; Vickers and Quan 1989). Cutting profiles during surface machining may be examined by considering the elliptical postures formed at the bottom of a cutting tool, if the latter is inclined and having its profile projected on a given sculptured surface. Major axes of projected ellipses do not seem to introduce important effect on the scallop magnitude given that they are oblique views of a cross-sectional plane of a half sphere, yet; minor axes greatly affect scallop magnitude since they depend on the tool’s inclined orientation. The fundamental attribute relating the scallop magnitude to tool orientation via the elliptical postures, is the effective radius R_{Eff} . At this point, several noticeable contributions such as the one of Duroobi et al. (2010) as well as Redonnet et al. (2013) concerning the investigation of effective radius variation during surface machining were taken into account to create expressions capable of extracting accurate data for scallop magnitudes during the sequential algorithmic evaluations from the intelligent algorithm embedded to the proposed optimization framework. Note that the requirement for gouge avoidance as well as cut tolerance is determined by the CAM environment in advance. Hence, only applicable tool paths are to be evaluated. Important studies estimating scallop magnitude with reference to the effective radius depending on the different cutting tool types may be found in Redonnet et al. (2013); Sarma (2000a); Sarma (2000b); as well as in Bedi et al. (1997). Eq. 4 was applied to compute R_{Eff} for flat-end; toroidal and ball-end mills. Note that for ball end-mills, R_{Eff} is not affected by tool inclination angles in 5-axis machining.

$$R_{eff} \begin{bmatrix} Flat - end \\ Toroidal \\ Ball - end \end{bmatrix} = \left\{ \begin{array}{l} \frac{R \times \cos^2 a_T}{\sin a_L \times (1 - \sin^2 a_T \times \sin^2 a_L)} \\ \frac{(R - r) \times \cos^2 a_T}{\sin a_L \times (1 - \sin^2 a_T \times \sin^2 a_L)} + r \\ R \end{array} \right\} \quad (4)$$

where:

R_{eff} : the effective cutting radius given the tool inclination;

R : the cutter's radius;

r : the cutter's corner radius for toroidal end-mills;

a_L : lead angle in degrees;

a_T : tilt angle in degrees.

An essential issue that is also taken into account by the proposed methodology is the false impression of perfectly sharp edges in the case of flat-end mills; mainly when it comes to the implementation of cutting inserts instead of solid carbide tools. To provide realistic outputs in the case of flat-end mill selection, the undercut magnitude designated by h_u parameter was also computed (Bedi et al. 1996).

Extraction of outputs necessary to assess generated tool paths, is the last step of the developed system. Tool path evaluation is performed by adopting the weighted summation scheme for the simultaneous multi-criteria optimization between surface machining error and cutter location points. The selection of this particular scheme was decided considering the practical need of a single optimal point against others existing on a Pareto front. Besides, the set of all potential solutions may be obtained by varying the weights for objectives involved given the end-user's preferences regarding the application (Manav et al. 2013). The aforementioned process developed for obtaining optimal tool paths for sculptured surfaces, is executed by the intelligent algorithm integrating the system under an iterative fashion; whilst driven by the goal of obtaining optimized values for independent parameters; cutting tool; step-over; maximum discretization step and lead-tilt angles, that will maintain both surface machining error and productivity under acceptable levels.

Viral intelligence

The intelligent module integrating the framework is an improved virus-evolutionary genetic algorithm which has been developed in Visual Basic® and is embedded to *Dassault Systèmes® CATIA® V5 R18* open API. Its main objective is the global optimization for tool paths by iteratively simulating the effect of each tool path parameter combination on the objectives of accuracy and efficiency (without the presence of any kind of mathematical relation among parameters and criteria, to play the role of objective function). Tool path parameters selected for optimization are responsible for defining different cutting tool orientations towards the whole sculptured surface during the algorithmic evaluations. In each iteration, a finite number of CC points existing on the surface, formulates a pattern of CC points (CC pattern) containing m and n rows of them, referring to X and Y directions respectively. Final topologies of CC points producing a $m \times n$ CC pattern are thus affected by the tool type; the magnitude of the two inclination angles as well as the two steps producing the surface machining error; discretization step and step-over. The last two aforementioned parameters (discretization step and step-over) are responsible for the maximum allowable chordal deviation and the maximum allowable scallop error respectively. Fig. 2 shows the effect on the number of CC points formulating a pattern on a sculptured surface when changing tool path parameters for the same cutting strategy. Fig. 2a shows a denser CC pattern referring to low values for tool path parameters (mainly for discretization step and step-over) whereas Fig. 2b shows a wider CC pattern referring to large values for parameters.

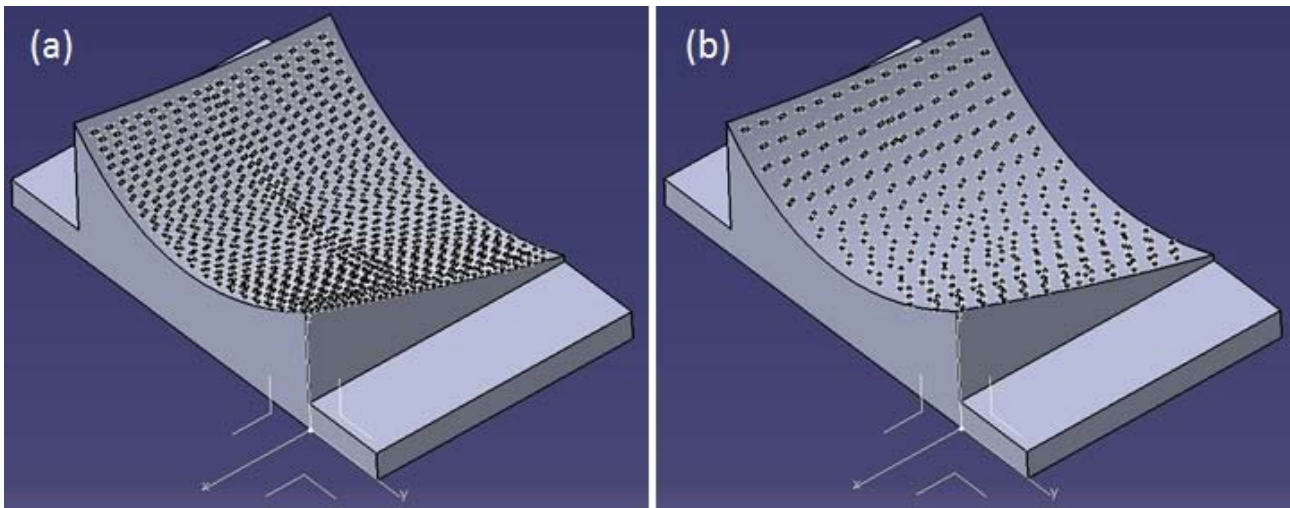


Fig. 2 CC patterns of two differently defined sculptured surface tool paths of the same cutting strategy

Therefore it is of great importance that, chromosomes in the proposed system should determine meaningful tool orientations (3D point coordinates) for real-time assessment within a preset limit range, whilst later genes should capture the need of identifying new values for tool path parameters, such that intelligently formulated CC point distributions towards X and Y directions are obtained to simultaneously minimize surface machining error and the number of CC points. The algorithm's general architecture adheres to the standard five-stage virus-evolutionary genetic algorithm cycle, pioneered by Kubota (1996) and modified prior to its employment for a typical algorithmic study, by Fountas et al. (2016). In order to clarify the contributing extend of its current status and further demonstrate the improved aspects of the existing work, only original attributes concerning the algorithm's infrastructure and novel functions are to be presented.

The virus-evolutionary genetic algorithm integrating the proposed intelligent tool path optimization framework implements two populations; the one of ordinary individuals and the one of viruses. The former population is conventionally used for the evolution process among candidate solutions, whereas the latter is used for recording patterns during the search so as to avoid the problem of premature convergence caused by the dominance of few elite candidate solutions. Virus individuals in the proposed algorithm ensure that effective patterns of candidate solutions won't be eliminated from the search space as the algorithm converges to the optimum result and diversity will be maintained. A mandatory scheme that contributes to the overall performance of the proposed algorithm is the artificial viral infection involving two operators; *reverse transcription* and *transduction*. That is; in every several generations, a few virus individuals have the chance to infect (reverse-transcribe) individuals' strings using their substrings whereas new viruses or modifications to the already existing ones, is performed via transduction. Since major scope is to preserve efficient solution patterns to avoid local trapping, transduction is employed to extract substrings from individuals and utilize these substrings as viral "chromosomes" to infect other individuals aiming at reinforcing their fitness. An extensive description about the role of virus infection and consecutively the two aforementioned viral operators is given in Kubota (1996).

Population of tool path "chromosomes"

Tool path "chromosomes" are represented as binary-encoded structures of the aforementioned tool path parameters. The number of bits determining the accuracy of represented tool path parameters sets also the accuracy for the search towards the optimum. To assess the effect of each of the randomly selected tool path parameters (individuals or simply "candidate" solutions) on the optimization criteria, the phenotype should be computed in order to dynamically achieve the argument passing from the virus-evolutionary GA to CAM environment and compute the resulting tool path. Thereby, the number of tool path "chromosomes" in a given population; the number of their bits as well as their locations in chromosome chains, are structured using series of arrays, as illustrated in the following equations.

$$C_{Pop}^{init} = \begin{bmatrix} [TlpPrm_{1,1}]_{N_b^1} & [TlpPrm_{1,2}]_{N_b^2} & [TlpPrm_{1,3}]_{N_b^3} & [TlpPrm_{1,4}]_{N_b^4} & [TlpPrm_{1,5}]_{N_b^5} \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ [TlpPrm_{n,1}]_{N_b^1} & [TlpPrm_{n,2}]_{N_b^2} & [TlpPrm_{n,3}]_{N_b^3} & [TlpPrm_{n,4}]_{N_b^4} & [TlpPrm_{n,5}]_{N_b^5} \end{bmatrix} \quad (5)$$

$$N_b^{TlpPrm_{i,j}} = \begin{bmatrix} [N_b^1]_{1,j} & [N_b^2]_{1,j} & [N_b^3]_{1,j} & [N_b^4]_{1,j} & [N_b^5]_{1,j} \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ [N_b^1]_{5,j} & [N_b^2]_{5,j} & [N_b^3]_{5,j} & [N_b^4]_{5,j} & [N_b^5]_{5,j} \end{bmatrix} \quad (6)$$

$$C_{Pop}^{init} = \begin{bmatrix} [Lgth_i] & u_i & l_i \\ \dots & \dots & \dots \\ [Lgth_n] & u_i & l_i \end{bmatrix} \quad (7)$$

C_{Pop}^{init} is the initial population of tool path “chromosomes”; $TlpPrm$ a tool path’s parameter; $N_b^{TlpPrm_{i,j}}$, the number of bits required for the accuracy of each tool path parameter ($TlpPrm$) for the i_{th} tool path “chromosome” of the j_{th} population and $Lgth_i$ is the i_{th} tool path’s chromosome length. CAM software simulations for tool path “chromosomes” are conducted after converting all parameters involved to the optimization problem from binary to real-encoded values. Hence, considering a given tool path parameter $TlpPrm_i$; its corresponding domain $D_{TlpPrm_i} = [u_i, l_i]$ and the i_{th} tool path’s chromosome length $Lgth_i$, the following expression is used for the conversion of binary encoding values to real-encoded ones.

$$TlpPrm_i = l_i + fnc(BinStr) \times \frac{u_i - l_i}{2^{Lgth_i} - 1} \quad (8)$$

Where, u_i , l_i are the parameter’s upper limit and lower limits; whereas $fnc(BinStr)$ is a function developed to return the decimal values for binary-encoded schemes depending on the accuracy requirements.

Weighted objective evaluation utility

Both aforementioned optimization criteria are expressed as a common weighted utility to simplify painstaking computations and to arrive at a single-point output that will facilitate practitioners against the concept of selecting a solution from many, existing to a Pareto front (Manav et al. 2013). Besides, all Pareto optimal solutions may still be reached by the same scheme, if the weights associated to each of the objectives are sequentially changed to modify the relative importance to the overall optimization output as it has already been mentioned. Prior to the employment of the weighted objective utility, proper normalization using the minimum and maximum values of the single criteria of machining error and CC point number, has been conducted to remove the bias owing to the different unit magnitudes (Chrysolouris and Subramaniam 2001). The weighted objective utility with respect to the single criteria and their weights is depicted in Eq. 9.

$$ObjFcn_{ws} = w_1 \times Crt_1 + \dots + w_n \times Crt_n \quad (9)$$

with, $w_i (i = 1, 2, \dots, n)$ and $\sum_{i=1}^n w_i = 1, 0 \leq w_i \leq 1$. Crt_i would correspond to the i_{th} single criterion for optimization whilst w_i would correspond to its related weight of importance.

For the criterion of surface machining error, a novel function has been developed to dynamically interact with CAM software and retrieve the randomly selected tool from the current tool path “chromosome” being under evaluation so as to identify its geometrical configuration and apply the proper mathematical expression to compute R_{eff} magnitude. The requirement of minimizing cutter contact points has been facilitated by the automatic retrieval of “APT points” geometrical set attached to the machining document’s project tree. The weighted objective values are then ranked in ascending order given their magnitudes. Since tool orientation in the case of sculptured surface tool path generation is quite sensitive even to small differences among objective values, an exponential expression was developed to make such magnitudes prominent enough to be able to distinguish them. Note that, when the difference between objective values is not that obvious, the algorithm may select parents randomly rather than heuristically. The expression is given in Eq. 10.

$$Fitted_i = FitSum \times \left[\exp(-ObjVal_i^2) \right] \quad (10)$$

In Eq.10 $FitSum$ represents the summation of ranked objective values of individuals and $ObjVal_i$ the objective value attained by i_{th} individual in a population. Fitted individuals are selected with a given probability to formulate a mating pool. Thereby single-point crossover and single-point mutation are applied to produce a new generation of candidate solutions for assessment.

Viral infection

It has been mentioned that the role of virus individuals is to preserve effective schemata from candidate solutions, since local information may be very useful for solving such complex optimization problems as sculptured surface machining. As a result, the process of operating with viruses as effective patterns to offer information for producing better offspring during the search, is finally what truly gives the system’s its merits. In general, the key procedure of viral infection overcomes the drawback of crossover and mutation operators referring to their inability of directly handling schemata. Note that an increase of a schema equals to an increase of local information in a given population. Unfortunately increase in schemata occurs not only to efficient but also to inefficient ones owing to proportional selection. On the other hand virus infection operation directly handles chromosome schemata since it creates substrings of candidate solutions as viruses. The combination of hosts and viruses prompts the algorithm to generate new candidates as a horizontal propagation which is beneficial for local information handling. In addition, viral individuals may vertically transmit their “genetic material” to the strings of hosts, thus; maintaining their local information from generation to generation as vertical inheritance. As a result global search is facilitated with the aid of local data as well.

Nevertheless, since no direct mathematical expression exists to relate independent tool path parameters to the criteria presented above and serve as the objective function, no guarantee that offspring will occur better than parents is given. Hence, good candidate solutions may be lost when crossover and mutation operators are applied. Note that mathematical relations used as objective functions, often “highlight” the proper search direction to facilitate several heuristics towards their way for finding optimum results. In the problem tackled here, neither a “known” mathematical model exists to serve as the objective function; nor continuity of the search domain is ensured; only real values directly obtained and appeared to a simple weighted-objective utility are retrieved for computing the outputs. To solve this issue, elitism was applied to copy small proportions of fittest solutions into next generations. As a consequence viruses to be created by the process of transduction will have higher probability to preserve outstanding seed to favor the infection of other individuals towards the search.

Initial virus individuals are created after evaluating the fitness of all individuals existing to the main population, as a fraction of the main population’s magnitude. Unlike the random selection of individuals performed in conventional virus-evolutionary genetic algorithm presented so far for producing the virus population, the one in this paper performs both targeted as well as random selection. The former selection is decided upon the elite of few outstanding individuals whereas the latter is normally decided upon the rest of the individuals through a probability, so as to sustain an unbiased selection procedure. The algorithm proposed initializes a series of functions to deliver the necessary outputs. These functions include also some components that resemble the original ones presented by Kubota (1996) as well as new ones to further extend the algorithm’s potentials.

The initial population of tool path “chromosomes”; C_{Pop}^{init} is randomly generated and then transduction operation is applied to both fitted and randomly selected individuals to create the population of viruses V_{Pop}^{init} . A virus created

by transducing from the i_{th} chromosome of the j_{th} population is denoted as Vrs_{ij} . Substrings being cut represent viruses' chromosomes whose length is denoted as $VrsLgth_i$. Locus $i=1$ is the starting point from which $VrsLgth_i$ length will be determined and Locus i_{max} is the ending point. These two loci are randomly determined and are constrained to the original host's chromosome length $Lgth_i$. The chromosome length ($Lgth_i$) of individuals of the main population is fixed; whereas the length of virus individuals ($VrsLgth_i$) extends as the evolution process continues. Viruses attack to infect host individuals, using *reverse transcription* operator for overwriting a string of a virus Vrs_{ij} to a portion of an individual's Idv_j string. The indices of both Vrs_{ij} and Idv_j are declared in advance so as to sequentially perform the replacement of selected binary digits according to the determined references.

Assessment of virus individuals is performed using their fitness scores denoted as $FitVrs_{i,j}$ reflecting their infection capability. This fitness is computed after the successful infection of Idv_j by Vrs_{ij} as Eq. 11 indicates:

$$FitVrs_{i,j} = FitInfIdv_j - FitIdv_j \quad (11)$$

The value obtained by Eq. 11 is the difference between the two fitness values of individual Idv_j before and after its infection by Vrs_{ij} . Given that Vrs_{ij} might infect more than a single individual (let S be the set of infected individuals) then $FitVrs_{i,j}$ reveals the improvement of fitness values of all infected individuals and is as Eq. 12 determines:

$$FitVrs_i = \sum_{j \in S} FitVrs_{i,j} \quad (12)$$

Efficiency in terms of a virus Vrs_{ij} infectivity is dictated through the infectivity force $InfForce_{i,g}$ where i the index of the virus Vrs_{ij} and g its current generation. This parameter is depended from the result of $FitVrs_i$ and is compared to the one attained by the virus Vrs_{ij} in a previous generation. If its value is lower than the one attained in the previous generation ($InfForce_{i,g-1}$) then new *transduction* operation is applied to produce a new virus. If the value of $InfForce_{i,g}$ is greater than $InfForce_{i,g-1}$ then the virus Vrs_{ij} increases its robust scheme by transducing a partially new substring from the infected individual. The mechanisms of *transduction* and *reverse transcription* are depicted in Fig. 3. Fig. 3a shows the generation of a virus individual by employing transduction; Fig. 3b depicts the virus infection process via transduction and Fig. 3c depicts the transduction for a partial individual string segment should the value of $InfForce_{i,g}$ is greater than $InfForce_{i,g-1}$ for a Vrs_{ij} .

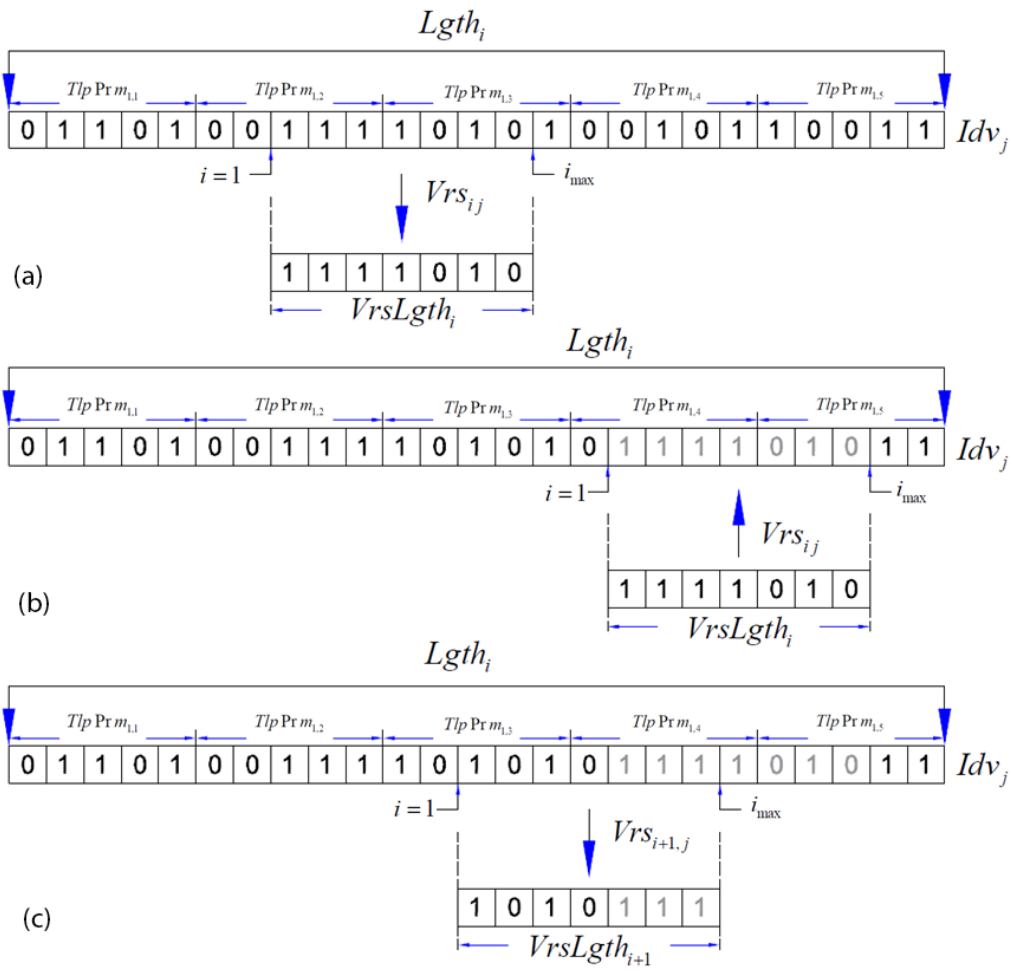


Fig. 3 Intelligent operations of virus-evolutionary genetic algorithm: (a) transduction; (b) reverse transcription; (c) partial transduction

In conventional virus-evolutionary genetic algorithm the infection rate of viruses is dictated via their fitness score $FitVrs_i$ and the infection rate of viruses; $InfRateVrs_{i,j}$, $0 \leq InfRateVrs_{i,j} \leq 1$ for self-adaptively adjusting the search between exploitation and exploration. In the proposed algorithm $InfRateVrs_{i,j}$ can be manually restricted to new intervals should the algorithm fail to increase effective schemata for candidate solutions. Hence, infection rate may take a minimum value so that $0 \leq MinInfRateVrs_{i,j} \leq 0.5$ and a maximum value so that $MaxInfRateVrs_{i,j}$, $0.5 \leq MaxInfRateVrs_{i,j} \leq 1$ in the case of handling a problem that is mainly of global search and local search respectively. In other words self-adaptation in terms of exploitation and exploration is controlled according to the engineering problem's nature using discrete operational intervals. Fig. 4 gives the workflow of the proposed virus-evolutionary genetic algorithm integrating the environment.

The entire tool path optimization framework has been developed in *Visual Basic*® and is hosted as an interactive module to *Dassault Systèmes*® *CATIA*® V5 R18 open API. The usage of the intelligent part of the application as a stand-alone module is also possible whereas the whole environment can be attached to other CAM systems provided that their software development infrastructures is supported by *Visual Basic*® for Applications (VBA) to ensure compatibility. Fig. 5 illustrates the developed intelligent tool path optimization infrastructure. At the early stage of tool path generation the user is prompted to determine an applicable range (minimum and maximum limits for tool path parameters) for the formulation of the initial tool path chromosome and design space. For each of the tool path parameters involved the number of binary digits to represent accuracy and facilitate computations for phenotype values is also specified in that stage. Determination for tool path parameters is performed by taking into account the available resources and technological constraints such as cutting tool types, machine tool limits in tool inclination, constraints imposed owing to part's geometry, etc. The rest of the procedure targeting to optimal values for tool path parameters automatically evolves until stopping criteria are met.

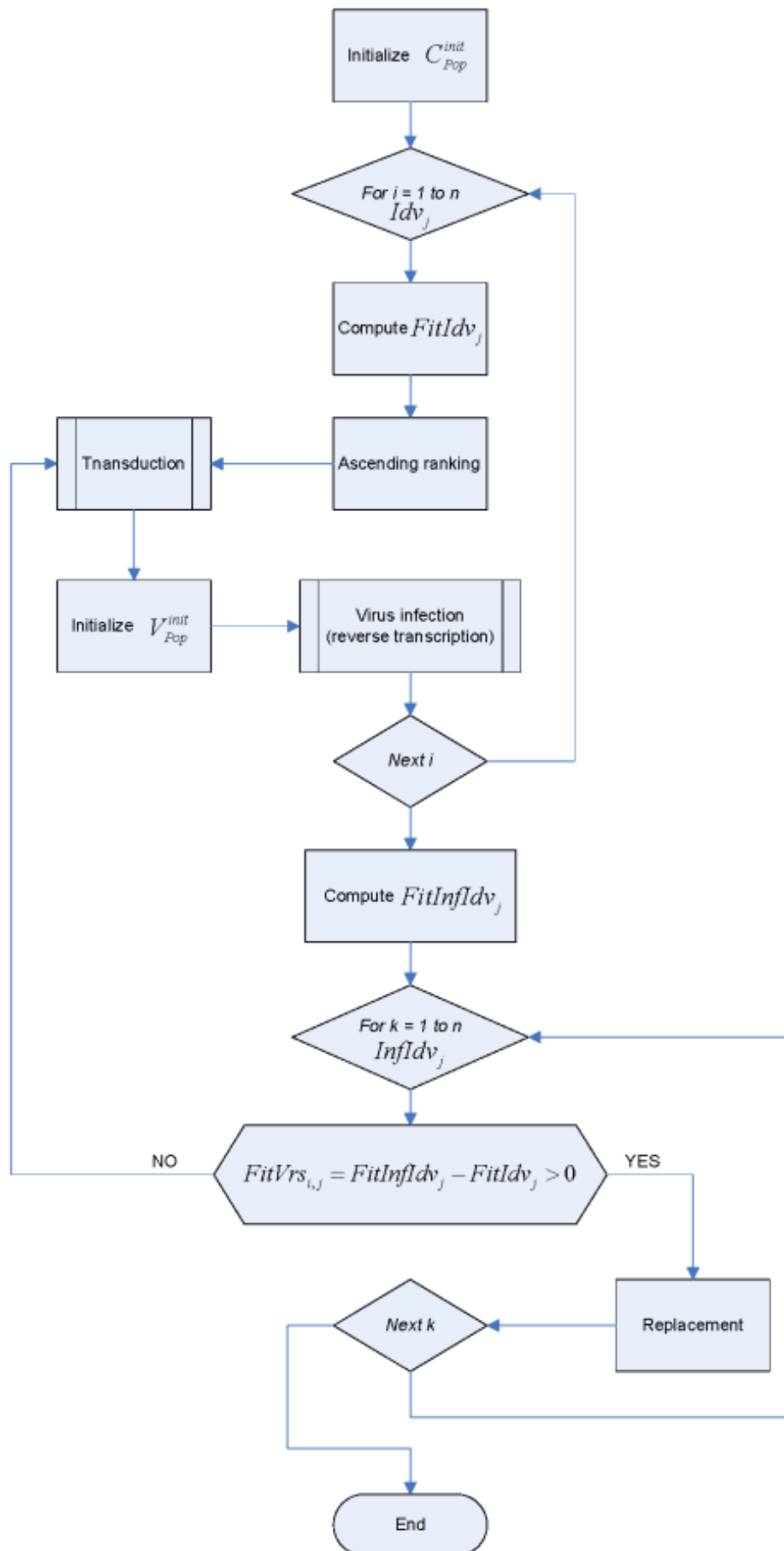


Fig. 4 Workflow of the virus-evolutionary genetic algorithm

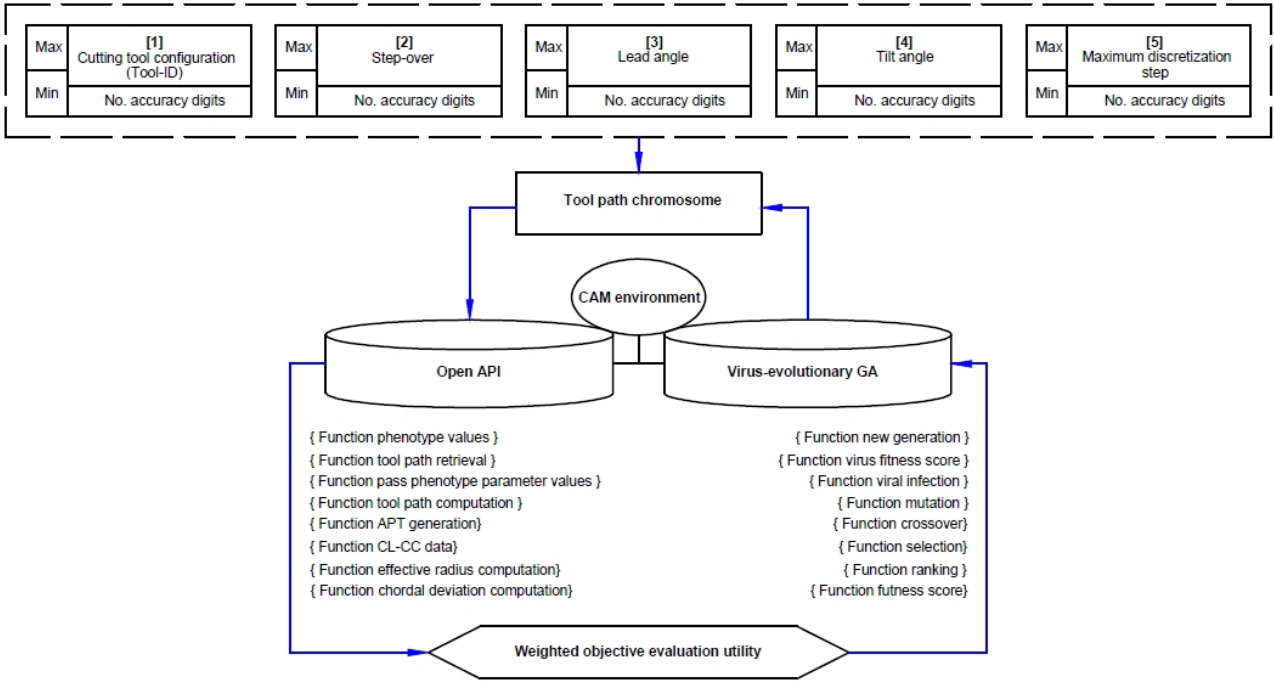


Fig. 5 Intelligent sculptured surface CNC tool path optimization environment

Experimental study for optimal algorithmic performance

For the proposed algorithm to reach its full potentials, a study aiming at finding best settings for viral parameters was conducted using a highly multimodal benchmark equation as objective function. The results obtained were compared to those found in Kubota (1996) for the same function. The function is given in Eq. 13.

$$f(x, y) = x^2 + y^2 - 0.3 \cos(3\pi x) - 0.4 \cos(4\pi y) + 0.7 \quad (13)$$

where $-1.0 \leq x, y \leq 1.0$, $x = y = 0, f(x, y) = 0$. In the experiment, the effect of variation of a viral parameter on the objective has been investigated at a time, whilst; the rest of the parameters were kept constant. The parameters investigated were virus population V_{Pop} , the length of substring for virus individuals $VrsLgth_i$, the infectivity force $InfForce_{i,g}$ and the infection rate $InfRateVrs_{i,j}$ having four levels each. During this experimentation the proposed algorithm run an evaluation process of 20 candidate solutions (population size) and a maximum of 50 generations. The experimental design is given in Table 1 and results of the experiment are illustrated in Fig. 6.

Table 1 Design table for factorial experiment addressing viral parameters for best settings

Viral parameters	Experimental levels			
	1	2	3	4
V_{Pop}	2	5	7	10
$VrsLgth_i$	10	20	30	40
$InfForce_{i,g}$	0.001	0.01	0.1	1.0
$InfRateVrs_{i,j}$	1 - (10%)	4 - (40%)	7 - (70%)	10 - (100%)

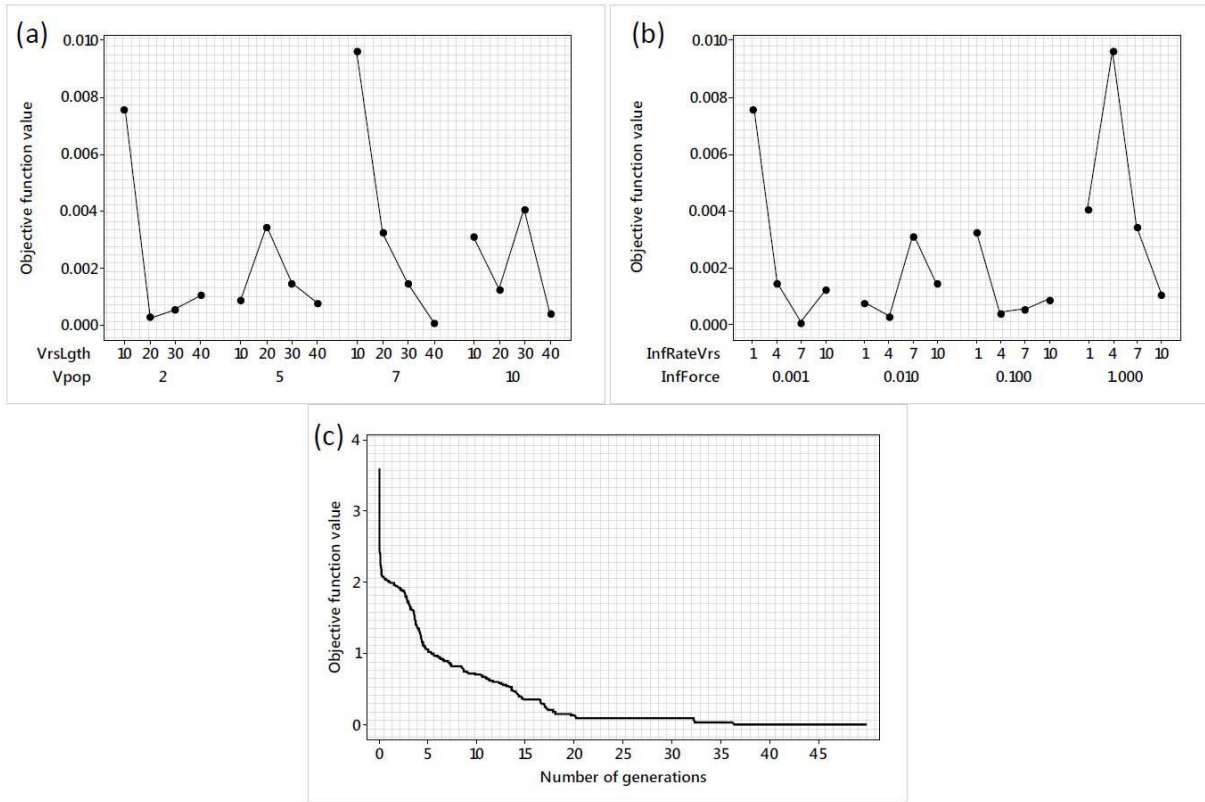


Fig. 6 Results of viral parameter settings experiment: (a) effect of virus population according the levels of substring length; (b) effect of infectivity force according the levels of infection rate; (c) Average fitness values according to the number of generations

The best result obtained gave $f(x, y) = 5.55E-17$ for $x = 0, y = 0$. These results outperform the original ones reported in Kubota (1996) attaining $8.12E-5$ for the same population size with conventional virus evolutionary genetic algorithm and $1.00E-4$ for the same population size using a steady-state genetic algorithm (SSGA). By considering the experimental results, for a main population size equal to 20, 7 viruses $Vrs_{i,j}$ deliver best fitness score, under a variable substring length $VrsLgth_i$ no more than 40 digits (out of 46, which is the length $Lgth_i$ of tool path chromosomes in the main population). For these settings, infectivity force $InfForce_{i,g}$ is advantageous when set using its lowest value (inheritance preservation) whilst infection rate $InfRateVrs_{i,j}$ ought to be set equal to 70% (good local search).

Experimental validation

To validate the proposed sculptured surface CNC tool path optimization environment a benchmark second-order parametric open form surface investigated by several researchers, (Gan et al. 2016; Rufeng et al. 2010; Warkentin et al. 2000; Rao et al. 1997) was selected as a reference part. Its free-form surface is defined as follows:

$$S(u, v) = \begin{bmatrix} -94.4 + 88.9v + 5.6v^2 \\ -131.3u + 28.1u^2 \\ a_1 + a_2 \end{bmatrix} \quad (14)$$

with $a_1 = 5.9(u^2v^2 + u^2v) - 3.9v^2u + 76.2u^2$; $a_2 = 6.7v^2 - 27.3uv - 50.8u + 25v + 12.1$.

Multi-axis, zig-zag style swept-cut tool path was applied to the part's surface. The pattern formulated by applying this surface tool path, is a typical offset of the surface in the form of tool trajectories. For this tool path cut tolerance was specified equal to 0.01 mm as it is indicated to previous research handled the same problem. Feed direction was towards the part's minimum principal direction dictated along X-axis direction. The proposed intelligent environment was applied to the aforementioned surface using best viral parameter settings. The proposed methodology was employed using a tool database with nine tools for the intelligent algorithm to select from; (three sets of flat-end,

ball-end and toroidal tools; having 8mm, 12mm and 16mm each). In order to determine a meaningful search domain for optimizing surface tool path parameters, machining modeling was restricted to produce gouge-free tool paths. The ranges for the rest of tool path parameters were determined as follows: Step-over [20%-45%] as a percentage of the selected cutting tool diameter, lead angle [0deg-5deg], tilt angle [0deg-2deg], and maximum discretization step [0.005-0.01].

Algorithmic validation: Comparison using computational results

The results obtained using the proposed virus-evolutionary genetic algorithm compared to those obtained by operating the same algorithm without the presence of viral operators. Both algorithms run with a population size of individuals equal to 20 for 50 generations. Fig. 7 shows the convergence of both algorithms tested. It is indicated that virus-evolutionary genetic algorithm starts from lower objective function value whereas its convergence trend dictates faster convergence to final point which was equal to 0.359608 against the value of 0.361086 suggested as the optimum by the conventional genetic algorithm. Tool path parameters corresponding to the above objective function values were transferred to *Dassault Systèmes® CATIA® V5 R18* advanced surface machining workbench in order to conduct comparisons in terms of algorithmic validation. Table 2 shows the tool path parameters attained by the algorithms.

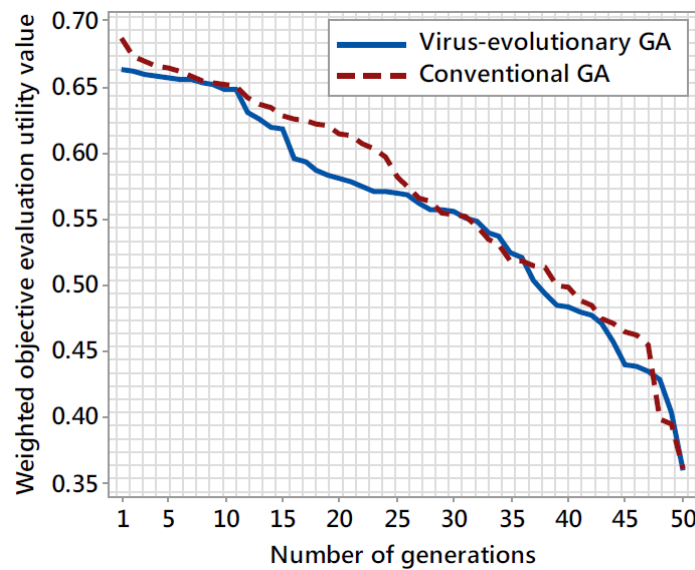


Fig. 7 Convergence trends for virus-evolutionary genetic algorithm and conventional genetic algorithm

Table 2 Optimal tool path parameters suggested by genetic algorithm and virus-evolutionary genetic algorithm

Intelligent heuristic	Best parameters attained				
	Tool-ID	Step-over (% ϕ & mm)	Lead angle (deg)	Tilt angle (deg)	Max. discretization step ($\times 10^{-3}$ mm)
Genetic algorithm	6 ($\phi 16Rc3$)	30.791 % 4.927 mm	3.379	0.040	8.451
Virus-evolutionary genetic algorithm	6 ($\phi 16Rc3$)	41.729 % 6.677 mm	2.957	0.027	6.338

The tool path recommended by the conventional genetic algorithm consisted of 337 cutter contact points whereas the corresponding mean chord error towards feed direction (X-axis) was equal to 0.0043 mm. The tool path recommended by the virus-evolutionary genetic algorithm integrating the proposed optimization framework, consisted of 289 cutter contact points whereas the corresponding mean chord error towards feed direction was equal to 0.0036 mm. With reference to the weighted objective evaluation utility scores (0.359608 for the virus-evolutionary genetic algorithm and 0.361086 for the conventional genetic algorithm) and the results concerning mean chord errors and cutter contact points, a false impression might be given speculating that, marginal differences between parameter values suggest insignificant differences in either simulated or physical outputs as well. To substantiate the opposite assumption, simulated surface topographies and machining times derived by the CNC unit were investigated. Simulated outputs were examined by implementing the point cloud method for analyzing 3D tolerance between

machined and desired surfaces. 3D tolerance analysis using digitized point cloud data was conducted in *Geomagic Qualify® 2013* 3D metrology and automation environment. “Best-fit alignment” utility was employed to automatically combine instances so as to realize optimum alignment. Results obtained indicated that both tool paths are within the preset tolerance range $[-0.01/+0.01]$, yet; the tool path optimized by the proposed methodology nearly reaches the theoretical surface to a larger extent than the one recommended by the conventional algorithm. 3D tolerance analysis involved also the examination of minimum/maximum point surface deviation as well as minimum/maximum average 3D deviation between machined and ideal sculptured surfaces. The values for these metrics that the conventional GA produced were Max $[+/-]$ 3D deviation equal to $[+0.0103 \text{ mm}/-0.0103 \text{ mm}]$ and Max $[+/-]$ average 3D deviation equal to $[+0.0041 \text{ mm}/-0.0050 \text{ mm}]$. For the same metrics, the proposed virus-evolutionary GA produced a 3D output model with Max $[+/-]$ 3D deviation equal to $[+0.0103 \text{ mm}/-0.0103 \text{ mm}]$ and Max $[+/-]$ average 3D deviation equal to $[+0.0046 \text{ mm}/-0.0052 \text{ mm}]$. By taking into account the absolute 3D deviation difference between the average values of both algorithms it is indicated that for the conventional GA is 9×10^{-4} and the virus-evolutionary GA is 6×10^{-4} . Despite this negligible result still the latter algorithm outperforms the conventional one in terms of simulation magnitudes. By simulating a cutting feed equal to 1000 mm/min , tool path times were approximately 3.00 min and 2.00 min for the conventional algorithm’s recommended tool path and the one by the virus-evolutionary genetic algorithm respectively. Fig. 8a illustrates the simulated results for surface machining error analysis and distribution of CC points, corresponding to the optimized tool path of the conventional genetic algorithm tested. Fig. 8b shows the analogous outputs for the virus-evolutionary genetic algorithm. It can be concluded that the final output deemed as the optimal by the virus-evolutionary, yields a profound improvement compared to the non-dominated solution the conventional algorithm achieved for tool path parameters. According to the solution recommended by the proposed algorithm, the toroidal cutter achieved a wider machining strip owing to the larger step-over distance. This step-over distance still satisfies the requirement of scallop elimination since the machining strip is produced under the flatted region for this toroidal cutter. The lower lead angle the proposed algorithm recommended allows for larger effective radius than the one of conventional algorithm’s solution. The same also goes for the results of tilt angle values the two algorithms proposed. The result of $6.338 (\times 10^{-3} \text{ mm})$ against the value of $8.451 (\times 10^{-3} \text{ mm})$ produces less pronounced “faceted” result owing to the interpolation, yet it could be responsible for increasing the number of CC points. However the larger step-over the proposed algorithm recommends against the one of conventional algorithm gives the merit to tool path efficiency by lowering the number of CC points. Therefore, by taking into account the final outputs in terms of surface finish as well as machining time, it comes as a conclusion that the improvements warranted by the proposed approach are significant and with practical merits.

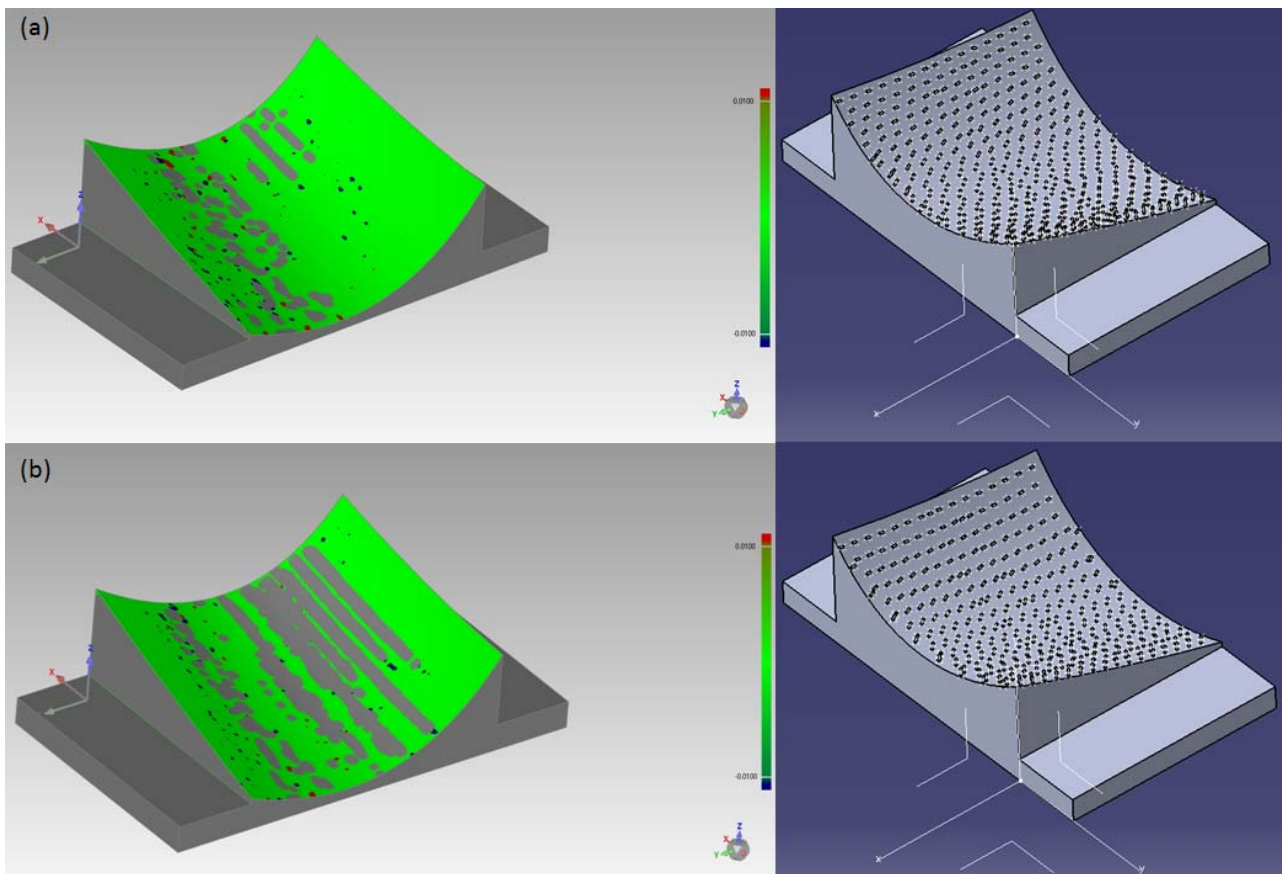


Fig. 8 Simulated results by implementing optimal parameters: (a) genetic algorithm; (b) virus-evolutionary genetic algorithm

Manufacturing validation: Comparison based on existing tool path optimization methodologies

A 5-axis dry machining test based on the algorithm's recommended tool path parameters was conducted on a Fooke® Endura 9 series spindle head-tilted (A and C axes), CNC machining center equipped with a Siemens Sinumeric® 840D CNC control. The stock material prepared for clamping was a 94.5 x 105 x 50; Al 7050 T-7451 aluminum alloy. The toroidal cutter proposed was an Osborn® supermill standard solid carbide $\varnothing 16$ mm, $R_c 3$ mm fixed to an HSK 63A tool holder. Spindle speed was set to 9950 rev/min and feed rate was set to 3000 mm/min. The total machining time was 1 min, 24 sec. Figs. 9a, 9b, 9c and 9d illustrate the part setup, the machine tool tilted spindle, the finishing operation and the resulting part respectively.

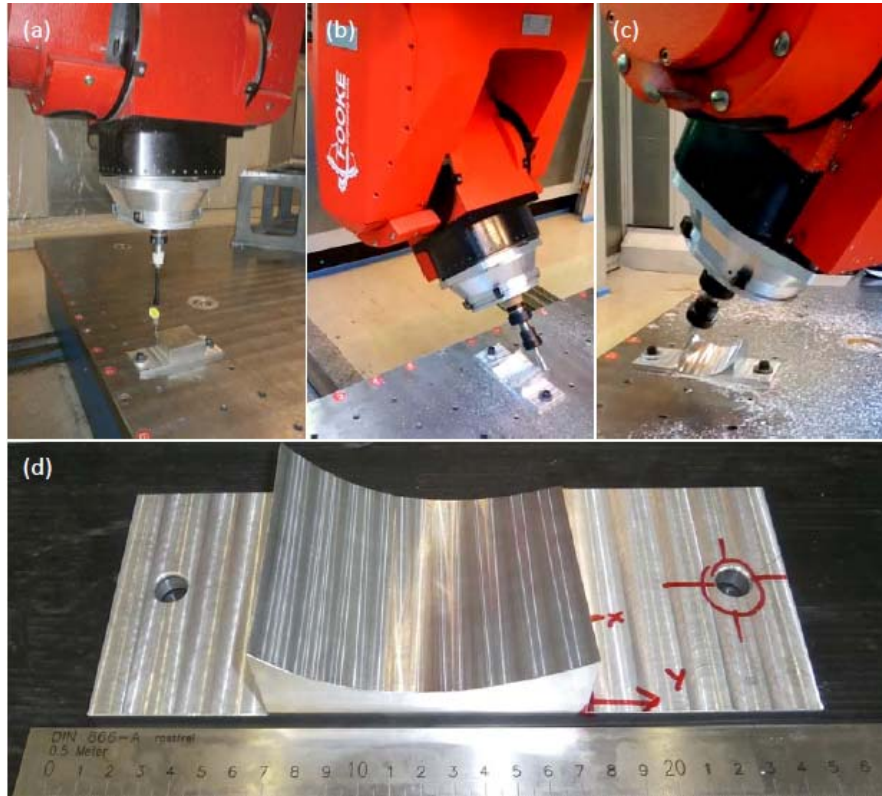


Fig. 9 Actual 5-axis dry machining of the part used for validating the proposed tool path optimization methodology: (a) part setup; (b) start of finish machining operation; (c) end of finish machining operation; (d) machined part

Final part was inspected on a DEA® gantry line CMM using a Renishaw® PH10M TP2 probe ($\varnothing = 2$ mm) with 10 mm stylus extension. Surface deviation was examined along the normal vectors between the machined and the designed sculptured surface in four selected profile sections of OYZ plane, at $X=-5$ mm; $X=-30$ mm; $X=-60$ mm and $X=-90$ mm as it was suggested for testing *principal axis* tool path planning (Rao et al. 1997); *multi-point* machining (Warkentin et al. 2000) as well as *dual-drive curve* tool path planning method (Rufeng et al. 2010) using the same part. The measuring step length of 0.5 mm towards Y direction was applied to measure all aforementioned profiles. The CMM inspection program was generated in LK-DMIS 5.22®/LK CAMIO® environment (Fig. 10). Fig. 11 shows the surface deviation trends according the inspection conducted to the aforementioned profile sections.

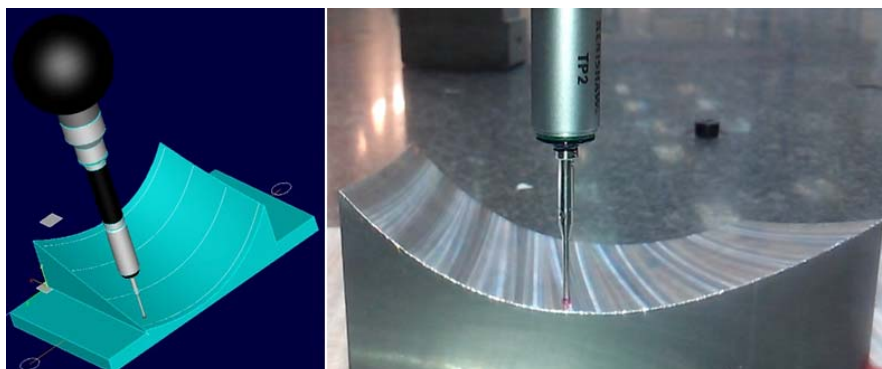


Fig. 10 Virtual and actual CMM measurements conducted on benchmark sculptured part

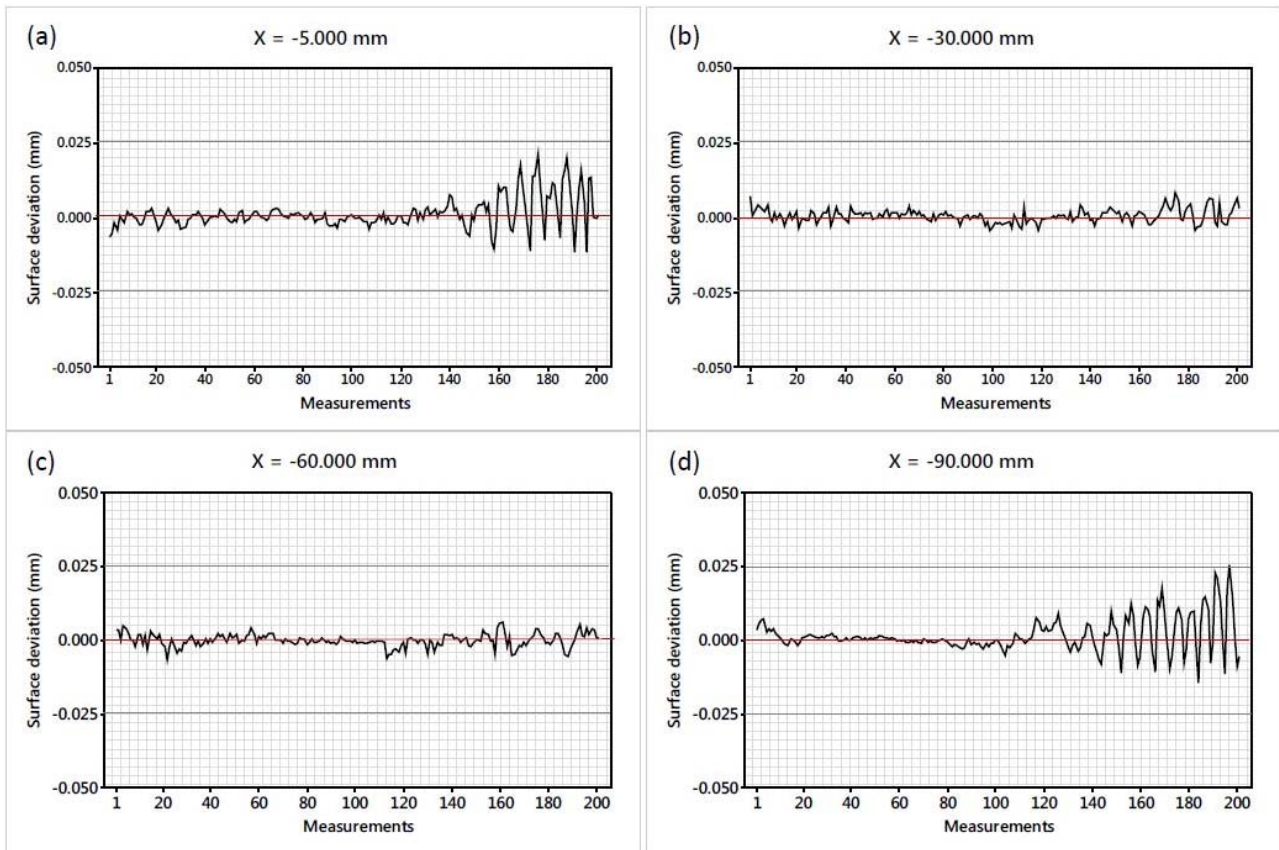
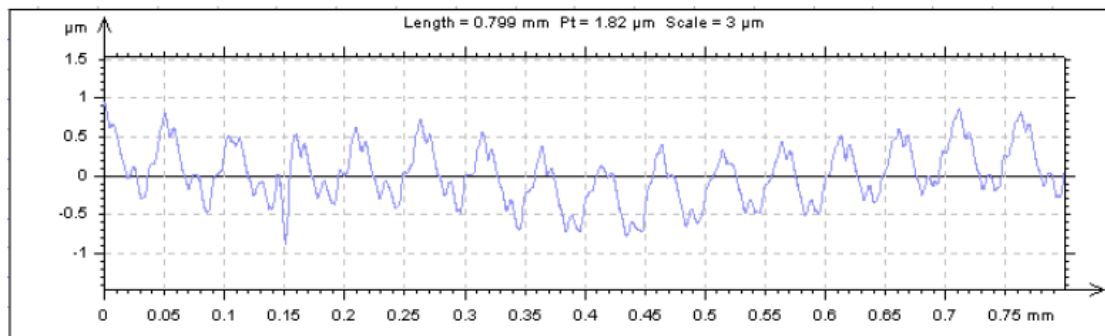


Fig. 11 Surface deviation measurements for selected profile sections of the benchmark part: (a) X=-5mm, (b) X=-30mm, (c) X=-60mm, (d) X=-90mm

By examining Fig. 11 it can be observed that surface deviation falls within the range of $[-0.01/+0.01]$ referring to all selected profile sections. Note that profile sections selected at X=-5mm and X=-90mm exhibit a large form error owing to tool deflections, CMM measurements, work holding, etc. By neglecting inconsistencies owing to form error, surface deviation still satisfies the cut tolerance. Besides these profiles are close to the fringes of the cutting tool's approach and departure where larger; yet smooth scallops exist. Largest scallops distribute between the 140th and 200th measurement for all selected zones, yet with a more profound result for X=-5mm and X=-90mm. The junctures between adjacent passes as well as connected postures between contact points were found to be continuous; smooth and free of sharp scallop edges. Note that roughness average was far below 1 μm . An illustrative roughness measurement diagram accompanied with its corresponding parameters using *Taylor-Hobson® Surtronic 3+* is given in Fig. 12.



Parameters calculated on the profile Scan5535 > Levelled

Unfiltered parameters

Pa	=	0.3 µm
Pq	=	0.369 µm
Pp	=	0.93 µm
Pv	=	0.89 µm
Pt	=	1.82 µm
Psk	=	0.0877
Pku	=	2.39
Pz	=	1.82 µm
PTp	=	56.4 % (1 µm under the highest peak)
PHTp	=	0.671 µm (20%-80%)
PSm	=	0.0444 mm
PDq	=	4.04 °
PLq	=	0.0329 mm
PLo	=	0.312 %
PPc	=	2.5 pks/mm (+/- 0.5 µm)
PzJIS	=	1.27 µm
P3z	=	1.54 µm
Pc	=	0.978 µm
Pfd	=	1.36
PHSC	=	21 peaks (1 µm under the highest peak)
PDa	=	*****
PLa	=	*****
PH	=	1.23 µm
PD	=	22.5 1/mm
PS	=	0.0276 mm
PVo	=	0.000395 mm3/mm2
Pmr	=	56.4 % (1 µm under the highest peak)
Pdc	=	0.671 µm (20%-80%)

Fig. 12 Typical roughness diagram and corresponding parameter values obtained for a selected sculptured surface profile section

From the consistency of simulated and inspected results, it can be verified that the proposed approach can improve sculptured surface machining operations in terms of both surface finish requirements and productivity. Results given in Fig. 11 may also be compared to those available by Gan et al. (2016); Rufeng et al. (2010); Warkentin et al. (2000) and Rao et al. (1997). It is indicated that the proposed approach outperforms these techniques and produces lower surface deviation as well as smaller scallops than the aforementioned multi-axis surface machining strategies.

Discussion

Tool path optimization for sculptured surface CNC machining is a well established field for which numerous effective approaches have been implemented. However their novel directions can hardly convince that a straightforward methodology indeed exists so as to provide a link between research and state-of-the art practice. Therefore, it is of paramount importance that practical strategies ought to be developed to suggest new knowledge under simple interfaces for facilitating current implementations. The environment proposed in this paper suggests the employment of a novel artificial heuristic for which no specialized knowledge is to be needed by practitioners since it operates “in-house” with existing tools such as CAM systems to generate tool paths for sculptured surfaces.

The presence of the new virus-evolutionary genetic algorithm implemented to address the problem represents a novel approach from the perspective that co-evolution among conventional and viral intelligent operators ensures a good exploration-exploitation ratio (global and local search for optimal solutions) without a formal knowledge of the objective function. Note that most heuristics ensure the balance between global and local search, yet; under the assumption that the search space is inherently continuous since a mathematical objective function exists to be optimized. The problem here is addressed without any speculation about continuity or non-continuity of the search

space. Consecutively previous intelligent approaches for solving the sculptured surface machining optimization problem have not been questioned for validating the proposed heuristic integrating the presented environment since there is a need to re-write and conjugate them to the developed open interface (API) of the CAM system to allow for rigorous comparisons. Another problem is that most of the proposed approaches do not validate their functionality against benchmark material for which results can be found from previous researchers to be compared with; they only illustrate their implementation philosophy to test cases. This substantiates the selection of the specific benchmark component to test the proposed approach. Note that comparative studies with results derived by the same problem could still be conducted if an experimental design was to be established so as to build a fitness model by regression and use it to verify all heuristics selected, yet; this direction would be less efficient based on the drawbacks of process models and is not the scope of this work.

Conclusions and future perspectives

This paper presents a novel framework for dynamically optimizing tool paths for sculptured surface CNC machining. At its core is the implementation of a new virus-evolutionary genetic algorithm for the generation and evaluation of robust tool path chromosomes. The evaluation process is fully automated under an iterative fashion towards the convergence to the optimal output that will simultaneously satisfy surface machining error and tool path efficiency. The former criterion is treated as a combined effect between chordal deviation and scallop height whereas the latter is primarily reflected by the number of CC points. Both criteria are measured using a weighted objective utility instead the formal representation of an objective function to represent the search domain. The developed framework is supported by series of novel interactive routines established in the open architecture of a cutting edge CAM system to select computational entities depending on the parameter selections and achieve cooperation with the algorithm integrating the proposed environment. Crucial parameters formulating tool paths involve the type of cutting tool; the step-over; the two 5-axis machining inclination angles (lead and tilt) and maximum discretization step. The proposed framework has been validated by employing both simulation tools as well as performing an actual simultaneous 5-axis CNC machining using a benchmark model for which a large number of outputs are available to compare with. Results for simulations by testing two algorithms were obtained via 3D tolerance analysis whilst results for actual cutting were obtained via CMM inspection. Results analysis provided enough evidence to support that the proposed methodology for optimizing sculptured surface CNC machining tool paths can guarantee optimality at a very large extent at least when it comes to the usage of computer-aided resources. Such an intelligent framework for sculptured surface CNC tool path optimization is presented for the first time in related literature whilst it is of significant contribution to industrial practice and academic research. The significance of the proposed approach is the straightforward global optimization of complex sculptured surface tool paths for maintaining productivity and quality without the necessity of extensive user experience. The work presented in this paper differs from other approaches in many aspects such as the representation of the problem handled, the automated collaborative infrastructure for producing and assessing intelligently optimized tool paths via computer programming and the direct interfacing to current industrial resources with the aid of a simple interactive environment. An important limitation of the work presented is that gouge detection when selecting cutting tools to machine sculptured surfaces is not provided. This part requires the sequential examination of local curvatures for tool postures with respect to local curvatures for CC points constituting the tool path; a function that adds a significant computational load. Nevertheless modern CAM systems provide this property in advance. Another general shortcoming is the restricted retrieval of automation objects for building interactive platforms between manufacturing software and intelligent systems. In other words the proposed methodology requires open CAM software architectures to ensure interaction among manufacturing functions and intelligent algorithms.

Looking further ahead, feed adaptation is to integrate the environment since local curvature information can be retrieved by the system. Sampling times from several CNC units are to be taken into account to develop a curvature-based feed adaptation module and a post-processor engine to translate optimal CL files for tool paths to ISO codes depending on the NC controller type and the machine tool configuration. Other future perspectives include experimental validation using other types of engineering problems and/or sculptured surfaces since it is realized that the results presented here refer to a single complex surface despite the fact that it covers most surfaces found in cutting-edge industrial applications. Nevertheless results presented here for the novel framework deemed very encouraging and quite promising with great potentials to further push the envelope of profitability and quality to support state-of-the-art practice.

Acknowledgement

The authors acknowledge the Hellenic Aerospace Industry (HAI) and senior CNC programming engineer, Mr. Mermelas Panagiotis, for providing the Fooke® Endura 9 series 5-axis CNC machining center and DEA® gantry CMM to support the research.

References

- Agrawal, R.K., Pratihari, D.K., & Roy Choudhury, A. (2006). Optimization of CNC isoscallop free form surface machining using a genetic algorithm. *International Journal of Machine Tools and Manufacture*, 46(7), 811-819.
- Bedi, S., Ismail, F., Chen, Y., & Mahjoob, M. (1996). Toroidal versus ball nose and at bottom end mills. *International Journal of Advanced Manufacturing Technology*, 13, 326-332.
- Brecher, C., & Lohse, W. (2013). Evaluation of toolpath quality: user-assisted CAM for complex milling processes. *CIRP Journal of Manufacturing Science and Technology*, 6, 233-245.
- Cai, Y., & Zhao, M. (2014). Cutter location point calculation for five-axis surface machining with ellipse cutter. *International Journal of Production Research*, 52(2), 436-444.
- Castelino, K., D'Souza, R., & Wright, P.K. (2003). Toolpath Optimization for Minimizing Airtime during Machining. *Journal of Manufacturing Systems* 22(3), 173-180.
- Chu, C.-H., & Hsieh H.-T. (2012). Generation of reciprocating tool motion in 5-axis flank milling based on particle swarm optimization. *Journal of Intelligent Manufacturing*, 23, 1501-1509.
- Duroobi, A. Mohamed, J., & Kazem, B.I. (2010). The Mathematical Description of End Mill Cutters and Effective Radius of Tool Geometry on Multi-Axis Milling. *Engineering and Technology Journal*, 28(8), 1596-1614.
- Durvedi, R.K., Bedi, S., Batish, A., & Mann, S. (2014). A multipoint method for 5-axis machining of triangulated surface models. *Computer Aided Design*, 52, 17-26.
- Fisher, R.B. (1989). *From Surfaces to Objects: Computer Vision and Three Dimensional Scene Analysis*. John Wiley and Sons.
- Fountas, N.A., Vaxevanidis. N.M., Stergiou, C.I., & Benhadj-Djilali, R. (2016). A virus-evolutionary multi-objective intelligent tool path optimization methodology for 5-axis sculptured surface CNC machining. *Engineering With Computers*, DOI: 10.1007/s00366-016-0479-5.
- Fountas, N.A., Vaxevanidis. N.M., Stergiou, C.I., & Benhadj-Djilali, R. (2014). Development of a software-automated intelligent sculptured surface machining optimization environment. *International Journal of Advanced Manufacturing Technology*, 75(5), 909-931.
- Gan, Z., Chen, Z., Zhou, M., Yang, J., & Li, S. (2016). Optimal cutter orientation for five-axis machining based on mechanical equilibrium theory. *International Journal of Advanced Manufacturing Technology*, 84, 989-999.
- Gray, P., Ismail, F., & Bedi, S. (2004). Graphics-assisted rolling ball method for 5-axis surface machining. *Computer Aided Design*, 36(7), 653 – 663.
- Kayal, P. (2007). Inverse offset method for adaptive cutter path generation from point-based surface. *International journal of CAD/CAM*, 7, 1-18.
- Krimpenis, A., & Vosniakos, G. C. (2009). Rough milling optimization for parts with sculptured surfaces using genetic algorithms in a Stackelberg game. *Journal of Intelligent Manufacturing*, 20(4), 447–461.
- Kubota, N., Fukuda, T., & Shimojima, K. (1996). Virus-evolutionary algorithm for a self-organizing manufacturing system. *Computers and Industrial Engineering*, 30(4), 1015-1026.
- Lazoglu, I., Manav, C., & Murtezaoglu, Y. (2009). Tool path optimization for free form surface machining. *CIRP Annals - Manufacturing Technology*, 58, 101-104.
- Li, L., Liu, F., Chen, B., & Li, B.C. (2015). Multi-objective optimization of cutting parameters in sculptured parts machining based on neural network. *Journal of Intelligent Manufacturing*, 26(5), 891-898.
- Lin, T., Lee, J-W., & Bohez, E.L.J. (2009). A new accurate curvature matching and optimal tool based five-axis machining algorithm. *Journal of Mechanical Science and Technology* 23, 2624-2634.
- Lin, Z., Fu, J., Shen, H., & Gan, W. (2014). An accurate surface error optimization for five-axis machining of freeform surfaces. *International Journal of Advanced Manufacturing Technology*, 71(5-8), 1175-1185.
- Manav, C., Bank, H. S., & Lazoglu, I. (2013). Intelligent toolpath selection via multi-criteria optimization in complex sculptured surface milling. *Journal of Intelligent Manufacturing*, 24(2), 349–355.
- Mellal, M.A., & Williams, E.J. (2016). Parameter optimization of advanced machining processes using cuckoo optimization algorithm and hoopoe heuristic. *Journal of Intelligent Manufacturing*, 27, 927-942.
- Paul, J., Gray, S.B., & Ismail, F. (2005). Arc-intersect method for 5-axis tool positioning. *Computer Aided Design*, 37,663-674.
- Rao, N., Ismail, F., & Bedi, S. (1997) Tool path planning for five-axis machining using the principal axis method, *International Journal of Machine Tools and Manufacture*, 37(7), 1025–1040.

- Redonnet, J.M., 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. *Computer Aided Design*, 45, 951-962.
- Redonnet, J.-M., Vazquez, A.G., Michel, A.T., & Segonds, S. (2016). Optimization of free-form surface machining using parallel planes strategy and torus milling cutter. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 1-11, DOI: 10.1177/0954405416640175.
- Rufeng, X., Zhitong, C., Wuyi, C., Xianzhen, W., & Jianjun, Z. (2010). Dual Drive Curve Tool Path Planning Method for 5-axis NC Machining of Sculptured Surfaces. *Chinese Journal of Aeronautics*, 23, 486-494.
- Sarma R. (2000a). On local gouging in five-axis sculptured surface machining using flat-end tools, *Computer Aided Design*, 32, 409-420.
- Sarma, R. (2000b). Flat-Ended Tool Swept Sections for Five-Axis Machining of Sculptured Surfaces, *ASME Journal of Manufacturing Science and Engineering*, 122, 158-165.
- Tunç, T.L. (2016). Rapid extraction of machined surface data through inverse geometrical solution of tool path information. *International Journal of Advanced Manufacturing Technology*, 87, 353-362.
- Tunç, T.L., & Budak, E. (2009). Extraction of 5-axis milling conditions from CAM data for process simulation, *International Journal of Advanced Manufacturing Technology*, 43(5-6), 538-550.
- Ulker, E., Turanalp, M.E., & Halkaci, H.S. (2009). An artificial immune system approach to CNC toolpath generation. *Journal of Intelligent Manufacturing*, 20, 67-77.
- Vickers, G.W., & Quan, K.W. (1989). Ball-mills versus end-mills for curved surface machining. *Journal of Engineering for Industry*, 111, 22-26.
- Warkentin, A., Ismail, F., & Bedi, S. (2000). Multi-point tool positioning strategy for 5-axis machining of sculptured surface. *Computer Aided Geometric Design*, 17(1), 83-100.
- Yildiz, A.R. (2009a). A novel hybrid immune algorithm for global optimization in design and manufacturing. *Robotics and Computer Integrated Manufacturing*, 25(2), 261-270.
- Yildiz, A.R. (2009b). A novel particle swarm optimization approach for product design and manufacturing. *International Journal of Advanced Manufacturing Technology*, 40(5-6), 617-628.
- Yildiz, A.R. (2012). A comparative study of population-based optimization algorithms for turning operations. *Information Sciences*, 210, 81-88.
- Yildiz, A.R. (2013a). A new hybrid differential evolution algorithm for the selection of optimal machining parameters in milling operations. *Applied Soft Computing*, 13(3), 1561-1566.
- Yildiz, A.R. (2013b). Cuckoo search algorithm for the selection of optimal machining parameters in milling operations. *International Journal of Advanced Manufacturing Technology*, 64(1-4), 55-61.
- Yildiz, A.R. (2013c). Hybrid Taguchi-differential evolution algorithm for optimization of multi-pass turning operations. *Applied Soft Computing*, 13(3), 1433-1439.
- Yusup, N., Sarkheyli, A., Zain, A.M., Hashim, S.Z.M., & Ithnin, N. (2014). Estimation of optimal machining control parameters using artificial bee colony. *Journal of Intelligent Manufacturing*, 25, 1463-1472.
- Zainal, N., Zain, A.M., Radzi, N.H.M., & Othman, M.R. (2016). Glowworm swarm optimization (GSO) for optimization of machining parameters. *Journal of Intelligent Manufacturing*, 27, 797-804.
- Zeroudi, N., Fontaine, M., & Necib, K. (2012). Prediction of cutting forces in 3-axes milling of sculptured surfaces directly from CAM tool path. *Journal of Intelligent Manufacturing*, 23(5), 1573-1587.
- Zhang, J. Y., Liang, S. Y., Yao, J., Chen, J. M., & Huang, J. L. (2006). Evolutionary optimization of machining processes. *Journal of Intelligent Manufacturing*, 17(2), 203-215.