THE APPLICATION OF MULTIOBJECTIVE EVOLUTIONARY ALGORITHMS TO THE GENERATION OF OPTIMIZED TOOL PATHS FOR MULTI-AXIS DIE AND MOULD MAKING

K. Weinert, J. Mehnen, M. Stautner
Department for Machining Technology Baroper Str. 301, 44227 Dortmund, Germany

Abstract
The generation of tool paths for the milling of dies and moulds requires immense knowledge about the process behavior. This high dimensional geometrical optimizing problem requires technical knowledge and user experience. In this paper, first approaches using evolutionary algorithms for automating tool path generation and optimizing given tool paths are shown. The proposed method combines evolutionary algorithms and a newly developed efficient multi-axis milling simulation. The evolutionary algorithm uses multiple criteria for the optimization of a tool path. The approach uses multiobjective algorithms because minimal or no movement of a machine axis can lead to high cutting forces or collisions.

Keywords:
Simulation, Cutting Process, Multiobjective Optimization

1 PROBLEM DESCRIPTION

In multi-axis 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 that need to be fulfilled simultaneously. One aspect is the avoidance of collisions between the non-cutting parts of the cutting tool and the work piece. A second aspect, which is limited to multi-axis milling, is the task of finding a tool orientation, which fulfills all constraints resulting from the used machine. Different milling machines have different constraints imposed upon the movement of their additional axes, which are needed for the more than three-axis movement. This can be seen as another 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 limited to 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, which is fast and safe, in order to be cost efficient.

Resulting from these constraints, multi-axis 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, extensive 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 to the actual process geometry of the free-form surface. So there is a demand for solutions providing functions to develop tool paths which are free of collisions and to use the possibilities of multi-axis tool path programming to obtain a fast and safe process course. This article introduces a first prototype of a tool, which enables the user to convert given three-axis tool paths into collision free multi-axis 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 the 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 that is free of collisions and fulfills all machine restrictions.

The general idea for converting three-axis milling paths into multi-axis paths is to keep the position of the tool tip and to add a varying of the orientation to the tool. The resulting surface remains the same if only ball end mills are used. Only the quality of the surface will improve, resulting from the more advantageous engagement conditions. Several studies [1] have shown, that this simple tool tilt strategy leads to useful and efficient multi-axis paths with a minimal amount of manual user input.

2.1 Manual Step

The user determines a tilt point in three-dimensional space. The multi-axis path is generated by tilting the tool, so that the main axis of the tool, the shank and the holder lead through an axis determined by the tilt point and the point on the tool path. (see fig. 1.)

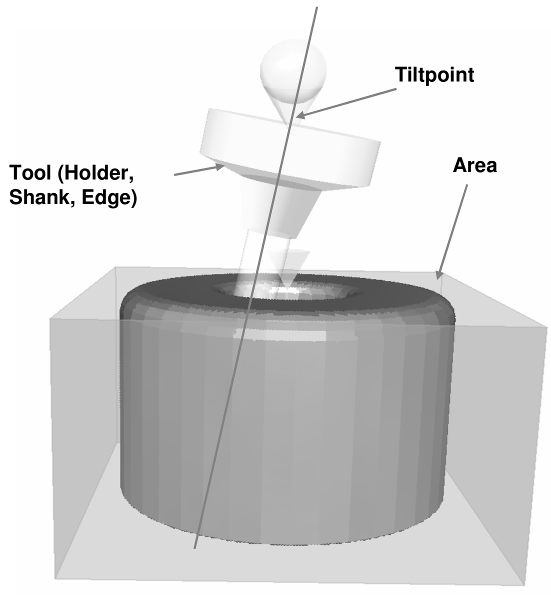

Figure 1: Scheme of the tilt-point strategy.

The user has to define an area, in which the tool paths have to be converted. By defining these areas, it is possible to define more than one tilt point on one tool path. This allows the conversion of tool paths for more complex workpieces. Tilt points can be defined as positive or negative. If a tilt point is defined positive, the line between tilt point and tool tip is drawn through the holder and shank of the tool. If it is defined negative, this line is drawn, so that the holder points away from the line. This strategy allows the manipulation of the tool orientations in a very easy way.

This tilt point strategy has several benefits. It is easy to develop, and the resulting tool paths lead to smooth movements of the machine axes because these moves can often be executed by tilting in one machine axis and rotating the second machine axis afterwards. One disadvantage is the indirect change of the tool paths by changing a point in three-dimensional space. It is easy to find a useful position for this point, but it is difficult to determine if this point is optimal, leading to a minimal, collision free movement of the tool axis.

3 EVOLUTIONARY STEP

The optimization of this tilt point is accomplished in the second step of the strategy by an evolutionary algorithm. The evolutionary strategy benefits from the fact, 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 is not 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.

3.2 Techniques

The input in the process chain is a three-axis milling path with additional user parameters. These parameters are an orientation of the tilting axis and an area, in which the path has to be tilted. Thereafter, in the evolution step the optimal additional parameters are searched. These additional parameters consist in this case only of the position of the tilt point. The output of this process is an optimized tilting position and the accordingly efficient multi-axis tool path. The schematic view on this process chain is shown in fig. 2.

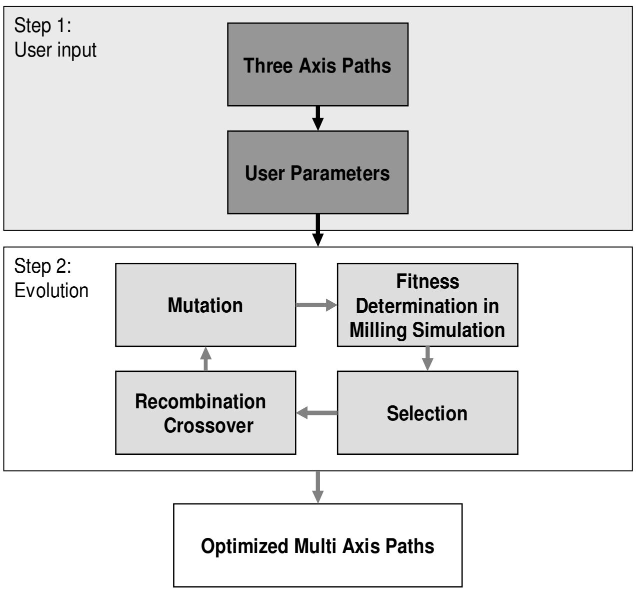

Figure 2: Proposed two-step process chain

4 DISCRETE MILLING SIMULATION

There are several simulations, which allow a simulation of milling processes, available on the market. Most of them [2] [3] are not capable of simulating multi-axis milling processes. There are simulations which can perform this task, but these techniques contain disadvantages, which prevent a utilization for the fitness determination [4] [5].

For the usage in these evolutionary algorithms a new simulation of milling processes based on these dexel [6] techniques has been developed. This new simulation allows a free scalable level of simulation resolution to reduce simulation time and provides a fast collision testing and extra interface functions to derive a fitness value for the simulated tool path. These functions deliver values for the multiple criteria, which contribute to the final fitness of the individual. The individuals implemented here consist of a single tilt point or its according tool path. The simulation, in three different sub-simulations, determines these values.

4.3 Collision Detection

The first sub-simulation is a collision detection simulation. Collisions between parts of the tool and the actual workpiece geometry can be detected. The parts of the tool are the holder or the shank of the cutter. The workpiece geometry is the actual geometry of the workpiece at the process time when the collision is tested. The simulation is able to determine a value, which indicates how much of the two volumes are in collision. This enables the optimization algorithm to distinguish between different collisions.

The other type of collision is a value that determines if the movement of the tool collides with the space bound by not reachable axis values. If such a collision occurs, it is impossible for the machine to reach the desired tool position and since the desired position is not reachable, the desired shape cannot be produced with this tool path. This collision value is more difficult to use for an optimization because the value indicates only a collision not a ratio between collision and not collision.

4.4 Machine Simulation

The second sub-simulation is a simulation of the machine capabilities. Every real milling machine, which is capable of five-axis milling, has different acceleration capabilities relating to the rotational axes. The simulation calculates a value, which indicates how well a specified machine can follow the given tool path, or how harmonic the real machine movement is compared to the desired movement. The simulation returns a harmony value, which represents how easily the machine may follow the suggested movement. This value has a direct influence on the real process milling times and the resulting surface quality. First preceding investigations have shown, that the ability of the **A** axis of the machine kinematics to follow the given positions is a good indicator for this value. Therefore, a simulation of the acceleration abilities of this axis delivers the difference of the desired axis position in relation to the real axis position. The second rotational axis, the **C** axis, has a lesser influence on the resulting surface because in many machine kinematics the **C** axis is of a much higher dynamic as the **A** axis. Thus, this optimization focuses on the two rotational axes. The ability of the machine to follow the three translating axes does not take this value in account.

4.5 Engagement Condition

The third sub-simulation models the actual engagement condition of the cutting edge on the workpiece. For this, the simulation calculates the actual chip volume for every simulated cut on the workpiece. This chip volume allows determining different properties of this engagement condition. Possible properties are the actual cutting force on the cutter or the type of engagement, which can be tangential feed or the amount of drill cutting. This computed cutting force is not an appropriate criterion for an evolutionary optimization. Because the actual cutting force is based on a model, which only takes the chip volume into account, it cannot be altered by changing the orientation of a ball nose mill.

For the evolution, this part of the simulation calculates the centricity of the cut, a value that indicates how much of the material is cut by the center of the cutter during the cutting operation. This value changes if the cut is done with a different angle without changing the surface of the workpiece or the actual cutting volume. Decreasing this value leads to an engagement of the outer areas of the cutting edges and thereby leads to a better surface quality.

These different values, which can be retrieved by the simulation, can be used for an optimization of the tilt point positions, which considers the characteristics of the resulting tool path. This optimization of a three-dimensional vector is done by a simple evolutionary algorithm.

5 THE EVOLUTIONARY ALGORITHM

As shown by Castelino et al. [7], 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 algorithms. Alander [8] presents an overview of this subject.

In this algorithm, the problem that needs to be solved is a simple three-dimensional search. Therefore, we can use a rather simple (**μ**, **λ**) ES. This leads to good first results without optimized evolutionary parameters and the general usability of this idea can be shown. 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 it is shown in the results section, simple problems can be solved in less than 20 generations with values for **μ** of 50 and a **λ** of 25. This behavior is important for 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. Thus, defining the fitness of a single individual phenotype can be very time consuming.

5.6 Fitness Determination

The fitness of a single individual is determined by the properties evaluated by the simulation of the resulting tool path. Not all criteria, which can be evolved, can be used for an evolutionary optimization. Some values cannot be influenced by a change in the orientation of the cutter. According to this, there are seven criteria, which contribute to the fitness evaluation:

1. Collision with the space bound by non-reachable machine positions (see sect. 4.3)
2. The size of the volume, which in collision between tool and the actual workpiece geometry summed up over the whole simulation process (see sect. 4.3)
3. The average engagement centricity of the cutting edge over the whole process (see sect. 4.5)
4. The average harmony of the ***A*** axis movement (see sect. 4.4)
5. The minimum of harmony in the ***A*** axis movement (see sect. 4.4)
6. The average harmony of the ***C*** axis movement (see sect. 4.4)
7. The minimum of harmony in the ***C*** axis movement (see sect. 4.4)

The first two criteria obviously have a different priority than the five other criteria. A desired tool path, which cannot be processed because positions are unreachable, does not need any further optimization of the remaining criteria. Thus, the other values cannot be evaluated in case of such a collision. They will be set to a fictive low value. In every other case, the values 2 to 7 will be evaluated simultaneously.

5.7 Order and Classification

As it can be seen in section 5.6, the fitness determination and the resulting values suggest a ranking of the objectives. A collision free path has higher priority as a harmonic movement. Ehrgott [9] calls an optimization lexicographic when comparing objective vectors in criterion space.

$$f(x_{\min}) = [f(x_1), f(x_2), f(x_3), f(x_4),...] \to \min_{x \in X} \qquad (1)$$

The function f(x) is called lexicographically smaller than or equal to the function f(y) if either f(x_n) = f(y_n), or there exists a number (l,1 <= n), such that f(x_i) = f(y_i), for 1 <= i <= l-1 and f(x_l) < f(y_l). The values retrieved from the simulation form a vector of seven dimensions. A lexicographic ordering of the vectors, which specify the fitness of a single individual, orders the individuals according to the given priority of the attribute. In this case reducing the collisions has a higher priority than decreasing the engagement centricity. This ordering allows implementing a simple selection method.

5.8 Selection Method

The algorithm uses a truncation or (**μ**, **λ**)-selection [10]. In this selection scheme, **λ** individuals are chosen from **μ** individuals if their lexicographic ordering is higher than from the **μ**-**λ** individuals. The selection scheme for the **λ** best individuals and their lexicographic classification (see Section 5.7) is shown in pseudo code in fig. 3, where I1 and I2 are the current individuals with their respective arrays of fitness values.

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

Figure 3 : Pseudo code for the evolutionary algorithm.

The criteria ranking depends on the order of the fitness values in the array. Therefore, a tool path is first tested for absence of collision. Secondly, a position with the most favorable engagement condition is searched. Only after that, the most harmonic movement of the machine axis is considered.

5.9 Recombination, Crossover and Mutation

The recombination is carried out via a two-individual crossover. Two individuals are randomly chosen from the **λ** selected ones, ignoring their fitnesses. The crossover function randomly chooses values from both individuals. This type of crossover reduces the fitness pressure. Therefore, the space of possible solutions can be searched by a more divergent population. The mutation is carried out after the recombination on all **μ** individuals of the new generation. The function for the calculation of the new parameter values is shown in equation (2), where ***f*** is the standardized fitness value of a single criterion.

$$x_{new} = x_{old} + m_1 r_1 (1 - f_1) + m_2 r_2 (1 - f_2) + m_3 r_3 (1 - f_3) + ... \quad (2)$$

The amount of mutation on a value ***x*** is based on ***n*** random values r_i. They are chosen for each fitness value in the attribute vector. Each value is combined with a factor m_i. This factor controls, how much a single deviation of a criterion can influence the search behavior of the complete algorithm. Due to the lexicographic ordering, it is helpful to choose a vector of the m_i values with $m_i >= m_{i+1}$ and $1 <= i <= n$.

6 FIRST RESULTS

First tests were carried out on a simple test workpiece, which allows a fast and plain design of the evolutionary algorithm. The test workpiece is designed with the CAD/CAM software CATIA to ensure the functionality of the complete process chain without using real workpieces in the first tests. The usage of this test case allows verifying the design of the EA and helps to reduce unwanted time losses by errors in the implementation of the algorithms. After the usefulness of the algorithm and the new developed milling simulation has been shown, the pre-parameterized algorithm is used for two real workpieces.

6.1 Tool-Shank-Holder

The used simulated tool is a ball end mill with a varying edge length of 1 mm up to 4 mm and a diameter of 1 mm to 8 mm depending on the workpiece. This part of the tool cannot lead to collisions. All tool paths, which are used in the test runs, originate from three-axis strategies. To ensure an increased surface quality after the conversion and evolution, a shorter and thicker tool shank is used. This tool cannot be used on the original three-axis paths. The colliding parts of the tool are the holder and the shank, which has a cone as basic form. In our test workpiece, only collisions between the workpiece and the shank are to be expected because of the length of the shank. Therefore, the distance between the point on the path and the holder is too big to get the holder into collision.

6.2 Column Workpiece

The artificial test workpiece is a column that has a steep rising edge in the middle of the workpiece, see fig. 4.

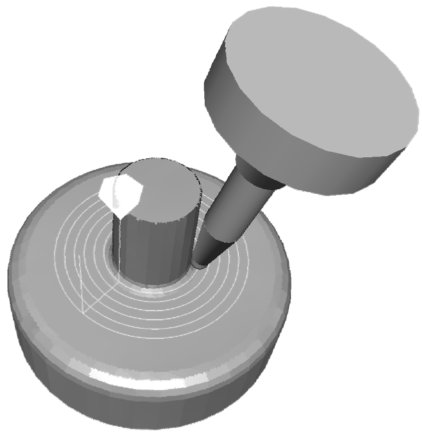

Figure 4 : Column test workpiece with the tool positioned at the base of the column.

This workpiece needs to be milled with a long shank in a three-axis strategy to enable the processing of the small diameter (1 mm) fillet at the base of the column. The milling of these paths is not possible with a three-axis strategy and the chosen shank-holder combination.

The collision is here most likely at the inner paths where the edge of the column is approached. The initial search space for the positioning of the tool points ranges from - 40 mm to + 40 mm along all axes of the global object space. Only 10 initial individuals, which are spread stochastically over this 80 x 80 x 80 mm search space, are used. To obtain an overview of the behavior of the algorithm, this test run was repeated 20 times. Fig. 5 shows a plot of the average fitnesses.

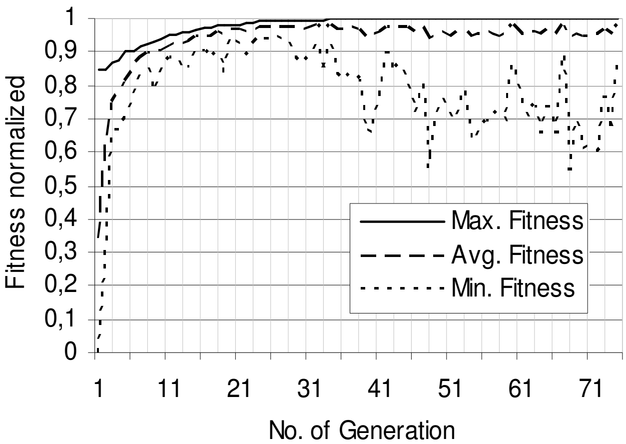

Figure 5: Fitness progression of the column test workpiece after 20 tests.

The fitness diagrammed above denotes the first criterion, which is evaluated. This criterion is the simulated amount of collisions between the workpiece and the tool. The values of the other criteria are not shown to focus on the general usability of the algorithm. As it can bee seen, a collision free tilt point is found not later than the 25th generation in most of the test runs. The development of the curve for the average minimal fitness is more interesting. For 25 generations the average minimal fitness value increases steadily, after the 25th generation the minimal fitness value starts oscillating due to the influence of the other criteria, which come to control when the first criteria, the absence of collisions, is fulfilled. The engagement criterion pulls the tilt point to a side of the collision free area to optimize the engagement. This leads to individuals, which, after a minimal mutation, may not be free of collisions. A closer look on this behavior is given on fig. 6.

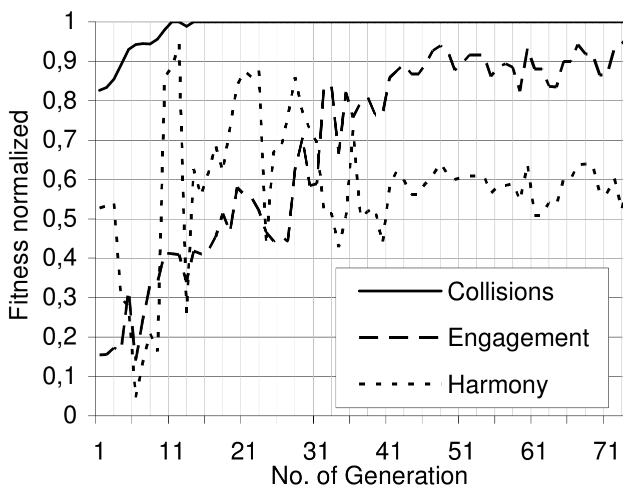

Figure 6: Fitness progression of three criteria shown on the column test workpiece in a single run.

This figure shows the fitness progression for the current best individual in a single run for three different criteria. As it can be seen, the collision criterion has reached a maximum at about generation 25 as in figure 5. At this point, the engagement value gains more importance, which leads to the described behavior. The third criterion, the harmony of the machine movement, is not increasing over the shown run. Due to the parameterization, an optimization of the engagement criterion is preferred.

6.3 Drop Forging Workpiece

The second workpiece is a part for the drop forging of a chain element. A conversion of the given three-axis tool path would allow using a shorter and sturdier shank for the process. To increase the speed of the fitness determination only an excerpt of the finishing step at the bottom of the cavity was used for the fitness determination (see fig. 5). This is a valid way if the selected paths are representative for the entire tool path.

Figure 7: The drop forging workpiece.

Both test workpieces, the column and the drop forging part, have different constraints. The collision spaces of the workpieces differ and therefore, the collision avoidance has varying demands on the evolution progression. The simulation can only deliver a value of how many collisions have occurred. Consequently, the fitness landscapes are different, and a parameterization that is suitable for both cases should be suitable for other more practical workpieces as well.

Cavities are difficult to mill if they are of a certain depth and have small diameters because this requires tools with long thin tool shanks. The use of long shanks causes undesirable effects on the surface quality and increases the risk of tool breakage. To overcome these effects the process speeds need to be reduced. This workpiece needs to be milled with a long shank as well to enable the processing of the small diameter (1 mm) fillet at the base of the column. 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. 8 shows the average progression of the collision criterion after 20 test runs for the drop forging part. As it can be seen, a collision free configuration is found much faster than in section 6.2. This is a typical behavior for solutions of cavity problems. One reason is that the calculation of collision free paths is much easier if there are no parts of the workpiece between the tool paths.

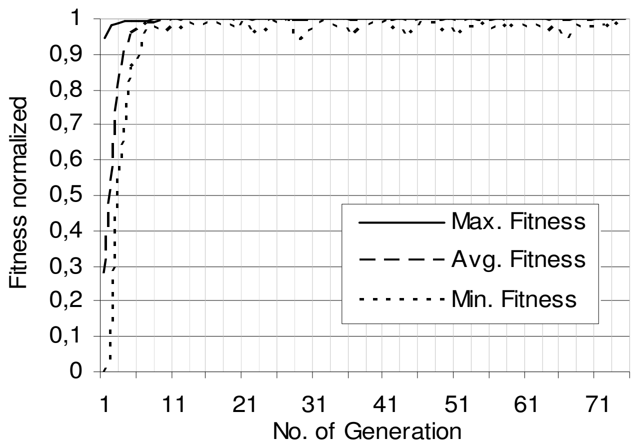

Figure 8: Fitness progression of the drop forging workpiece after 20 tests.

The optimization of the other values of the criteria vector is not successful with this workpiece. Fig. 9 shows that both criteria have much better values at the beginning of the evolution. During the collision search the value for engagement decreases, but after generation five, the algorithm tries to increase the value and keep the paths free of collisions. This may result from the smaller area, in which the algorithm can search for collision free paths with good engagement conditions. The best engagement conditions lead obviously to collisions. Again, the harmony criterion is not very well optimized due to the preceding engagement optimization.

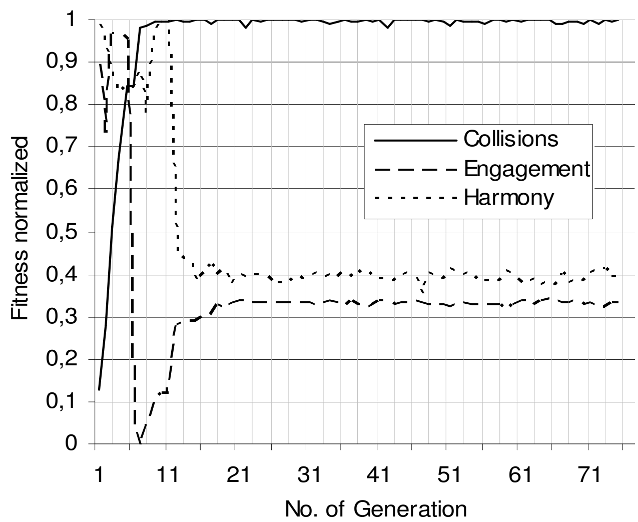

Figure 9: Fitness progression of the drop forging in a single run.

7 PRACTICAL WORKPIECE CAR CRANKSHAFT

The pre-parameterized algorithm is now used for a real practical workpiece a mould for a car crankshaft. The chosen workpiece and the three-axis milling paths are not specially designed for five-axis milling or for this evolutionary path design. The original paths were designed for a three-axis milling process. The collision situation is very difficult for this workpiece due to its deep cavities. To exclude 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 lead to a collision if they are used with this tool, though.

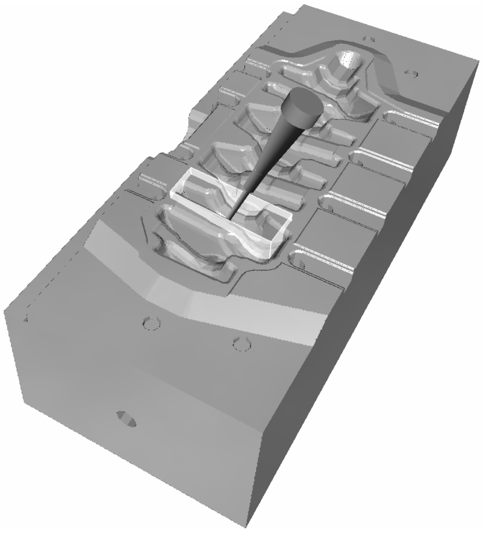

Figure 10: Car crankshaft mould with one area marked at the cavity selected for the tool path conversion.

For test processing a finishing step was chosen. As in the preceding workpiece, an excerpt of the tool path is used to decrease the time for fitness calculation. As in the test runs, 10 individuals per population are used. The initial search space for these tests is increased to 200mm x 200mm x 200mm according to the larger dimensions of the workpiece (160mm x 160mm x 40mm). All other parameters were kept constant.

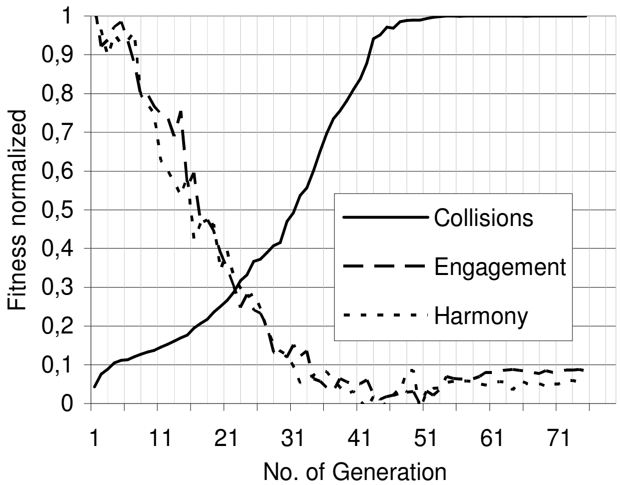

Figure 11 : Average fitness's from 10 evolution runs with 50 individuals each generation

Fig. 11 shows a fitness plot for the test run over 75 generations. Because of the deep cavity and the difficult collision situation, a collision free configuration is found later than in the preceding test, around generation 50. As a consequence of this, the optimization of the two other criteria has only few generations left to develop a solution. However, the general behavior is comparable to the first tests. Good engagement conditions are easy to find if there may be collisions. With the decreasing of collisions, the space for optimal engagement conditions decreases. After the first optimization only a few or small changes in the engagement conditions are possible.

The optimization works well on all tested types of workpiece - tool path combinations. The used fitness calculation is suitable for the specific order of criteria. As it can be seen in all three optimization tests, a high optimization of the movement harmony after the engagement conditions is very difficult. This may be a backdrop from the lexicographic optimization. Further researches will include hybrid techniques of lexicographic and pareto optimization.

8 CONCLUSIONS AND OUTLOOK

This paper presents a new system for the optimization of tool paths and engagement conditions for multi-axis milling as well as the generation of efficient multi-axis tool paths. First studies show that the proposed multiobjective 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 to different problem cases without further user knowledge about the evolutionary parameters. Although the introduced software system is an academic prototype, this technique may be used as a basis for a commercial solution for multi-axis path generation in CAM applications.

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

9 ACKNOWLEDGEMENTS

This research was supported by the Deutsche Forschungsgemeinschaft as part of the Collaborative Research Center “Computational Intelligence” (SFB 531).

10 REFERENCES

[1] Damm, P.; Friedhoff, J.; Stautner, M.: Maschinenoptimiertes 5-Achsen-Simultanfräsen nimmt einzug in die Welt des Werkzeugbaus. Schmiede Journal, March 2004, pp. 22-26

[2] 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 Modelling and Applications, Elber, G.; Shapiro, V. (eds.), 2003, pp. 258-261

[3] 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

[4] Hook, T. van: Real Time shaded NC Milling Display. Computer Graphics 20 (1986) 4, pp. 15-20

[5] Weinert, K.; Müller, H.; Friedhoff, J.: Efficient discrete simulation of 3-axis milling for sculptured surfaces. Production Engineering 3 (1998) 2, pp. 83-88

[6] Weinert, K.; Stautner, M.: An Efficient Discrete Simulation for Five-Axis Milling of Sculptured Surfaces. Production Engineering, 9 (2002) 1, pp. 47-51

[7] Castelino K.; D'Souza, R.; Wright, P. K.: Tool-path Optimization for Minimizing Airtime during Machining, Mechanical Engineering Dept. University of California at Berkeley, http://kingkong.me.berkeley.edu/~kenneth/research/pubs/airtime_minimization_joms.pdf

[8] Alander, J. T.: An Indexed Bibliography of Genetic Algorithms and the Travelling Salesman Problem. Internal Report 94-1-TSP, University of Vaasa, Department of Information Technology and Production Economics, Finland, 1994

[9] Ehrgott, M.; Multicriteria Optimization. In: Lecture Notes on Economics and Mathematical Systems, No: 491, Springer-Verlag Berlin Heidelberg, 2000

[10] 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., 1998