



UDC 65.011.42

# GENETIC ALGORITHMS APPLICATION TO DECIDE THE ISSUE OF SINGLE-DIMENSIONAL CUTTING OPTIMIZATION

Belova N.A.<sup>1</sup>, Lisiuk G.P.<sup>1</sup>

<sup>1</sup> Odessa National Polytechnic University, Odessa

<sup>1</sup> E-mail: [maxvell2k13@gmail.com](mailto:maxvell2k13@gmail.com)

Copyright © 2014 by author and the journal “Automation technological and business - processes”.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



DOI: [10.15673/atbp.v8i2.161](https://doi.org/10.15673/atbp.v8i2.161)

## Annotation

**This paper considers an example of solving the problem of single-dimensional cutting optimization. It is shown that elements of the theory of genetic algorithms can be used successfully for its solution. A distinctive feature of this task is the next: a set of parts and profiles, which the parts are cut from, does not change in the search for solutions but their sequence order changes. Results of the solution were used to write the ‘dll’-library easily integrable into various application software.**

## Keywords

**Optimization, single-dimensional cutting, genetic algorithm**

The modern market offers a variety of specialized programs to solve the problem of single-dimensional and two-dimensional cutting in a variety of industries. There are both paid and free programs, with minimum functionality, as well as with the possibility of setting up the parameters used. Among the significant disadvantages of such programs is the fact that they are absolutely independent and completed products. In practice, the process of cutting optimization is only one of the stages of production. The results of different design systems operation are the input data for it. Output data may also be used in other applications, e.g. for printing cutting charts.

Representatives of a furniture manufacture company addressed the authors of the article with request to develop a universal library for the optimization of one-dimensional cutting of MDF profile. The existing programs were unacceptable because of the impossibility of their integration into the internal production process. The customer’s representatives also noted that the search for the solution of algorithms of the complete enumeration for projects requiring more than 50 ÷ 70 parts is performed impossibly long time (tens of seconds, minutes). The use of traditional algorithms, in some cases, leads to the obvious non-optimal solutions.

Fig. 1 is a diagram showing a required sequence of data processing during the cutting profile process. The used at the enterprise specialized design software allowed the specification of parts of the current project to be presented in the form of a plain text file with the *.txt* extension.

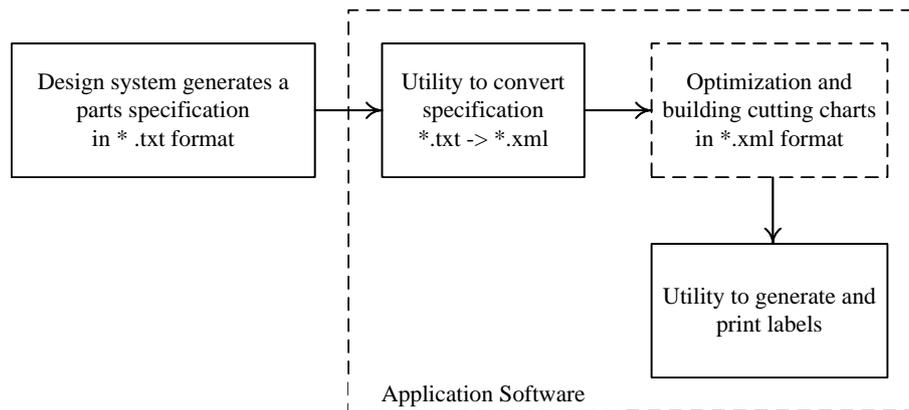


Fig. 1. The information processing diagram in the profile cutting process.

Company's specialists have partially developed an application software that allows automating the further data processing. Thus, the utility for the specification conversion has been developed. It made possible to separate and count all parts manufactured from a profile, as well as to record the existence of available whole and already cut profiles. The utility's output was in the *xml*-file. Information in this file was the raw data for the functions of the optimization library and for cutting mapping, the development of which has been assigned to the authors.

Result of finding the profile's cutting charts had to be presented as another *xml*-file that subsequently allowed, with the help of a separate utility, to generate and print labels indicating the identifiers of parts and their sizes on a special self-adhesive paper.

This process management allowed simplifying the procedure of identification and recording of parts in different shops and thus accelerating the assembly process. When dealing with the profile cutting, a worker immediately attaches the label to each part indicating full information about the parameters of the part. Thus, there was no more need to re-measure each part and to verify against the assembly drawings when passing to the assembly plant.

In general, the cutting optimization issue was formulated as follows. There is a number of whole profiles of different sizes. There are many remnants / scraps of profiles of known length, which is still possible to be bring into production. The following has been specified: the width of cut  $w_c$  and the minimum length of the profile's remainder  $w_{r\ min}$  suitable for further use. The parts' specification is also given. So, it is necessary to calculate the cutting chart indicating which parts should be placed on specific profiles or their residues. At that, it is necessary to observe a number of requirements:

- 1) it is necessary to minimize the consumption of whole profiles;
- 2) it is necessary to minimize the amount of residues with the length less than  $w_{r\ min}$  ;
- 3) it is necessary to minimize the total amount of waste.

Preliminary analysis showed that the problem relates to the category of "containerization" [1]. The problem is NP-hard, and there are many algorithms to solve it. The exact solution can be obtained by the method of complete enumeration, but for obvious reasons the method is not applicable to a large number of parts because of the significant time consumption. The existing algorithms NF, FF, BF, FFD and others allow quickly getting result close to the optimal solution. At the same time, preliminary modeling results showed that, in some cases, such algorithms generate solutions far from optimal on the task at issue.

In further studies, it was proposed to use elements of the theory of genetic algorithms to solve the task set forth [2]. Unfortunately, it was not possible to present solution to the problem within the framework of the classical genetic algorithm definition. It is to be recalled that in the classical definition a solution is given in the form of a chromosome consisting of a gene sequence, and each gene has a value of '0' or '1'. A set of genes can vary, for example, all genes can take the '0' value.

At the same time, various modifications of classical genetic algorithm allow extending the range of tasks. Thus, in [3] there was proposed a method for optimizing of heat-radiating equipment switching plan in a residential cottage. Plan - solution featured a chromosome where each gene could take a value from '0' to '7', which corresponded to one of the possible equipment switching combinations.

Analysis of the results of work [3] enabled proposing a modification of the classical genetic algorithm to solve the cutting optimization task set forth.

Suppose each gene is a structure in the form of:

- Profile ID;
- Profile type (whole / residue);



- Length;
- A list of parts that are planned to cut from this profile;
- Completeness (the overall length of parts based on the cutting width  $w_c$  ).

Suppose a chromosome is the immutable sequence of genes-containers sorted in ascending order. Here is an example. Suppose, 1, 2 and 3 - the remains of the profiles, but 6 and 7 are whole profiles of the appropriate length. Then the chromosome can be described by the following set of genes:

1	1	2	2	3	3	3	6	6	6	6	7	7	7	7
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

At that, the number of whole profiles of each size should be sufficient to solve the task set forth. This requirement is desirable, but optional, as residues and other whole profiles will be used in the course of evaluation. Meeting this requirement potentially expands the available alternate solutions.

Suppose that we are given the following set of components (parts) that are need to be cut:

2	2	2	2	4	4	4
---	---	---	---	---	---	---

Then, in the course of evolution, we will try to locate the parts in a container more accurately. For example, possible options may be as follows:

Containers	1	1	2	2	3	3	3	6	6	6	6	7	7	7	7
Parts			2	2	2	2		4	4	4					

Containers	1	1	2	2	3	3	3	6	6	6	6	7	7	7	7
Parts			2		2			4	4	4					
								2	2						

Containers	1	1	2	2	3	3	3	6	6	6	6	7	7	7	7
Parts			2	2				4	4						
								2	2						
									4						

Note that in the latter scheme the overall parts' length is more of the container size  $4 + 2 + 4 > 6$ . This situation will occur often enough in the course of evolution, so the value of "congestion" should become one of the criteria used in the objective function.

After a series of preliminary studies, the following objective function was proposed to assess the fitness of each chromosome:

$$J = 1000 \frac{L_{over}}{L_{sum}} + 100 \frac{N_u^0}{N_{sum}^0} + 10 K_{load} \quad (1)$$

where:  $L_{over}$  — the overall "congestion" value for containers;

$L_{sum}$  — the overall length of all parts;

$N_u^0$  — number of whole profiles used;



$N_{sum}^0$  — overall amount of whole profiles;

$K_{load}$  — load factor for containers.

$L_{ower}$  value was determined as the sum of the differences of lengths of all the parts  $L_u$  in this container and its actual length

$L_0$  :

$$L_{ower} = \sum L_u - L_0, \text{ ecnu } L_u > L_0 \quad (2)$$

The perfect choice is when there is no congestion, and the first item in (1) becomes 0.

The situation is similar with the whole profiles. The perfect choice is when the task is solved only with the use of scraps. Then the second item in (1) is also converted to 0.

$K_{load}$  value - is an integral value which characterizes how fully the containers are loaded. '1' corresponds to the minimum load, and '0' means a fully loaded container. In other words, the perfect load means absence of residues when cutting profiles.

Further studies have shown that the best result can be obtained if  $K_{load}$  value is of non-linear character and is also dependent on the size of each container congestion  $L_{ower}$  :

$$\left\{ \begin{array}{l} K_{load} = \frac{\sum f}{N_b + N_u^0} \\ f = \begin{cases} \left( \frac{L_0 - L_u}{L_0} \right)^2 \\ 1 - e^{-0.125}, \text{ ecnu } L_{ower} = 0 \\ 1, \text{ ecnu } L_{ower} > 0 \end{cases} \end{array} \right. \quad (3)$$

where:  $N_b$  — the amount of unutilized residues of profiles.

Next, it was necessary to develop a search for solution algorithm. Classical crossing-over and mutation operations used in the genetic algorithm are not applicable in this case, since they lead to introduction of a new set of genes. In our case, a set of genes-containers remains unchanged. Changes touch only the filling of each container with parts. Different approaches were tested and the final version of the algorithm took the following form.

1) We determine the maximum number of the whole profiles of each size required to limit the size of the problem. Rough estimate was used, when all the parts are laid out in sequence one after the other, taking into account the cutting width. When the overall length of the parts is greater than the length of the profile, the next whole profile is added.

2) We form the initial population of chromosomes. One solution is based on FFD algorithm (the first suitable with ordering). Random filling of containers with parts forms another 10 solutions. One more solution-based FFD algorithm using only the particular size is added for each size of a whole profile.

3) Starting the main loop. Calculation is finished if the number of iterations, in which the objective function value of the best chromosome is not changed, makes more than 30.

4) For each chromosome:

5) Choose a random non-empty container  $K_1$ . Select a random item inside and place it in another random container  $K_2$ . Add the derived chromosome to a new population.

6) Check the possibility to improve the load factor by moving the content of the container  $K_1$  into any other, but of smaller size. If positive, add the derived chromosome to a new population.

7) Check the possibility to improve the load factor by moving the content of the container  $K_2$  into any other, but of bigger size. If positive, add the derived chromosome to a new population.

8) Swap contents of two random containers. Add the derived chromosome to a new population.

9) Shift parts from the overloaded containers into the first appropriate ones. Add the derived chromosome to a new population.

10) In case if not all the chromosomes of the original population are handled, then go to p. 4, otherwise to p. 11.



11) Calculate the value of the objective function for all the chromosomes of the new populations and perform selection, choosing the 20 fittest chromosomes.

12) Fix the value of the objective function of the best chromosome and proceed to p. 3.

*Dll*-library was designed to implement all the calculations after debugging the algorithm in C ++ language. Testing was done on a computer with features as follows: Windows 7, Intel Core i5-4590 3.3 GHz, 8 Gb RAM. Run time for the projects with a number of parts of  $70 \div 100$  pcs did not exceed 15 seconds. Run time for parts of  $10 \div 20$  pcs made  $2 \div 3$  seconds. In the majority of cases, the proposed algorithm allowed improving the solution obtained using the FFD algorithm.

The developed library has been successfully implemented at a customer's enterprise and integrated into its own software.

### Conclusions

1. Results of the work performed show that genetic algorithms theory elements can be successfully applied to solve problems beyond the classical formulation of the optimization problem for the genetic algorithm.

2. The proposed optimization algorithm allows steady improving of solutions, formed on the basis of traditional methods (NF, FF, BF, FFD) used to solve tasks of containers packing.

3. The developed library is a tool that is easily integrated into a variety of application software and allows automating certain elements of a single-dimensional cutting of profiles. Using of *xml* format for the input and output data can easily be linked with other software modules written by other authors.

### Referenses

- [1] Sayt instituta matematiki im. S.L. Soboleva [Elektronnyiy resurs]. – Rezhim dostupa: [www.URL: http://www.math.nsc.ru/LBRT/k5/OR-MMF/lec4.pdf](http://www.math.nsc.ru/LBRT/k5/OR-MMF/lec4.pdf);
- [2] Rutkovskaya D. Neyronnyie seti, geneticheskie algoritmyi i nechetkie sistemyi / D. Rutkovskaya, M. Pilinskiy, L. Rutkovskiy; Per. s polsk. I. D. Rudinskogo. — M. : Goryachaya liniya Telekom, 2006. — 452 s.
- [3] Maksimova O.B. Upravlenie sistemoy teplosnabzheniya s izmenyaemoy strukturoy tehniceskikh sredstv / O.B. Maksimova, V.O. Davyidov, S.V. Babich // Problemyi upravleniya i informatiki: mezhdunar. nauch.-tehn. zhurn. 2014. – #3. – S. 50–60.

Received by edition: 5.04.2016. Approved for the press: 4.06.2016



## AUTOMATION NEWS

### *New 3D Printing Plant, Software and Education*

Like most technologies, there is a huge wave of interest surrounding their introduction followed by a slowdown in news as users evaluate the technology and determine its usefulness through initial applications. The rate of developments through this slowdown in news often determines the staying power of the new technology. And though the news around 3D printing never slowed significantly, recent activity suggests a surge in 3D printing use across industry in the near term.

A few of these announcements I'm referring to include Alcoa's opening of its 3D printing metal powder production facility at the Alcoa Technical Center in Pennsylvania. The facility will reportedly produce proprietary titanium, nickel and aluminum powders for 3D printed aerospace parts. Alcoa is focusing on this area because metal powders used for 3D printing durable aerospace parts are only available in limited quantities. In addition, Airbus recently selected Alcoa to supply 3D printed titanium fuselage and engine pylon parts for its commercial aircraft.

Beyond producing powders for 3D printing, Alcoa recently unveiled its Ampliforge process—a technique that combines additive and traditional manufacturing. Using this process, Alcoa designs and 3D prints a nearly complete part then treats it using a traditional manufacturing process, such as forging. The process is said to increase the toughness and strength of 3D printed parts.

Meanwhile, Stratasys Ltd., a supplier of 3D printing and additive manufacturing technologies, has released new software designed to make 3D printing easier. GrabCAD Print is an open architecture design-to-3D print workflow application that resides on the GrabCAD platform. Stratasys says this platform is supported by a community of more than 3 million designers, engineers and students.

Stratasys explains that the 3D printing design process is typically characterized by significant “model fixing” time, which forces businesses to devise manual solutions to construct an acceptable workflow. GrabCAD Print addresses this issue by eliminating the requirements to translate and repair CAD files. Using GrabCAD Print, users can send native CAD files to a Stratasys 3D Printer or service bureau directly from their familiar CAD environments.

<http://www.automationworld.com>