

UDC 65.011.42

**K.K. Smirnova,
V.O. Davydov, PhD**

Odessa National Polytechnic University, 1 Shevchenko Ave., Odessa, Ukraine, 65044, e-mail: www.sunnystudent23@gmail.com

USE OF GENETIC ALGORITHM AS A SOLUTION OF UNIDIMENSIONAL CUTTING RATIONALIZATION PROBLEM

K.K. Смірнова, В.О. Давидов. Застосування генетичного алгоритму для вирішення задачі раціоналізації одновимірного розкрою. Розглянуто приклад вирішення задачі раціоналізації одновимірного розкрою. Показана можливість використання елементів теорії генетичних алгоритмів для її визначення. Специфічність даної задачі в тому, що в процесі пошуку рішення кількість профілів та нарізаних з них деталей залишається незмінним, а змінюється лише їх послідовність. Модифікації класичного генетичного алгоритму дали можливість розробити новий алгоритм пошуку рішення. Результати рішення були використані для написання dll-бібліотеки, яка легко інтегрується в різноманітні прикладні програмні засоби.

Ключові слова: раціоналізація/ефективність, одновимірний розкрій, генетичний алгоритм

K.K. Smirnova, V.O. Davydov. Use of genetic algorithm as a solution of unidimensional cutting rationalization problem. The following research is dedicated to studying an example of a solution of unidimensional cutting rationalization problem. The possibility to use the elements of GA (genetic algorithm) theory at its determination is shown. The specific of the following problem is that in the process of solution search the number of frontages and cutting details is steady, but their order is variable. The modifications of the classical genetic algorithm allowed us to develop a new algorithm of solution search. The results of the solution were used to create a DLL-library, which is easily integrated in different kinds of application software.

Keywords: rationalization/efficiency, unidimensional cutting, genetic algorithm

Introduction. Nowadays, furniture manufacturing enterprises are facing the necessity to reduce the amount of odds during unidimensional pattern cutting, especially when they are dealing with furniture frontages. These days, when the information acts as a product and the technologies are developing rapidly, the market offers a wide range of software (SW) to deal with such tasks. The main disadvantage of such SW is a strict limitation of its manufacturing application capability. This kind of SW is ready-made products, while the process of rationalization of pattern cutting is a temporary stage of manufacturing on enterprises. The results of different graphic-engineering systems are the initial data for the rationalization cutting process and the results may be applied, for instance, for creating and printing cutting charts. Also an important fact is that the existing programs may not always fulfill all the enterprise criteria.

The authors of this article were asked to create a multifunctional library, that allows to rationalize unidimensional cutting MDF (medium density fiber) frontage bases with minimum loses in a limited period of time. The existing programs allow neither to involve them into inner manufacturing process, nor to maintain all the enterprise criteria.

We were given electronic version of a specification of needed details in *txt format by the enterprise-client. The specialists of this enterprise developed utility program for converting *txt format into *xml format to automatize the process of information processing. This program recorded the number of whole frontage pieces and already cut frontages, as well as all the details that were made from the frontage. Converter specifications are the initial data for such library functions as rationalization and development of cutting charts. The authors of this article had to create and develop such general library.

The results of determination of cutting charts in *xml format were the bases to create a separate utility program, that could develop and print stickers. There were codes and sizes of the details printed

DOI: 10.15276/opu.3.53.2017.08

© 2017 The Authors. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

on the stickers. Because of the sticky bases, the paper would be easy to stick on each detail and this could make the joining process faster, as we omitted the phase of detail fitting.

A step-by-step information processing of the MDF frontage cutting is shown in Fig. 1.

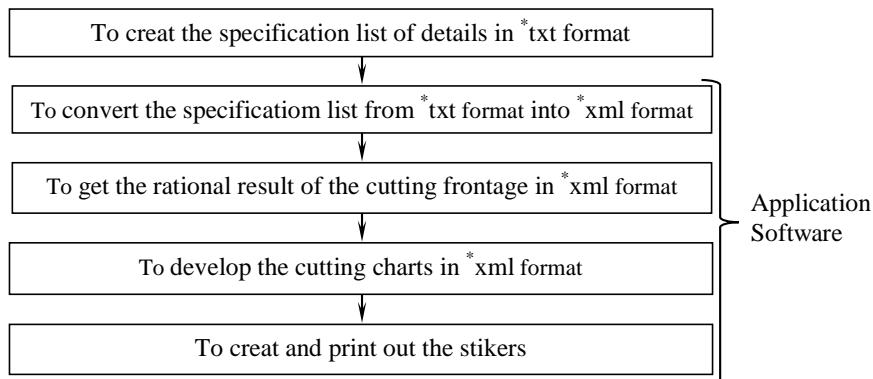


Fig. 1. Step-by-step information processing scheme of the MDF frontage cutting

Purpose and objectives of the study. The purpose was to rationalize the cutting chart of the frontage. A big quantity of different whole frontage pieces was given. There was given the net weight of frontage odds, which size was also given and could be used in manufacturing. The width of frontage was given as well as the minimum length of frontage odds, which are going to be used in manufacturing later. Also the specification sheet of the details that needed to be cut out was given.

We need to design and create the cutting chart in a way that all the details would be put rationally on the whole frontages or on the frontage odds, at the same time reduce the usage of whole frontages as well as reduce the quantity of odds production.

Analysis of literary data and formulation of the problem. The analysis of special literature has shown that this problem can be classified as a NP-completed problem [1], and the category is “containerization” [1], that’s why algorithm development for finding optimal and approximate solutions is a quite complicated task. In [2] we studied the dialog version of the containerization problem with open end, where the container could be filled till the level, which overloads its capacity. This problem is NP-hard. In [3] approximate algorithms of FFF and FFH solutions for dialog version of the containerization problem are shown, where there are different sized containers. However, in case when all the containers are of the same size these algorithms coincide with FF algorithms. In dialog versions of the problems the items come one by one and have to be allocated to the container according to their arrival. These algorithms allow us to get the rational result in a moment, while using direct enumeration method the time spent rises to dozens of minutes, which is inexcusable on large enterprises.

Materials and methods of studying. Pre-modeling has shown that the set problem could not be fulfilled by above mentioned algorithms, as the results are far from rational ones. On the basis of further researches in order to find the algorithm to fulfill the set task we decided to apply the elements of the GA theory [4]. The classical GA does not give the chance to find a solution to this problem, as the gene in chromosome may take the value either 0 or 1, while the gene set may differ. Modifying the classical GA in varied ways shows the wide range of usage.

As we have analyzed the results of the research, where the gene could take the value from 0 to 7, we offer to use the modified classical GA to solve the rationalization issue of the cutting chart.

Let us accept that each gene describes some specific construction that consists of:

- frontage index and type;
- frontage length – whole or residue;
- list of details, that could be cut out from this frontage;
- level of fullness.

Frontage index is a unique number, which is given to the frontage to differentiate the details on the enterprise and to make the process of recording or fitting easier.

The level of fullness is the sum of lengths of cut details, taking into account the width of kerf.

Let us accept that the chromosome is a constant sequence of genes-containers, which we are not subject to change and are put in a descending order. Here is an example for better understanding of the information. Let us suppose that chromosome is described by some specific set of genes given below, where 1, 2 and 3 are frontage odds and 5, 6 and 7 are whole frontage pieces.

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 7 | 7 | 6 | 6 | 5 | 5 | 5 | 3 | 3 | 3 | 3 | 2 | 2 | 1 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

In order to increase the quantity of potential solutions it is preferable to provide the quantity of the whole frontage pieces that would be enough to solve a given problem. As we use the frontage odds while fulfilling the task, this requirement is unnecessary to observe.

Let us assume that this is a list of details that need to be cut:

| | | | | | | |
|---|---|---|---|---|---|---|
| 4 | 4 | 3 | 3 | 3 | 2 | 2 |
|---|---|---|---|---|---|---|

The rationalization process refers to the search of the most efficient solution of the problem. Some possible allocation schemes of details in containers are shown in Fig. 2.

| | | | | | | | | |
|----------|---|----------|---|---|----------|---|---|---|
| 7 | | 7 | 4 | 3 | 7 | | | |
| 7 | | 7 | 4 | 3 | 7 | 4 | 3 | 2 |
| 6 | | 6 | | | 6 | | | |
| 6 | | 6 | 3 | 2 | 6 | 3 | 3 | |
| 5 | 4 | 5 | 2 | | 5 | 4 | | |
| 5 | 4 | 5 | | | 5 | | | |
| 5 | | 5 | | | 5 | | | |
| 3 | 3 | 3 | | | 3 | | | |
| 3 | 3 | 3 | | | 3 | | | |
| 3 | 3 | 3 | | | 3 | | | |
| 3 | | 3 | | | 3 | | | |
| 2 | 2 | 2 | | | 2 | | | |
| 2 | 2 | 2 | | | 2 | | | |
| 1 | | 1 | | | 1 | | | |
| 1 | | 1 | | | 1 | | | |
| <i>a</i> | | <i>b</i> | | | <i>c</i> | | | |

Fig. 2. Allocation schemes of details in containers: maximal usage of frontage odds (a); usage of only whole frontages (b); overloaded container (c)

In scheme c, the container assuming the value 7 is overloaded, as the total length of the details is bigger than the size of the container, that is $7 < 4+3+4$. It is the so-called value of excessive load of the container. To avoid such situations, the value of excessive load of containers has to be determined as one of the main criteria, which is used in target function.

The researches have shown that the level of adaptation of each chromosome is determined by different criteria.

1. The total value of excessive load of the containers, which is determined as the sum of length differences of all details l_{sum} and its actual length l_m , while the value l_{sum} has to be bigger l_m :

$$L^{exc} = \sum l_{sum} - l_m, \quad l_{sum} > l_m. \quad (1)$$

2. The index of total value of dependence of containers excessive load L^{exc} on total length of all details L^{sum} :

$$L_1 = \frac{L^{exc}}{L^{sum}}. \quad (2)$$

3. The index of dependence of already used frontages quantity N^n on the total quantity of whole frontage pieces N^{sum} :

$$N_1 = \frac{N^n}{N^{\text{sum}}}. \quad (3)$$

4. The saturation exponent K^{sat} , which is an integral quantity and describes the fullness of container capacity (loadedness), and which means while maximal loadedness index is equal to “1”, minimal loadedness index is equal to “0”. At the same time further researches have shown that efficient result could be received in case when the value K^{sat} depends on the value L^{exc} and has nonlinear response. The following functions describe nonlinear response and determination of saturation exponent:

$$f = \begin{cases} 1 - e^{-\frac{(l_{\text{sum}} - l_m)^2}{0.125 l_{\text{sum}}}}, & \text{if } L^{\text{exc}} = 0, \\ 1, & \text{if } L^{\text{exc}} > 0; \end{cases} \quad (4)$$

$$K^{\text{sat}} = \frac{\sum f}{N^m + N^n}. \quad (5)$$

Criterion function (target function), which is used to determine the level of adaptation of each chromosome, is determined the following way:

$$Z = 10^3 L_1 + 10^2 N_1 + 10 K^{\text{sat}}. \quad (6)$$

In case, when the containers excessive loadedness is absent, we use $L_1 = 0$. In case, when the problem was solved without any usage of whole frontages, but with the frontage odds only, we use $N_1 = 0$.

After the abovementioned operation was fulfilled it was needed to develop the rational search algorithm, which enables effective problem solution. As the set of genes-containers is steady, the mutation and crossing-over operations that are used in classical GA cannot be applied. In this case the loadedness of the containers is the only thing that is subject to change.

After analysis and testing of all ways of completing the task, the algorithm of rational search was the following.

1. To make the dimension of the problem (task) limited, the first thing to determine is maximum quantity of whole frontage pieces of all typical sizes. Apart from the length of details, it is necessary to consider the frontage kerf. When the total frontage length is excessive, we add the next frontage piece.

2. The next step is to create the initial population of chromosomes. According to FFD algorithm, if we take the first suitable unit in descending order, then there is only one solution. By random loading of containers with details 10 possible solutions are determined. In case of usage of a specific size of the whole frontage we add one more solution according to FFD algorithm.

3. After all manipulations were done, the basic cycle starts. The condition is stated in the cycle: if the value of target function (TF) of the best chromosome remains steady during 30 and more iterations, the manipulations (calculations) are over.

For each chromosome:

1. In a random manner two containers are chosen. In a random manner we take the detail from the first container and put it into the second container. The received chromosome is added to the new population.

2. In order to check the improvement of value of loadedness we transfer the content of two containers in any other containers of smaller size. If it is possible new chromosomes are added to a new population.

3. We add one more chromosome to a new population by replacing the content of two containers.

4. If there are overloaded containers, the details are moved to the first suitable containers. The new chromosome is added to a new population.

5. If there is an odd chromosome in the initial population see point 1). If all the chromosomes of initial population are sorted we go to point 6).

6. For all the chromosomes of a new population we calculate the value of the TF. After that we select 30 chromosomes, which are the most adapted.

7. The value of the best chromosome TF is determined. Then we move to the initial phase of the basic cycle.

Testing of the algorithm and development of the DLL (dynamic link library) was conducted on Microsoft Visual C++2006 software. The testing was done on ECM (electronic computing machine) with the following characteristics: Windows 7, Max x64, Intel Core i3-2350M, 2.3 GHz, 1 Gb RAM. In order to conduct the calculation (manipulation) for up to 30 details it took 4 s, and for up to 100 details it took 15 s. The given algorithm allowed to rationalize the solution (development of the task), which we got with the help of FFD algorithm.

The mentioned library was successfully developed and implemented at the client's enterprise.

Conclusions. In the article the usage of GA for solving the problem of cutting is demonstrated. The developed algorithm of solution search on the basis of classical GA increases the efficiency of the cutting process in comparison with traditional methods.

The general DLL was developed, the DLL presents the problem solution in xml format and is easily integrated in different kinds of software.

Литература

1. Гэри М. Джонсон Д. Вычислительные машины и труднорешаемые задачи. Москва: Мир, 1982. 416 с.
2. Zhang G. Parametrized On-Line Open-End Bin Packing. *Computing* 60. 1998. P. 267–273.
3. Zhang G. Worst-Case Analysis of the FFH Algorithm For-Online Variable-Sized Bin Packing. *Computing* 56. 1996. P. 165–172.
4. Рутковская Д., Пилинский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы / Пер. И.Д. Рудинский. Москва: Горячая линия Телеком. 2006. 452 с.

Reference

1. Gary, M.B. & Jonson, D. (1982). *Computing machines and hard-to-solve problem*. Moscow: Mir.
2. Zhang, G. (1998). Parametrized On-Line Open-End Bin Packing. *Computing* 60, 267–273.
3. Zhang G. (1996). Worst-Case Analysis of the FFH Algorithm For-Online Variable-Sized Bin Packing. *Computing* 56, 165–172.
4. Rutkovskaya, D., Pilinskiy, M., & Rutkovskiy, L. (2006). *Neural networks, genetic algorithms and fuzzy systems*. (I.D. Rudinskiy, Trans). Moscow: Goryachaya liniya Telekom.

Received November 14, 2017

Accepted November 29, 2017