Int J Adv Manuf Technol (2003) 22: 118–124
DOI 10.1007/s00170-002-1450-8

ORIGINAL ARTICLE

M. Kovacic · J. Balic

Evolutionary programming of a CNC cutting machine

Received: 26 March 2002 / Accepted: 17 June 2002 / Published online: 19 March 2003

Abstract The scope of the paper is to construct an autonomous, intelligent CAD/CAM programming system for the cutting device controller (for instance a CNC laser cutting machine tool) based on evolutionary methods. The CNC cutting device should be able to optimise paths autonomously between cutting trajectories, determined by the product's CAD model. An evolutionary GA was used for this purpose.

Keywords Genetic algorith · CNC laser cutting · Path planning

Related work on automatic programming of CNC machines

Conventionally CAM systems for programming of CNC machine tools are well known and commercially available. In recent years automatic CNC programming systems have been applied more and more in tool shops. There are several systems reported in the literature.

The literature [1] describes a numerical controller which contains a programming device to construct a learning-processing program which is combined with the entered program in order to produce the resulting CNC program of the workpiece. This system uses a new method for learning a processing program and enabling the machine operator to select the learning mode and change the CNC program.

The literature [2] describes a tool path data generation apparatus which can speedily and securely generate the tool path data on the basis of CAD data. The tool path data generation includes a feature data extractor to extract features in relation to a CAD model of a workpiece, the tool-cutting data for selecting a cutting mode, the cutting method to set an optimal cutting method and a tool path data generator. An operator does not have to input any data for generating tool path data. Tool path data can therefore be generated automatically and speedily.

The dialog-oriented programming system described in [3] is used for program generation of a CNC machine. All the actions are initiated by the dialog between operator and CNC unit, which has the ability to process program input and download it to the CNC unit.

The concept of the distributed network manufacturing mode is outlined in [4]. This research is concentrated on enhancing the intelligence of conventional NC machine tools and their ability to communicate with the outside world and coordinate the work. The experimental results of the distributed network manufacturing prototype system shows that the system is intelligent and it enhances the ability of a conventional NC milling machine to improve its efficiency and quality and protect the cutting tool.

A system and method for creating varying characteristic products from an automated production line is presented in [5]. The present system includes an automated laser-cutting device. The method comprises programming the laser cutter with a parametric computer program to receive the data file. The method may also involve an automatic power source for the laser cutter upon receipt of the data file.

In [6] an NC feature unit is proposed in order to generate tool path data in real time and is implemented in 2.5 D profile/pocket and 3D surface milling operations.

Aspects all systems have in common are that they are not intelligent and it is not possible to generate a CNC program for unknown parts workpieces. The learning ability of the systems is not presented.

A concept of biological manufacturing systems (BMS) that is based on biologically inspired ideas of self-organization, evolution and learning is discussed in [7].

M. Kovacic · J. Balic (✉)
Faculty of Mechanical Engineering,
Laboratory for Intelligent Manufacturing Systems,
University of Maribor, Smetanova ulica 17,
2000 Maribor, Slovenia
E-mail: joze.balic@uni-mb.si

Autonomous mobility is one of the essential requirements for such elements of a BMS as they are the transporters carrying products to the manufacturing cells.

CNC programming system

Conventional CNC programming

Introducing CNC machine tools into a production workshop requires a series of special methods for the preparation of work and the utilisation of the machine tools. The extent of preparation work increases strongly with modem CNC machine tools that meet the specific requirements of the users. Thanks to machining operations on highly capable CNC machine tools it is possible today to manufacture even the most advanced and exact products. Mechanical parts are becoming more and more complicated. This involves increased preparation work and set-up time on the CNC machine tools. The cost of programming CNC machine tools accounts nowadays for 30% of the production costs of a product; therefore, rational programming of CNC machine tools is very important. Automation of the programming process plays an important role [8]. In general, programming includes the following activities (Fig. 1):

- basic operation layout
- detailed working plan layout
- programming
- output information.

Essential for NC programming are:

- geometrical data
- tools data
- machine tools data.

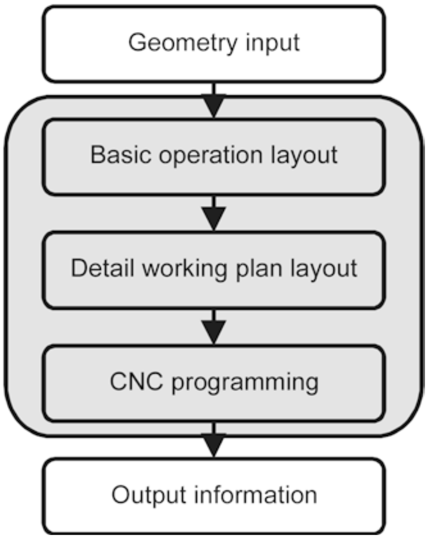

Fig. 1 CNC machine tool programming

Conventional versus evolutionary system

An evolutionary-controlled CNC machine tool is a machine tool controlled by an evolutionary-developed program. An evolutionary-controlled CNC machine is capable of autonomous basic operations and working layout planning for the programming and manufacturing of technologically and geometrically complex products, which result in a decrease of the above-mentioned 30% production costs and the product price proportionally (Fig. 2).

Programming and manufacturing are the main product manufacturing phases. Programming consists of planning and optimisation. Manufacturing planning determines product manufacturing, so planning has to be carefully reconsidered. Simple calculation can prove that statement. The evolutionary-supported programming phase is fully automated so it reduces programming costs by nearly 30%, which means that we can reduce production costs by 10%. If in the next step the manufacturing costs are reduced by only 10% we gain an enviable 20% of production costs.

An evolutionary GA was used in this work. It has been proved to be an effective optimisation tool for multicriterial and multiparametrical problems in a constantly changeable environment with many different, unexpected and sometimes even contradictive conditions, which are difficult to foresee and even more difficult to control by conventional methods.

Short introduction to evolutionary programming

The evolutionary optimisation processes differs from conventional optimisation methods that use probabilistic principles (stochastic operations); therefore, none of the above limitations applies to them. Figure 3 shows the general process of evolutionary optimisation methods. Solving the optimisation problem (usually) starts with a random creation of solutions (points) [9].

Since we stick to the biological metaphor in solving, the solutions are called organisms or chromosomes. Each organism generated randomly represents a more or less accurate solution of the optimisation problem. The organisms are then evaluated. A greater probability of cooperating in the selection and variation operations is assigned to those organisms that solve the problem better,

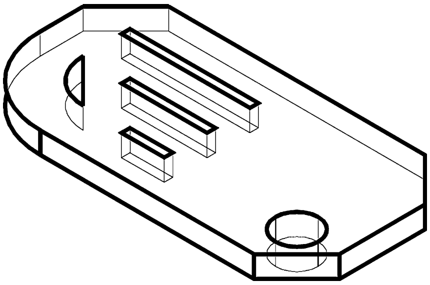

Fig. 2 From workpiece to product

```
Evolutionary algorithm
    $t$: =0
    create initial random population $P(t)$
    evaluate $P(t)$
        repeat
            vary $P(t) = P(t+1)$
            evaluate $P(t+1)$
            $t$: $= t + 1$
        until not terminate
```

Fig. 3 Evolutionary algorithm in pseudo code

i.e. they are better adapted to the environment. A selective operation assures the survival of the fitter individuals in the population and they advance unchanged into the next iteration, also called the next generation. The variable operation affects one or more parental organisms, which create their offspring. After completion of selection and variation a new generation is obtained and evaluated. The process is repeated until the termination criterion of the process is fulfilled. This can be a prescribed number of generations or a sufficient quality of solutions.

Genetic algorithms

Since their introduction by J. Holland [10] the use of genetic algorithms has spread to almost all areas of research [11, 13].

A population consists of organisms. Organisms are points in the space of the solutions. An organism has coordinates that are called genes (Fig. 4). It is a characteristic of a conventional GA that organisms represent coded values of variables (parameters) of the mathematical model (cost function). The most widespread method is to code variables into fixed-length binary strings. The length of organisms is determined with the respect to the size of the interval with which the solution is searched for and with respect to the desired resolution of the variables. The binary representation of the organism is called genotype and the actual value of the organism is called a phenotype (Fig. 4). The evaluation of organisms is the driving force in the evolutionary process. The quality of the individual organism is determined on the basis of its ability to solve the problem. A higher probability of cooperating in basic operations of the conventional GA (e.g. reproduction, crossover, and mutation) is prescribed to organisms (solutions) of higher quality. Thus, the fitter organisms transfer their genetic material more frequently into the next generation, whereas less fit organisms slowly die away from the population.

Let us look at the basic genetic operations. The reproduction operation (Fig. 5) gives a higher probability of selection to organisms of higher quality. They are copied unchanged into the next generation.

The crossover operation (Fig. 6) ensures the exchange of genetic material between organisms. From two parental organisms, two offspring organisms result at the level of a genotype.

Mutation (Fig. 7) at the level of the genotype randomly introduces new genetic material into organisms.

The evolutionary development of organisms usually leads to better and better solutions. In the literature it is possible to find many variants of the conventional GA that are adapted to the specific characteristics of the optimisation problem dealt with. The variants differ particularly in the representation of the organisms and the use of additional or modified genetic operations [9, 12, 13]. Evolution is terminated when a termination criterion is fulfilled. This can be a prescribed number of generations or a sufficient quality of the solution. Since evolution is a nondeterministic process, it does not end with a successful solution in each run. In order to obtain a successful solution, the problem must be processed in several independent runs. The number of runs required for the satisfactory solution depends on the difficulty of the problem.

Evolutionary-programmed laser cutting

Problem statement

The evolutionary-supported laser cutting programming problem is to obtain an optimal cutting path between determined cutting trajectories.

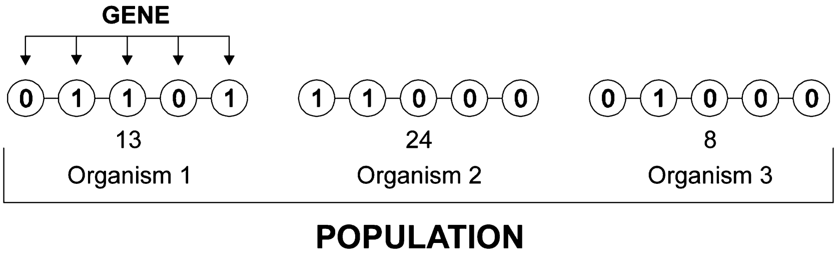

Fig. 4 Genetic algorithm population

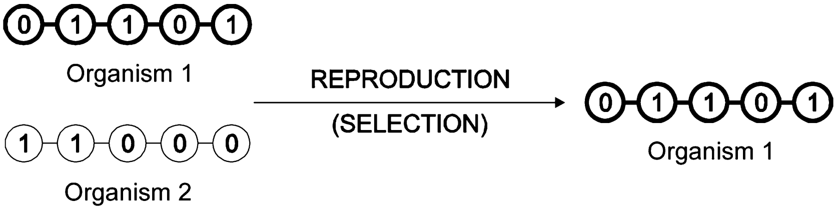

Fig. 5 Genetic algorithm reproduction

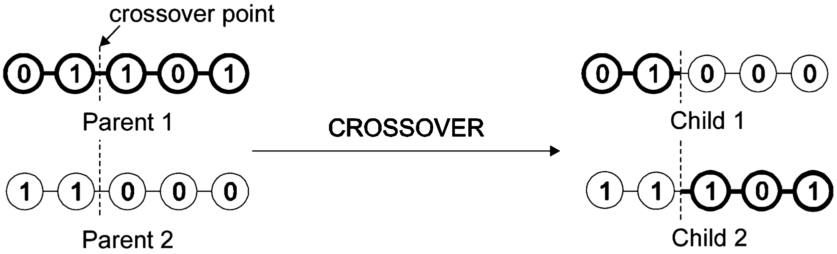

Fig. 6 Genetic algorithm crossover

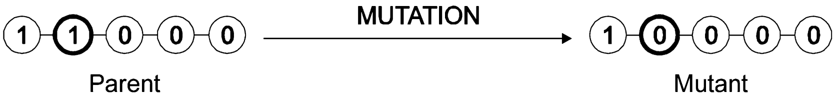

Fig. 7 Genetic algorithm mutation

The proposed evolutionary-programmed model imitates two-dimensional laser cutting.

Figure 8 shows three-dimensional laser cutting using TopLas 3D software (Erlanger Lasertechnik, Erlangen, Germany, http://www.erlas.de). A laser cutting programming task is collision-free material cutting using the determined cutting trajectories.

Evolutionary laser cutting parameters

A sheet metal workpiece has a constant thickness. The product is cut using the determined cutting trajectories. The evolutionary laser cutting programming task is to produce comprehensible optimal manufacturing workpiece instructions.

Figure 9 shows tool start, end point and tool path, which consist of four cutting and five non-cutting trajectories. The tool cutting trajectories are marked bold.

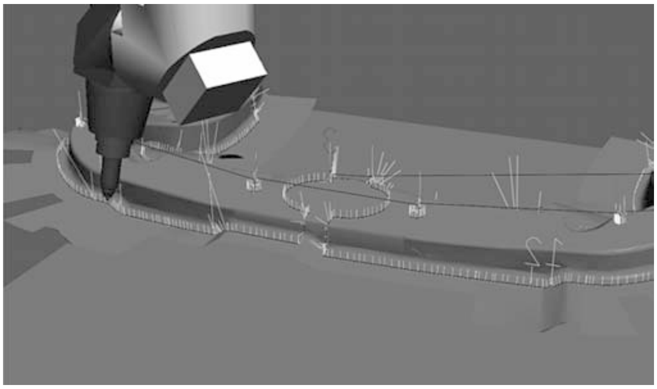

Fig. 8 3D laser cutting

For one open cutting trajectory a machine tool has two manufacturing strategies. The tool can cut from the end of the first cutting trajectory or the end of the second trajectory. For 2 open cutting trajectories the cutting machine can choose between 8 cutting strategies, for 3 open cutting trajectories there are 48 cutting strategies and for 10 open cutting trajectories there are an immense 3.71 billion different cutting strategies.

The aim of the evolutionary-supported controller is to find an optimal or even the best solution to the problem of an optimal collision-free cutting strategy.

Figure 10 shows the cutting trajectories transformed to elements. An element gene consists of a trajectory start and end point and a trajectory length. A chromosome is a cutting strategy consisting of genes and is made up of cutting trajectory sequences. Chromosomes are organisms that represent the population (Fig. 11).

Crossover, reproduction, mutation and permutation are simple genetic operations which were used at planning and optimisation of the evolutionary-programmed laser cutting.

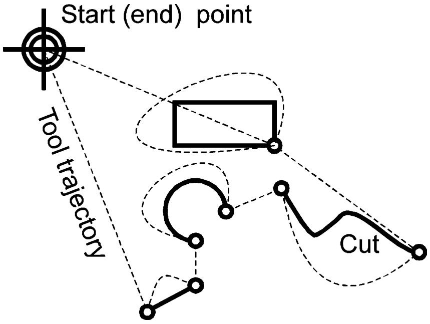

Fig. 9 Tool path from start to end point

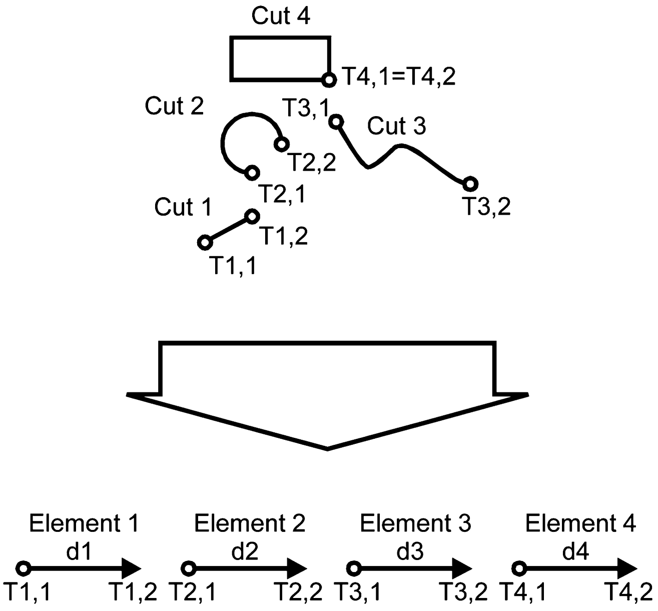

Fig. 10 Cutting trajectories transformation to elements

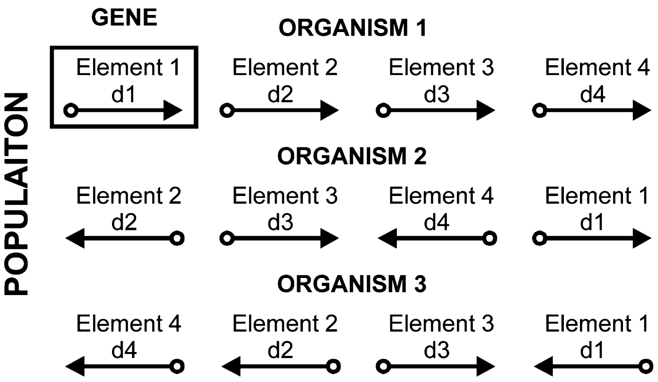

Fig. 11 Genes, organisms, population

The reproduction operation gives a higher probability of selection to organisms of higher quality. They are copied unchanged into the next generation.

The crossover operation (Fig. 12) ensures the exchange of the genetic material between organisms.

Mutation randomly changes genetic material into organisms. Figure 13 shows the swap of trajectory 3, i.e. of element 3 start and end point.

Permutation (Fig. 14) changes the order of cutting. After permutation trajectory 1 (element 1) is cut before trajectory 3 (element 3).

The evolutionary development of organisms usually leads to better and better solutions. Evolution is terminated when a termination criterion is fulfilled. This can be a prescribed number of generations or a sufficient quality of the solution. In order to obtain a successful solution, the problem must be processed in several independent runs called civilizations.

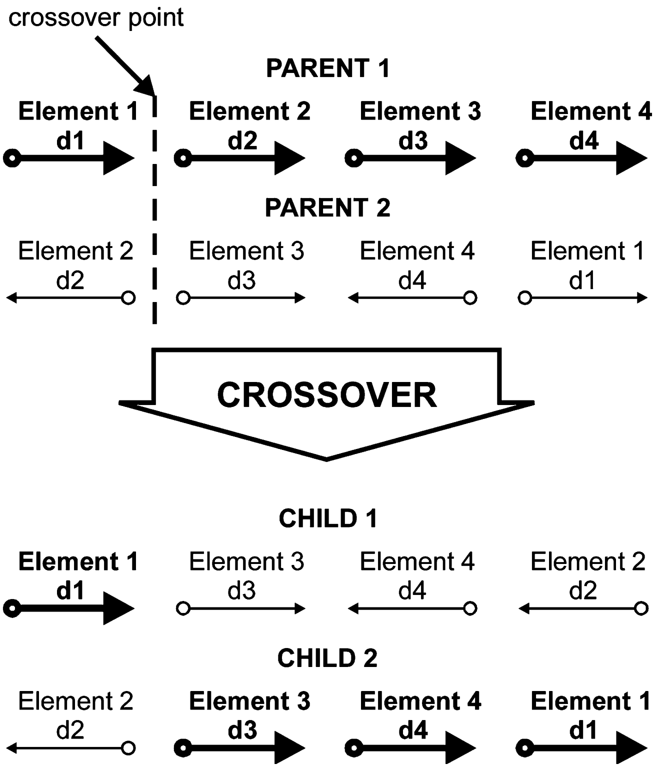

Fig. 12 The crossover operation

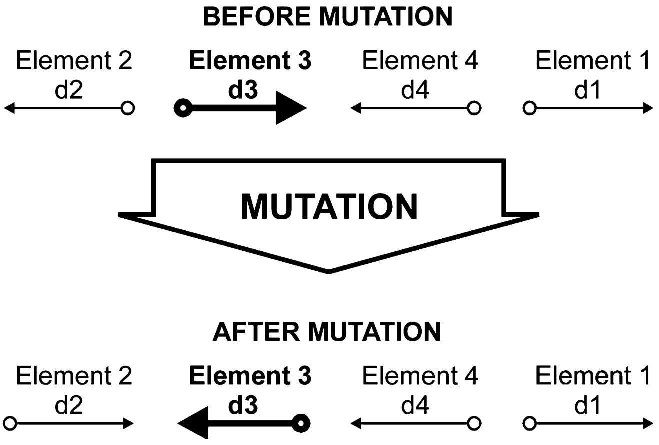

Fig. 13 Mutation

Case study

Definition of sample workpiece

The actions of the evolutionary-supported laser cutting programming are illustrated by the following practical example. The product has to be manufactured in a constant thickness sheet metal workpiece (Figs. 15, 16).

Figure 17 shows the cutting trajectories as results of geometrical data input of the sheet metal workpiece and product.

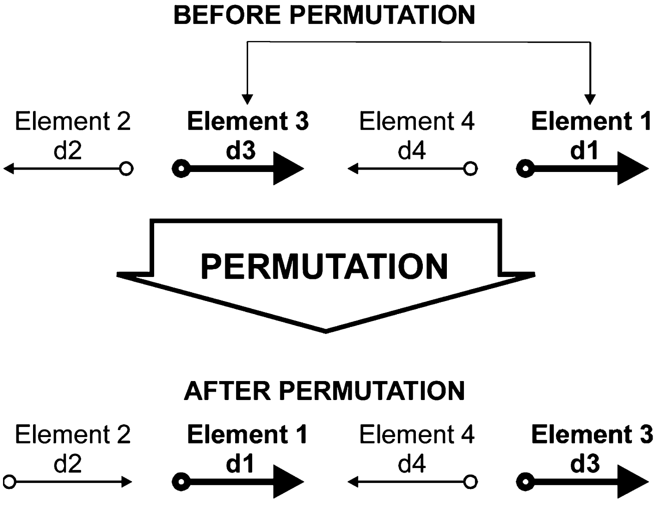

Fig. 14 Permutation

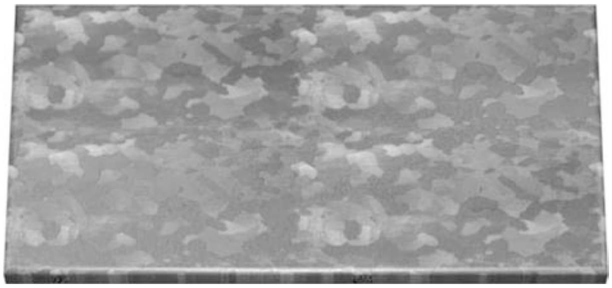

Fig. 15 Sheet metal workpiece

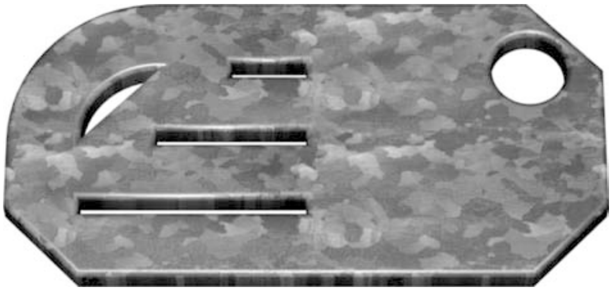

Fig. 16 Product

A population of 500 individuals was used. The reproduction probability was 0.1 and crossover, mutation and permutation probability were 0.3. The number of generations to be run was 100. The average computation time per run was 8 s on an AMD K7, 1 GHz, 384 MB RAM.

Comparison of results

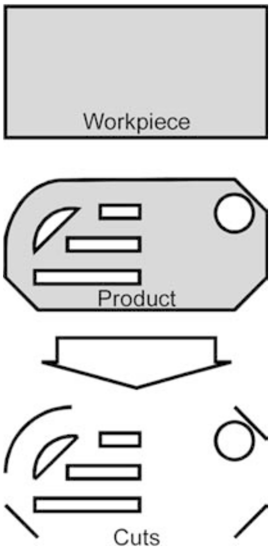

Fig. 17 Metal sheet workpiece, product and cutting trajectories

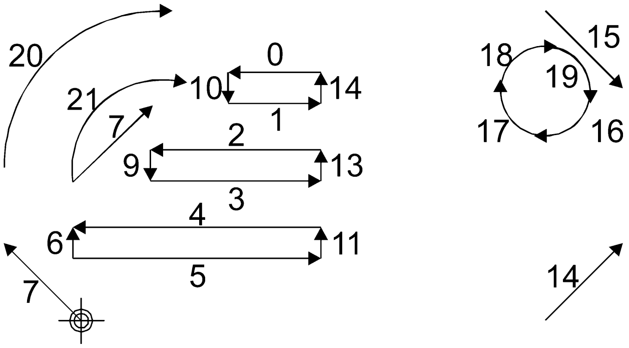

Fig. 18 Random organism treatment

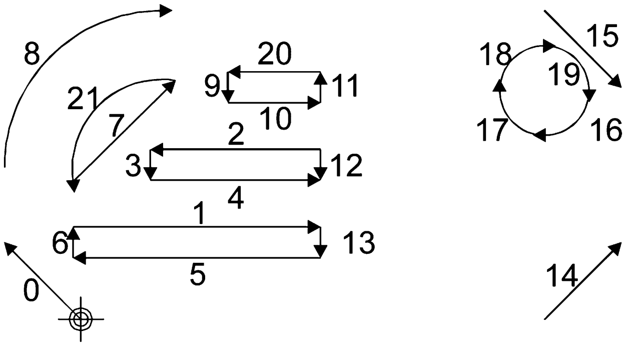

Fig. 19 Optimal cutting strategy the best organism in civilization

The comparison of the random cutting strategy (Fig. 18) and optimal cutting strategy (Fig. 19) gives a difference of 31.81%. It is important that the difference between the two strategies in Figs. 18 and 19 is not obvious. The evolutionary-programmed machine tool has to establish autonomously a hidden optimal solution.

Figure 19 shows an optimal solution, i.e. an optimal cutting strategy. The arrows show a cutting direction; the numbers from 0 to 21 represent the cutting sequence. A random cutting strategy length is 1515 units. The optimal cutting strategy length is only 1033, making the difference 31.81%.

Conclusion

The evolutionary method GA, which has been proved to be an effective optimisation tool for multicriterial and multiparametrical problems, was successfully implemented for autonomous laser cutting programming. CNC programming represents 30% of the production costs. The programming phase, i.e. the manufacturing planning and optimising phase, was successfully fully automated.

The proposed concept is a contribution to the intelligent manufacturing systems of the future. The objectives are autonomy, distribution, hierarchical organization, decentralised control, self-organization, intelligent decision making in case of unexpected events and a consideration of exact ecological criteria. Intelligent systems of the future will be a fusion of non-deterministic and deterministic approaches.

The conventional production concepts for flexible manufacturing systems (FMS) are not sufficient and flexible enough to meet manufacturing criteria of the future. FMSs are insufficient from the technical and ecological points of view.

The proposed concept of the production system uses biological evolution principles of genetic combinations and natural selection of the fittest. The FMS is therefore capable of autonomous and universal problem solving and adaptation to unexpected events. Future research is developing with reference to the practical application of the proposed concept. Preliminary results are optimistic.

References

1. Kato K, Momochi T (1990) Numerical controller for machining tool with learning function. Patent number DE4011591
2. Kadono M (2001) Tool path data generation apparatus for NC machine tool and numerical controller provided with it. Patent number US2001000805
3. Waldenmaier P, Wirth H, Monz J (1996) Dialogue orientated programming system for a CNC machine tool. Patent number EP0726509
4. Cheng T, Zhang J, Hu C, Wu B, Yang S (2001) Intelligent machine tools in a distributed network manufacturing mode environment. Int J Adv Manuf Technol 17(3):221–232
5. Torvinen JM (2001) Parametric programming of laser cutting system. Patent number: WO0150404
6. Zhang L, Deng J, Chan SCF (2000) A next generation NC machining system based on NC feature unit and real-time tool-path generation. Int J Adv Manuf Technol 16:889–901
7. Ueda K (1999) Learning approaches to autonomous robots in biological manufacturing system. In: Proceedings of the tenth international DAAAM symposium: Intelligent manufacturing and automation, Vienna, Austria, 21–23 October 1999, pp 559–560
8. Balic J (1999) Contribution to integrated manufacturing (Manufacturing technology series). DAAAM, Vienna, Austria, p 223
9. Brezočnik M, Balič J, Kampuš Z (2001) Modeling of forming efficiency using genetic programming. J Mater Process Technol 109(1/2):20–29
10. Holland J (1975) Adaptation in natural and artificial systems. University of Michigan
11. Goldberg DE (1989) Genetic algorithms in search, optimization, and machine learning. Addison-Wesley, Reading, MA
12. Koza JR (1994) Introduction to genetic programming. In: Kinnear EK (ed) Advances in genetic programming. MIT Press, Cambridge, MA, pp 21–42
13. Mitsuo G (1997) Genetic algorithms and engineering design. Wiley, Canada