

# Generating Multiaxis Tool Paths for Die and Mold Making with Evolutionary Algorithms

Klaus Weinert and Marc Stautner

Department for Machining Technology Baroper Str. 301,  
44227 Dortmund, Germany  
{Weinert, Stautner}@isf.de  
<http://www.isf.de>

**Abstract.** Defining tool paths for multiaxis milling is a difficult task that requires knowledge about the process and the behavior of the machine tool. Because of the high risks of multiaxis movements within a real machine, the resulting paths are safe but normally sub-optimal. This article introduces a new method to obtain tool paths, which are free of collisions and optimal in consideration, the machine and its kinematics. The proposed method combines evolutionary algorithms and a newly developed efficient multiaxis milling simulation. The evolutionary algorithm uses multiple criteria for the optimization of a tool path. These criteria are minimized axial movement of the machine, minimal cutting forces and the exclusion of collisions between the workpiece and the non-cutting parts of the cutting tool. The approach uses multi-objective algorithms because minimal or no movement of a machine axis can lead to high cutting forces or multiple collisions. Along with the evolutionary algorithm, this article introduces a simulation for multiple axis milling. This simulation is able to compute the fitness of the given tool paths by considering the engagement condition and collision situation of a cutting tool.

## 1 Problem Description

In multiaxis milling of free formed surfaces for die and mould making, one of the main tasks is a near optimal definition of the tool path. This definition is placed under various constraints, which need to be fulfilled simultaneously. One point is the avoidance of collisions between the non-cutting parts of the cutting tool and the work piece. A second one that is exclusive to multiaxis milling is the task of finding a tool orientation that fulfills all constraints resulting from the used machine. Different milling machines have different constraints imposed upon the movement of their additional axis, which are needed for the more than three axis movement. This can be seen as an additional collision condition between tool and possible movement area of the machine in addition to the collision between tool and free form surface.

The third constraint is not exclusively given in multi axis milling; it is also valid for all milling processes. In milling processes, the cutting force on the cutting edge is

an important factor for a stable and safe process control. The programming of tool paths must lead to a process that is fast and safe to be cost efficient.

Resulting from these constraints multiaxis tool path strategies are often designed as an addition to the existing three axis strategies. These paths are planned as a three-axis movement with an additional two-axis movement, which changes the orientation of the tool to the work piece. This additional rotational movement has to be free of collisions. Therefore, plenty of user experience is needed to design these tool paths. There are some software tools on the market that deliver some kind of collision detection but they are often restricted to three axis milling or only to collision detection against the final geometry and not the actual process geometry of the free form surface. So there is a demand for solutions which provide functions to develop tool paths which are free of collisions and use the possibilities of multi axis tool path programming to gain a fast and safe process course. This article introduces a first prototype of a tool that enables the user to convert given three axis tool paths to collision free multiaxis tool paths.

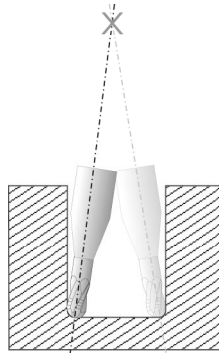
## 2 General Idea

The main concept consists of two steps. In the first step, a user starts designing a conversion solution for a given three-axis tool path and a given workpiece. In a second step, a software tool optimizes the user's parameters considering all constraints and optimizing facilities. After this second step, the resulting tool path leads to an efficient multi axis milling process, which is free of collisions and fulfills all machine restrictions.

The general idea for converting three axis milling paths into multiaxis paths, is to keep the position of the tool tip and to add a varying the orientation to the tool. The resulting surface remains the same when only ball end mills are used. Only the quality of the surface will improve resulting from the more advantageous engagement conditions. Several studies have shown that this simple tool tilt strategy leads to useful and efficient multi axis paths with minimal amount of manual user input.

### 2.1 Manual Step

The user determines a tilt point in 3D space. The multiaxis path is then generated by tilting the tool so that the main axis of the tool, shank and holder lead through an axis determined by the tilt point and the point on the tool path. This strategy is illustrated in Fig. 1. This tilt point strategy has several benefits: it is easy to develop and the resulting tool paths lead to smooth movements of the machine axis, because these movements can often be executed by tilting in one machine axis and then by rotating the second machine axis. One disadvantage is the indirect change of the tool paths by changing a point in 3D space. It is easy to find or see a useful position for this point, but it is difficult to determine that this point is the optimal one that leads to a minimal movement of the tool axis and is free of collisions.



**Fig. 1.** Strategy for generating a tilt position for every cutter location by tilting through an axis given by a point in 3D space and the cutter position

## 2.2 Evolutionary Step

The optimization of this tilt point is done in the second step of the strategy by an evolutionary algorithm. The evolutionary strategy benefits from the factor that only a few (three) parameters need to be evolved for the tilt point. For further implementations of this technique the optimization of more than one tilt point is planned. Additionally the optimization of further parameters such as the form of the area in which the orientation is changed by one tilt point should be tested.

The difficulty in this evolutionary step lies not in the number of parameters. The main problem of this evolutionary algorithm is a fast computation of the fitness function. Therefore a new system for the simulation of five-axis milling has been developed. This simulation provides a tool to analyze the real process quality of the evolved individuals.

## 3 Techniques

The inputs in the process chain are a three axis-milling path with additional user parameters, these parameters are an area in which the path has to be tilted and an orientation of the tilting axis. After that, the evolution step searches for the optimal additional parameters, here the position of the tilt point and evaluates the fitness. The outputs of this process are optimized tilting positions and accordingly efficient multi-axis tool paths. The schematic view on this process chain is shown in Fig. 2.

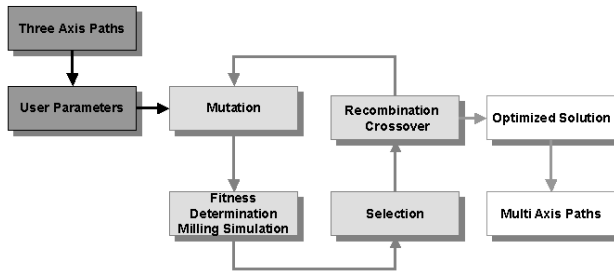


Fig. 2. Proposed two step process chain

### 3.1 Discrete Milling Simulation

Several products on the market allow a simulation of milling processes. Most of them [1] are not capable of simulating multiaxis milling processes. There are simulations that can perform this task but these techniques contain disadvantages, which prevent utilization for the fitness determination.

There is the group of exact simulation approaches. Kawashima [2] describes an accurate technique called “Grafttree” to speed up a solid modeling approach. The workpiece and the arbitrarily shaped tool are modeled by a CSG-tree. Constructive Solid Geometry (CSG) is an object-construction method used in several 3D-CAD systems. Objects are basically constructed by combining simple solid objects with Boolean operators, such as union or intersection as described by Foley et al. [3]. Sourin and Pasko [4] developed another approach; they simulate the cutting tool analytically with the use of procedurally defined time-dependent defining functions. Both approaches, the “Grafttree” and the one made by Sourin and Pasko are exact but time consuming.

A second group of techniques is the approximate approaches. One technique is the polygonal approach described by Glaeser [5]. Here a polygonal representation of the workpiece, called “T-buffer”, was processed with a polygonal representation of the swept volume of the tool, which was determined by differential geometry. This approach can lead to large mesh sizes that are reduced for display by a recursive subdivision scheme.

Another group of approximate techniques is the “dixel” approach. Hook [6] converts the workpiece into a discrete “dixel structure” which is derived from the z-buffer approach in computer graphics [3]. One problem of Hook’s approach is that, similar to the z-buffer algorithm, the dixel structure has to be aligned with a viewing vector. Therefore, after changing the viewing position, the whole simulation process has to be redone. A determination of fitness values through the simulation is not possible.

Extensions to this approach were made by Huang [7] and Weinert [8]. These extensions are independent of the viewer’s position and process the input NC-file by modeling the cutting operation at each discrete position on a given path. To achieve

high accuracy with low memory usage, these discrete simulations do not have the ability to simulate undercuts, thus they are only able to simulate a three-axis milling process.

For the usage in these evolutionary algorithms a new simulation of milling processes based on these dextral techniques has been developed. This new simulation allows free scalable level of simulation resolution to reduce simulation time and provides a fast collision testing and extra interface functions to derive a fitness of simulated tool path. The simulation is able to detect and benchmark the amount of collisions and to take account of the restrictions resulting from the different mechanical capabilities of the used machine.

### 3.2 The Evolutionary Algorithm

As shown by Castelino et al. [9] evolutionary algorithms, which are used in process planning for nc-machining are primarily, specialized in the optimization of general tool movement. These solutions refer to algorithms developed for the traveling salesman problem or the lawnmower problem. Due to their large non-linear search spaces, both problems are often used for evolutionary algorithm. An overview on this subject is presented by Alander [10]. The algorithms in this area of problems do not focus on the machine abilities and the engagement conditions.

In our algorithm, the problem that needs to be solved is a simple three dimensional search. Therefore we can use a rather simple  $(\mu, \lambda)$  evolutionary strategy ES. This leads to good results without optimized evolutionary parameters. This is a consequence of the simple three dimensional evolution problems, which have to be solved. A tournament selection operator defines two groups of individuals that compete with each other. Every survivor of a tournament replaces the inferior individual. A survivor of a tournament is the individual that has a higher fitness value compared to its competitor.

As is shown in the results section, simple problems can be solved in less than 20 generations with values for  $\mu$  of 50 and a  $\lambda$  of 25. This behavior is important the usefulness of the introduced system. The determination of the fitness of a single individual can be very time consuming. Milling processes have real running times of up to 20 hours. State of the art milling simulations can be 10 times faster than the real process. So defining a fitness of a single individual phenotype can be very time consuming.

#### 3.2.1 Fitness Determination

The fitness function uses three different types of fitnesses that were determined separately and result from different constraints and collisions.

1. Collision with the space bound by not reachable axis values of the used machine.
2. Collisions of the workpiece with the non-cutting parts of the tool such as shank or holder. All collisions in a simulation run are added up to a single value. This value shows how much volume of the material collides with the tool. Therefore,

slight touching collisions result in different values as full collisions. This is a measurement for the amount of collisions with the machine.

3. The third value is the amount of harmony in the resulting movement of the machine. The simulation returns a value that represents how easy the machine may follow the suggested movement. This value has a direct influence on the real process milling times and the resulting surface quality.

These three values represent multiple criteria for the fitness determination. A combination is used for the selection of the individuals.

### 3.2.2 Selection Method

The algorithm uses a truncation or  $(\mu, \lambda)$  selection [11]. The selection schema for the  $\lambda$  best individuals is in pseudo code.

```
for (c=0;c<[NumberOfFitnessValues];c++){
    if (I1.f[c] > I2.f[c]) return true;
    if (I1.f[c] < I2.f[c]) return false;};
```

Where I1 and I2 are the current individuals with their respective arrays of fitness values. In our algorithm, there are three different fitness values. The ranking depends on the order of the fitness values in the array. In our algorithm the order is

1. machine constraints,
2. amount of collisions and
3. movement harmony.

Therefore, at first a tool path needs to be free of both types of collisions and after that the algorithm searches for the position that leads to the most harmonic movement of the machine axis.

### 3.2.3 Recombination and Crossover

The Recombination is done via a two individual crossover. Two individuals are randomly chosen from  $\lambda$  selected ones (see Sect. 3.2.2) ignoring their fitnesses. The crossover randomly chooses values from both individuals. This crossover reduces the fitness pressure so that searches in the solution space are more efficiently.

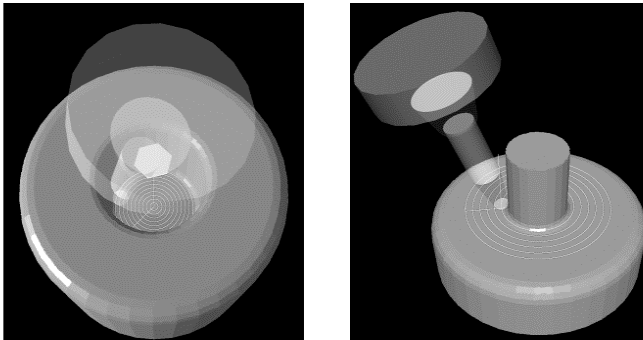
## 4 First Results

First tests were carried out on specially designed workpieces, which allow a fast and simple design of the evolutionary algorithm. The use of these test cases allows verifying the design of the EA and helps to reduce unwanted time losses from errors in the implementation of the algorithms. The test workpieces are designed with the CAD/CAM software CATIA to ensure the functionality of the complete process chain.

## 4.1 Test Workpieces

The tests should verify the algorithm on both opposed possibilities that occur in real workpieces. Therefore, the first workpiece is a deep cavity. This is a typical form in die and mould making. Cavities are difficult to mill if they are of a certain depth and a small diameter, because this necessitates tools with long thin tool shanks. The use of long shanks causes undesirable effects on the surface quality and increases the high risk of tool breakage. To overcome these effects the process speeds need to be reduced. The second workpiece is a dome that has a steep rising edge in the middle of the workpieces Fig. 3. (right). This workpiece needs to be milled with long shank as well to enable the processing of the small diameter (1 mm) fillet at the base of the dome.

Although both test workpieces have their ideal tool tilt point in the center of the workpieces they have different constraints. The collision spaces are completely different so that the collision avoidance leads to different demands on the evolution progression. The simulation can only deliver a value of how many collisions have occurred. As a consequence the fitness landscapes are different, and a parameterization which is suitable for both cases should be suitable for real workpieces as well.



**Fig. 3.** Test workpieces and their tooling paths. Negative Cavity (left), Positive Dome (right)

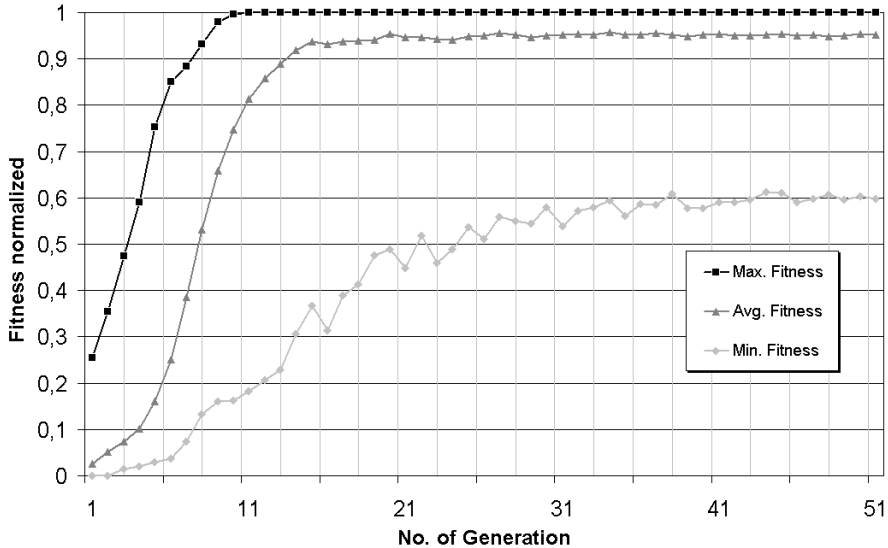
### 4.1.1 Tool-Shank-Holder

The employed, simulated tool is a ball end mill with a diameter of 2 mm and an edge length of 1 mm. This part of the tool cannot lead to collisions. To ensure an increased surface quality after the evolution a shorter, thicker tool shank was used for the evolution. This tool cannot be used on the original three axis paths. The colliding part of the tool is the shank, which has a cone as basic form and the holder. In our test workpiece, only collisions between workpiece and shank are to be expected because of the length of the shank.

### 4.1.2. The Cavity Workpiece

The initial tool paths for the cavity are some short finishing paths on the ground of the cavity (see Fig. 3). At this point of the milling process, the chance of a collision is most likely. The milling of these paths is not possible with a three-axis strategy and the chosen shank-holder combination.

The initial search space for the positioning of the tool points is from -40 mm to +40 mm along all axes of the global object space. There are 50 initial individuals that are spread stochastically over this 80x80x80 mm search space. To obtain an overview of the behavior of the algorithm this test run is repeated 20 times.



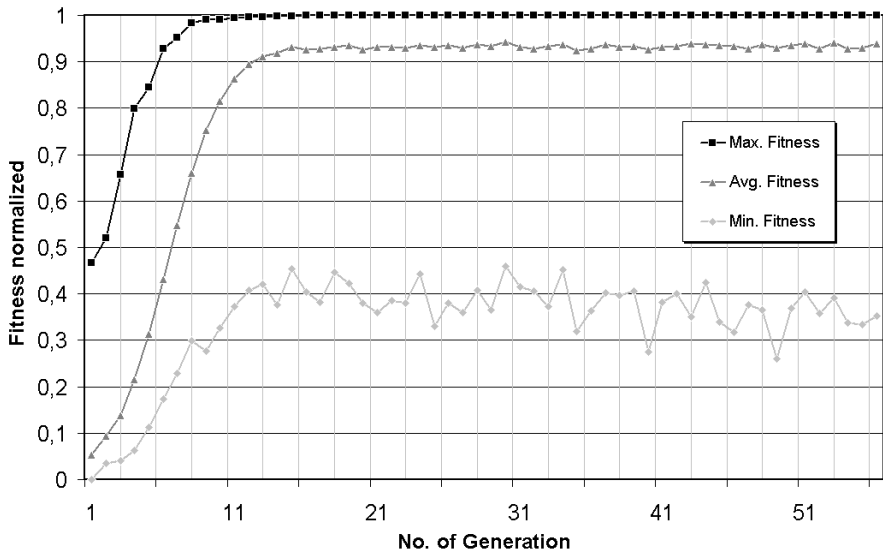
**Fig. 4.** Average fitnesses from 20 evolution runs with 50 individuals in each population. (Cavity Workpiece)

The diagrammed fitness in Fig. 4. is the accumulated, normalized fitness of both collision constraints. As can be seen, the maximal fitness reaches 1 in approx. 10 generations. This shows that collision free tilt points can be reached and there is enough time for a more difficult search for the optimal tool point.

### 4.1.3 The Dome Workpiece

The tool path on the second workpiece is different to the path introduced before. The collision here is most likely at the inner paths where edge of the dome is approached, see Fig. 5. Because of the similar dimensions of the workpiece all parameters remain the same. This allows obtaining an overview of the behavior of the algorithm on different parts without the use of extra knowledge.





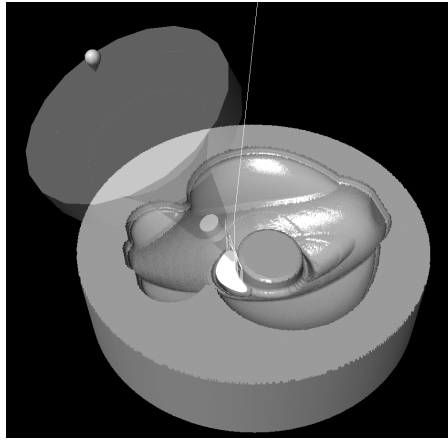
**Fig. 5.** Average fitnesses from 20 evolution runs with 50 individuals each generation. (Dome Workpiece)

As it can be seen in Fig. 5, the general behavior of the algorithm is similar a collision free position is found in approx 10 generations. After that the fitness is determined by the harmony fitness value.

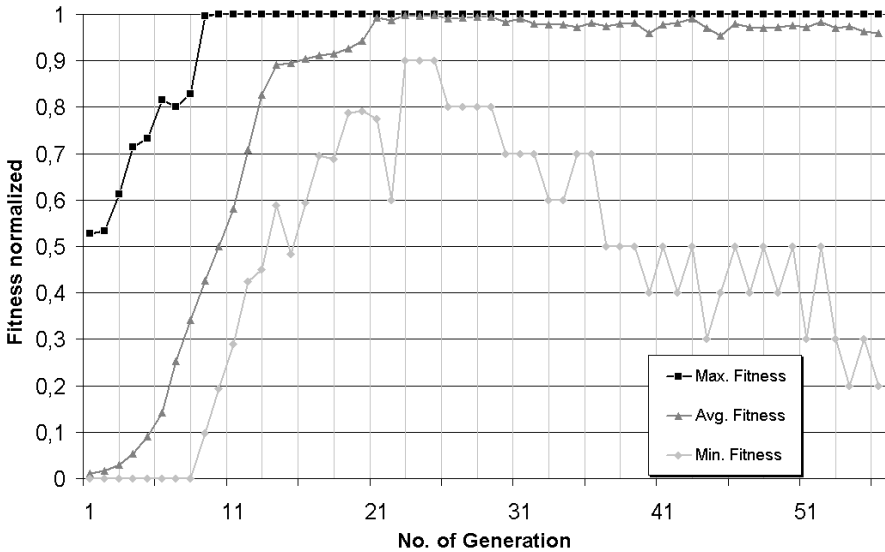
## 4.2 Real Workpiece

The pre-parameterized algorithm is now used for a real workpiece. The chosen workpiece and the three axis milling paths are not specially designed for five-axis milling or for this evolutionary path design. The workpiece originates from the European High Speed Machining Award 2000. To preclude a three axis processing a shank based on a cone is used. Surface quality is expected to be better if the process is carried out with the shorter and sturdier conical shank, the present three-axis tool paths will lead to a collision if they are used with this tool, though.

As a test-processing step, a finishing step was chosen. This finishing step consists of approx. 65000 single lines of nc-code. As in the test runs, 50 individuals per population are used. The initial search space for these tests is 200mm x 200mm x 200mm according to the larger dimensions of the workpiece (160mm x 160mm x 40mm). The tool paths can be seen in Fig. 6.



**Fig. 6.** Workpiece, from the European High Speed Machining Award 2000



**Fig. 7.** Average fitnesses from 10 evolution runs with 50 individuals each generation

Fig. 7. shows a fitness plot for 10 subsequent test runs. Each run consist of approx. 50 generations with 50 individuals in each generation. In this plot only the collision (workpiece tool and tool machine space) fitnesses are combined. The path of the first curve which is the average maximal fitness of an individual shows that a collision free position is found by the algorithm on average within the first 10 generations of the evolution. Until generation 20 the other values for average fitness and for minimal

fitness increase accordingly. After generation 20 the average minimal fitness decreases. The reason for this behavior lies in the machine constraints: the collision free tilt point is close to the edge of the region reachable by the machine. The algorithm draws all individuals close to this border and therefore a minimal movement of the tool point leads to tool paths that cannot be realized by the machine kinematics. The fitness of these tool points is set to zero. Therefore, the more points reach the optimal point that is close to this edge the more points can accidentally jump over. This is the reason for the slightly decreasing average fitness.

## 5 Conclusions and Outlook

This paper presents a new system for the optimization of tool paths and engagement conditions for multiaxis milling as well as the generation of efficient multiaxis tool paths. First studies show that the proposed multi criteria evolutionary algorithm is suitable to deliver an efficient evolution in combination with an efficient simulation system. The simple parameterization of the algorithm allows the application of the system on different problem cases without further user knowledge about the evolutionary parameters. Although the introduced software system is an academic prototype, this qualifies enables this technique as basis for a commercial solution for multiaxis path generation in CAM applications.

Further implementations are planned based on other generation strategies and require a more complex genome. More advanced multiaxis strategies may be developed with this system without the use of three axis tool paths as basis for the evolution.

**Acknowledgements.** The authors like to thank the Deutsche Forschungsgemeinschaft for financial support as part of the Collaborative Research Center “Computational Intelligence” (SFB 531).

## References

1. Müller, H.; Surmann, T; Stautner, M; Albersmann, F.; Weinert, K.: Online Sculpting and Visualization of Multi-Dexel Volumes. In: SM'03, Eighth ACM Symposium on Solid Modeling and Applications, Elber, G.; Shapiro, V. (eds.), Seattle, Washington, USA, ACM 1-58113-706-0/03/0006, ACM Press, NY, USA, June 16-20, S. 258-261
2. Kawashima, Y; Itoh, K.; Ishida, T.; Nonaka, S; Ejiri, K.: A flexible quantitative method for NC machining verification using a space-division based solid model. *The Visual Computer* 7 (1991), pp. 149-157
3. Foley, J. D.; Dam, A. van; Feiner, S. K.; Hughes, J.F.: *Computer Graphics: Principles and Practice*. Addison-Wesley, New-York 1992
4. Sourin, A.I.; Pasko A.A.: Function Representation for Sweeping by a Moving Solid. *IEEE Transactions on Visualization and Computer Graphics* 2 (1996) 2, pp. 11-18

5. Glaeser, G.; Gröller, E.: Efficient Volume-Generation During the Simulation of NC-Milling. Technical Report TR-186-2-97-10, April 1997, Institute of Computer Graphics and Algorithms, Vienna University of Technology
6. Hook, T. van: Real Time shaded NC Milling Display. *Computer Graphics* 20 (1986) 4, pp. 15-20
7. Huang, Y.; Oliver, J.H.: NC Milling error Assessment and Tool Path Correction. In: *Computer Graphics Proceedings, Conference Proceedings July 19-24 (1994)*, pp. 287-294
8. Weinert, K.; Müller, H.; Friedhoff, J.: Efficient discrete simulation of 3-axis milling for sculptured surfaces. *Production Engineering* 3 (1998) 2, pp. 83-88.
9. Castelino, K.; D'Souza, R.; Wright, P. K.: Tool-path Optimization for Minimizing Airtime during Machinig, Mechanical Engineering Dept. University of California at Berkeley, [http://kingkong.me.berkeley.edu/~kenneth/research/pubs/airtime\\_minimization\\_joms.pdf](http://kingkong.me.berkeley.edu/~kenneth/research/pubs/airtime_minimization_joms.pdf)
10. Alander, J., T.: An Indexed Bibliography of Genetic Algorithms and the Traveling Salesman Problem. Internal Report 94-1-TSP, University of Vaasa, Department of Information Technology and Production Economics, Finland
11. Banzhaf, W; Nordin, P; Keller, R. E.; Francone, F. D.: *Genetic Programming : An Introduction : On the Automatic Evolution of Computer Programs and Its Applications*. Morgan Kaufmann Publishers, Inc. San Francisco, Cal.