



## UTILIZATION OF GENETIC ALGORITHM BY THE TOOL PATH PROGRAMMING

CUBONOVA, N[adezda]

**Abstract:** *Tool path verification and optimization are two of the best ways you can dramatically improve the manufacturing operation and save money with relatively little work. Genetic algorithms can be used for improvement of these operations and considerably reduce length of toolpaths which leads to reducing of machine times. This article deals with the use of genetic algorithms as an optimization method and their use for optimization of complex tasks. There are described concrete application possibilities of genetic algorithms for solving of technological process optimization as well as description of Toolpath Optimizer functions.*

**Key words:** *optimization, genetic, CAM, system*

### 1. INTRODUCTION

Mathematic analysis disposes with a large variety of mathematic models for solving large variety of optimizing problems. But many of real world tasks cannot be solved by those techniques, or their solutions are not as good, as solutions of some special nonlinear optimization techniques. Therefore, some special optimization techniques were designed, which can find a solution very close to the optimal one, but search only in a very little fragment of solution space. One of these optimization techniques is Evolution Programming (EP). EP is a name of large variety optimization techniques based on evolution principles. Genetic algorithms (or GA for short) are one of these evolution techniques, which mostly imitate principles of natural evolution process. GA algorithms show themselves as a very strong optimization technique able to solve very complicated task with large search space in a very short time with very good results. There is a lot of application, where is not used any optimization technique or there is one, which results can be improved. GA can be used for improvement of these tasks and considerably reduce length of toolpaths which leads to reducing of machine times. GA can be used for a very special optimization task named TSP (Traveling Salesman Problem). Definition of this task is very simple: the salesman must visit every place in his trading area once and then return to the starting city. If he knows the cost of the travel among individual cities, how should he plan the tour for the minimal costs? The search space is then permutation of  $n$  cities, every permutation is then one possible solution and optimal solution is the permutation with minimal traveling costs. Size of the search space is then  $n!$  (Zbigniew, 1999).

### 2. INTRODUCTION TO GA

Concisely stated, a genetic algorithm is a programming technique that mimics biological evolution as a problem-solving strategy. Given a specific problem to solve, the input to the GA is a set of potential solutions to that problem, encoded in some fashion, and a metric called a *fitness function* that allows each candidate to be quantitatively evaluated. These candidates may be solutions already known to work, with the aim of the GA being to improve them, but more often they are generated at random. The GA then evaluates each candidate

according to the *fitness function*. In a pool of randomly generated candidates, of course, most will not work at all, and these will be deleted. However, purely by chance, a few may hold promise - they may show activity, even if only weak and imperfect activity, toward solving the problem. These promising candidates are kept and allowed to reproduce. Multiple copies are made of them, but the copies are not perfect; random changes are introduced during the copying process. These digital offspring then go on to the next generation, forming a new pool of candidate solutions, and are subjected to a second round of fitness evaluation. Those candidate solutions which were worsened, or made no better, by the changes to their code are again deleted; but again, purely by chance, the random variations introduced into the population may have improved some individuals, making them into better, more complete or more efficient solutions to the problem at hand. Again these winning individuals are selected and copied over into the next generation with random changes, and the process repeats. The expectation is that the average fitness of the population will increase each round, and so by repeating this process for hundreds or thousands of rounds, very good solutions to the problem can be discovered. Before a genetic algorithm can be put to work on any problem, a method is needed to encode potential solutions to that problem in a form that a computer can process. These are the most common methods for solution encoding: *binary string* (0's and 1's), *string of values* (integer, real, letter...), *string encoded as permutation* (every value can appear in the string only once), *branching data structures* (trees), *matrix encoding*. After that, some selecting mechanism has to be applied for keeping promising individuals to reproduce. There are many different techniques which a genetic algorithm can use to select the individuals to be copied over into the next generation, but listed below are some of the most common methods. Some of these methods are mutually exclusive, but others can be and often are used in combination: *Elitist selection*, *Fitness-proportionate selection*, *Roulette-wheel selection*, *Scaling selection*, *Tournament selection*, *Rank selection*, *Generational selection*, *Steady-state selection*, *Hierarchical selection*. Once selection has chosen fit individuals, they must be randomly altered in hopes of improving their fitness for the next generation. There are two basic strategies to accomplish this: **mutation** (where changes one gene to another) and **crossover**, entails choosing two individuals to swap segments of their code (Michalco & Čuboňová 2009).

### 3. APPLICATION POSSIBILITIES OF GA

TSP problem can be applied for some of the technological processes such as drilling like operations or local milling operations. Most of the common CAM and CAD/CAM systems are not using any optimization technique, or are using some kind of linear mathematic based technique for solving this task due to the computing time. GA due to their efficiency can solve mentioned problems in few minutes instead of a long time (hours, days) with very good result. This is mostly possible

because of their basic functions and preferences like *parallelism*, *schema theorem* and *crossing*.

At our Department of Automation and Production Systems, research activities were aimed at resolving of GA application in practice. Using GA was designed and developed implementation of optimization model and GA was used as an optimization tool for both symmetric and asymmetric TSP problem. This algorithm is a main core of a software application for optimization of toolpaths created by CAD/CAM system ProEngineer (Pro/E) Wildfire 3.0. Application "Toolpath Optimizer" was written in Borland Delphi language and works as an independent optimization tool. On the Fig. 1.a) is displayed simulation of unoptimized toolpath for drilling process generated with system Pro/E. System is using some kind of linear optimization of rapid moves only for drilling like operations such as drilling threading etc. This moves can be shorten which leads to reducing of machine times. On the Fig.1.b) is displayed the toolpath of drilling process with the main point for optimization.

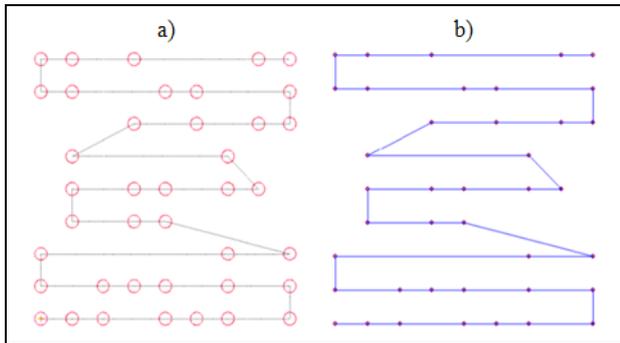


Fig. 1. a) Unoptimized toolpath for drilling process; b) Toolpath for optimization in Toolpath Optimizer

Graphic flow of the optimizing process is displayed at Fig.2. On the right side are graphical informations about flow of evolution process. On the left side is displayed original length (*Length init*) of toolpath generated from the CL data file and new graphic flow (*Length new*) generated in the optimizing process by GA (Michalco & Čuboňová 2009).

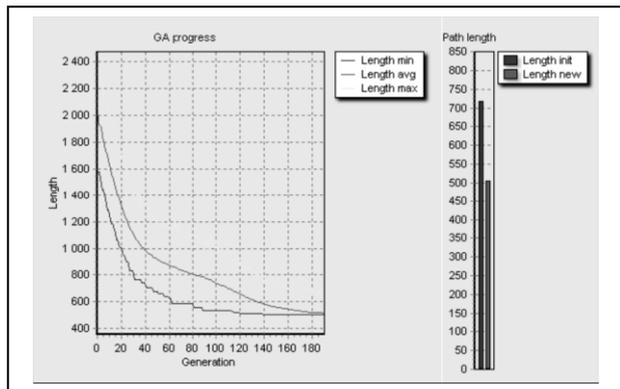


Fig. 2. Graphical flow of optimizing process

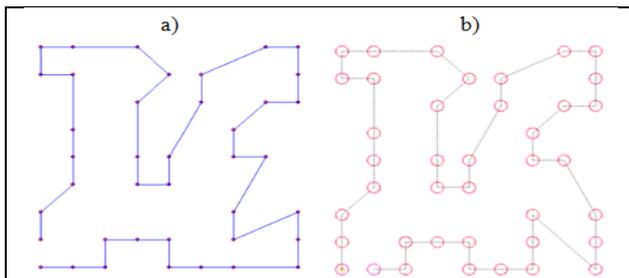


Fig. 3. a) Optimized toolpath for drilling process; b) Optimized Toolpath loaded to Pro/E

On the Fig.3.a) is displayed the resulting toolpath with the optimized point by using application *Toolpath Optimizer*.

After verification of optimizing process are made changes in CL data files Fig.4. On the figure 3.b) is displayed toolpath generated from the optimized CL data file.

```

$$-> FEATNO / 585
MACHIN / UNCX01, 1
$$-> CUTCOM_GEOMETRY_TYPE / OUTPUT_ON_CENTER
UNITS / MM
LOADTL / 3
$$-> CUTTER / 2.000000
$$-> CSYS / 1.0000000000, 0.0000000000, 0.0000000000, 0.0000000000, $
0.0000000000, 1.0000000000, 0.0000000000, 0.0000000000, $
0.0000000000, 0.0000000000, 1.0000000000, 0.0000000000
SPINDL / RPM, 1.000000, CLW
RAPID
GOTO / 95.0000000000, -45.0000000000, 1.0000000000
CYCLE / DRILL, DEPTH, 3.000000, MMPM, 1.000000, CLEAR,
1.000000
GOTO / 95.0000000000, -45.0000000000, 0.0000000000
GOTO / 85.0000000000, -40.0000000000, 0.0000000000
GOTO / 80.0000000000, -35.0000000000, 0.0000000000
GOTO / 80.0000000000, -30.0000000000, 0.0000000000 . .
CYCLE / OFF
SPINDL / OFF

```

Fig. 4. Optimized CL data file

## 4. CONCLUSION

Production process optimization is a powerful tool for massive affection of productivity, part cost and thereby its implementation on the market. Because of that it is important to pay increased attention to this field and notice especially new trends in the field of CA systems, which can aid us to make our production process more effective.

Tool path verification and optimization are two of the best ways you can dramatically improve the manufacturing operation and save money with relatively little work. GA can be used for improvement of these operations and considerably reduce length of toolpaths which leads to reducing of machine times. There are concrete application possibilities of genetic algorithms for solving of technological process optimization as well as description of Toolpath Optimizer functions. Toolpath Optimizer was designed on the base of theoretical and empirical knowledge acquired from genetic algorithm problematic and is designed for optimization of tool paths.

## 5. ACKNOWLEDGEMENTS

The results presented in this article are part of the solution VEGA research project titled *Implementation of genetic algorithms into optimization of technological processes*. In this work, GA based on TSP task was used as an optimization tool for improving the rapid move efficiency and reducing the length of toolpaths. Work deals with common problems of GA's like comparison of several techniques and searching for method, good enough for our purpose.

## 6. REFERENCES

- Zelinka, I.; Oplatková, Z.; Šeda, M.; Ošmera, P.; Včelař, F. (2009). *Evoluční výpočetní techniky – principy a aplikace*, BEN, ISBN 978-80-7300-218-3
- Michalco, M.; Čuboňová, N. (2009). *Implementation of Genetic Algorithms Into Solving of Optimization Task*. In: *Advanced Technologies in Production Engineering*. Monograph Lublin, ISBN 978-83-87833-88-6, s. 122-126.
- Zbigniew, M. (1999). *Genetic algorithms + data structures = evolution programs*, SPRINGER, ISBN 3-540-60679-9, Berlin
- \*\*\* (2006) <http://www.talkorigins.org/faqs/genalg/genalg.html> - The TalkOrigins Archive. Genetic Algorithms and Evolutionary Computation. Accessed on : 2006-12-13