Mathematical and Software for Designing Rational Schemes of Cutting Rectangular Materials on Flat Geometric Objects With Complex Configuration of External Contours

Viktor Chuprynka (0000-0001-6869-3091), Natalia Chuprynka(0000-0003-1507-489X)

Kyiv National University of Technology and Design,
street N.-Danchenko 2, 01011, Kyiv, Ukraine
Chuprinka_V_I@ukr.net, Chuprinka_N_I@ukr.net

Abstract. The paper considers mathematical and software for automated design of rational schemes of cutting rectangular materials into flat geometric objects with a complex configuration of external contours. To solve this problem successfully, it is divided into the following four tasks: construction of a set of dense tabs for each of the flat geometric objects; generating a rational cutting scheme (sections) for each of the flat geometric objects; dense placement of sections in the cutting scheme; interactive adjustment of the received cutting scheme. For each of these four problems, mathematical models and methods for solving them are proposed. The tasks were implemented in software for the design of rational schemes for cutting rectangular materials on flat geometric objects with a complex configuration of external contours.

Keywords: Rational cutting · System placement, · Interactive adjustment · Salesman task

1 Introduction

In any industry, the issue of material consumption in production has always been very relevant. High material consumption and significant cost of materials used make the task of minimizing costs especially important for the footwear industry. Rational and economic costs of material and energy resources, as well as protection of the environment from pollution by waste, which arise during the cutting of materials are important tasks of production. Automated design of rational cutting schemes will allow rational use of materials when cutting parts, reduce the amount of waste that pollutes the environment, reduce the cost of products.

2 Statement of the problem

Many works have been devoted to the design of rational schemes for cutting materials into flat geometric objects. Mathematical models of compact arrangement of convex flat geometric objects are presented in [1-3]. Guo et al. [1] proposed a tree representation called O-tree: Two ordered trees for the horizontal and vertical directions are used to represent a coded solution Chang et al. [2] extended the result by Guo et al. [1]. They proposed another tree representation called B*-tree; it is easy to implement this data structure and a decoding algorithm for B*-tree runs in linear time with respect to the number of items. Sakanushi et al. [3] proposed another coding scheme called quarter-state sequence. They utilized a string of items and labels to represent a solution and their decoding algorithm runs in linear time of the number of items. But in most cases, the details of the shoes are not convex flat geometric objects.

Okano[4] designed his algorithm for the irregular two-dimensional bin packing problem; however, his technique is also useful for treating the irregular strip packing problem. Lesh et al. [5] proposed a stochastic search variation of the bottom left heuristics for the strip packing problem. Their algorithm outperforms other heuristic and meta heuristic algorithms based on the bottom left strategy reported in the literature. Imahori et al. [6] proposed an improved meta heuristic algorithm based on sequence pair representation. Meta heuristic algorithms generate numer-ous number of coded solutions and evaluate all of them. Hence, the efficiency of meta heuristic algorithms strongly depends on the time complexity of decoding algorithms.

Genetic algorithms are used in [7-10]. But these algorithms do not always give a satisfactory result in a limited time. Therefore, the task of this work is to develop a method of automatic design of rational schemes of cutting materials by any configuration of the outer contour for flat geometric objects with a complex configuration of outer contours. But the problem of automated design of rational schemes of cutting rectangular materials into flat geometric objects was not considered in such a statement. The technological formulation of this problem is as follows: on a roll of material of limited length to place a given set of flat geometric objects taking into account technological requirements (orientation of these objects, minimum technological distance Δ between two neigh boring objects in the cutting scheme), so that waste was the smallest.

The mathematical formulation of this problem is as follows: for a given set of flat geometric objects S_i and a given number of these objects $\check{N}_i$, where $i = 1..q$, from the set of admissible layouts in a rectangular region of length *DlMat* and width *ShMat* find such a rational layout, which provided the maximum value of the goal function:

$$F = \max\{F_j\}, \text{ where } F_j = \frac{\sum_i^q \left|S^i\right| \cdot N_i^j}{DlMat_j \cdot ShMat} \text{ and } j = 1,2...\infty$$

$DlMat_j$ – the length of the rectangular area occupied by *j* rational layout.

This takes into account the technological requirements (orientation of these objects, the minimum technological distance Δ between two neigh boring objects in a rational layout).

Based on the practice of cutting in light industry, consider such a simplified mathematical model of the problem. Consider three consecutive tasks:

Task A – System placement N_i^j ($N_i^j \le \check{N}_i$) of flat geometric objects S_i, $i = 1..q$ in a rectangular region of fixed width *ShMat* (Section);

Task B - Designing a cutting scheme from sections (Scheme);

Task C - Interactive adjustment of the scheme of the designed cutting scheme (Interactive adjustment).

We give a mathematical formulation of each of these three problems.

Task A - Section. For N_i^j ($N_i^j \le Ň_i$) flat geometric objects from the set of system schemes of cutting find such a region (section $\hat{S}_i$) of rectangular shape S^i of size *ShMat x Dl_S$_i$*, $i = 1..q$, for which the objective function takes a maximum, that is

$Q_i = \max\{Q_i^j\} = \max\left\{\frac{\left|S^i\right| \cdot N_i^j}{ShMat \cdot Dl\_S_{i^i}^j}\right\}$, where $N_i^j \le Ň_i$ and $j=1,2..p_i$.

Task B - Scheme. For the cyclic permutation $\mu = [\hat{S}_1, \hat{S}_2 ... \hat{S}_q]$ of sections $\hat{S}_i$, $i = 1,2..q$ to find such permutation μ* is μ that at dense combination of sections the formed scheme will have the smallest length, that is $L^* = L(\mu^*) = \min_{\mu}(L(\mu))$ ·

Task C - Interactive adjustment. In many cases, it is not possible to automatically build cutting schemes that would meet the technological requirements. Therefore it is necessary to adjust the received schemes or to build new in an interactive mode. To successfully solve this problem, you need to solve two problems: the placement of flat geometric objects in a rectangular area of given size with control:

- belonging of a flat geometric object to a rectangular area of specified dimensions;

- not the intersection of the active flat geometric object with already placed in the rectangular area of flat geometric objects;

remove any previously placed flat geometric object from the cutting scheme.

Since flat geometric objects in most cases have a complex configuration of the outer contour, which cannot be described analytically, we will approximate it. For the approximation, we choose the piecewise-linear method of approximation as one that does not impose restrictions on the outer contour of a flat geometric object. In a piecewise linear approximation, the outer contour of a flat geometric object will be approximated by a polygon with vertex coordinates. Therefore, in what follows we will assume that flat geometric objects are polygons with a known number of vertices and their coordinates.

Determine the maximum values of the coordinates of the vertices of the approximate polygon for a flat geometric object:

$$\begin{aligned} MaxX_i &= \max\left\{X_j^i\right\} \\ MaxY_i &= \max\left\{Y_j^i\right\} \\ MinX_i &= \min\left\{X_j^i\right\} \\ MinY_i &= \min\left\{Y_j^i\right\} \end{aligned}, \ j = 1..k_i \quad .$$

List the coordinates of the approximating polygons as follows:

$$\begin{aligned} X_j^i &= X_j^i - (MaxX_i + MinX_i)/2 \\ Y_j^i &= Y_j^i - (MaxY_i + MinY_i)/2 \end{aligned}, \ j = 1..k_i$$

After listing the coordinates of the approximating polygons we obtain the following expressions for:

$$MaxX_i = -MinX_i = MX_i = (MaxX_i + MinX_i)/2$$

$$MaxY_i = -MinY_i = MY_i(MaxY_i + MinY_i)/2$$

After recalculating the coordinates of the vertices of the approximating polygons for flat geometric objects S^i, $i = 1..q$ the coordinates of the vertices of the approximating polygon will be determined relative to the centre of the rectangles described around the flat geometric objects S^i. These points are called the poles of flat geometric objects S^i.

3. Task A - Section

To generate a set of system schemes of cutting for a flat geometric object, it is necessary to generate a set of lattice dense stacks, as a set of prototypes of cutting schemes.

3.1 Tight styling

A system of flat geometric objects S^i, $i=1,2..p$ forms a stack on the plane, if for each pair of these flat geometric objects the condition of their mutual intersection is fulfilled. This condition can be represented as follows:

$$\begin{matrix} S_i \cap S_j \neq 0 \\ \operatorname{int} S_i \cap \operatorname{int} S_j = 0 \end{matrix}, \text{ where } \operatorname{int} S_i = S_i - S_i^{\wedge}$$

and $S_i^{\wedge} -$ the boundary of a flat geometric object S_i .

We denote by $S + \vec{a}$ a flat geometric object that arises when each point of a flat geometric object is moved to a vector, and we will call this flat geometric object S a translation of a flat geometric object $\vec{a}$.

Set of vectors

$$\vec{r} = n\vec{a}_1 + m\vec{a}_2 , \quad n, m = 0, \pm 1, \pm 2, \ldots, \tag{1}$$

is called a lattice with basis $\vec{a}_1, \vec{a}_2$, where $\vec{a}_1, \vec{a}_2 -$ linearly independent vectors, and is denoted $\Lambda = \Lambda(\vec{a}_1, \vec{a}_2)$.

Consider a system of flat geometric objects

$$\bigcup_{n,m} S_{nm}, \qquad n, m = 0, \pm 1, \pm 2, \ldots, \tag{2}$$

which consists of translations $S_{nm} = S + n\vec{a}_1 + m\vec{a}_2$ of a flat geometric object S into lattice vectors $\Lambda = \Lambda(\vec{a}_1, \vec{a}_2)$.

If the system (2) is a layout, then such a layout is called a lattice layout of flat geometric objects S , made behind the lattice $\Lambda = \Lambda(\vec{a}_1, \vec{a}_2)$. The lattice Λ in this case is called permissible for laying flat geometric objects S .

In the future, only those flat geometric objects that can be translated into each other by translation to some vector will be considered the same. From this point of view, the basic flat geometric object $S(0)$ and the same flat geometric object $S(\pi)$, but rotated by 180º are further considered as different. Let's make from these figures dense single-row placements:

$$\bigcup_n (S(0)+n\vec{a}), \qquad \bigcup_n (S(\pi)+n\vec{a}), \qquad n=0,\pm1,\pm2,\ldots, \quad . \quad (3)$$

By alternating the formed rows and pressing them tightly, we create a laying W on the plane so that the mutual arrangement of the row consisting of flat geometric objects $S(0)$, in relation to the adjacent rows of flat geometric objects $S(\pi)$, in the whole laying was the same (Fig. 1). .

Laying W is a combination of two lattice layouts $\bigcup_{n,m} S(0)+n\vec{a}+m\vec{b}$ and $\bigcup_{n,m} S(\pi)+n\vec{a}+m\vec{b}$, where $n,m=0,\pm1,\pm2,\ldots$ made on lattices $\Lambda=\Lambda(\vec{a},\vec{b})$ with the same basis $\vec{a},\vec{b}$. Therefore, the layout W of the form is called double lattice laying of flat geometric objects $S(0)$ and $S(\pi)$.

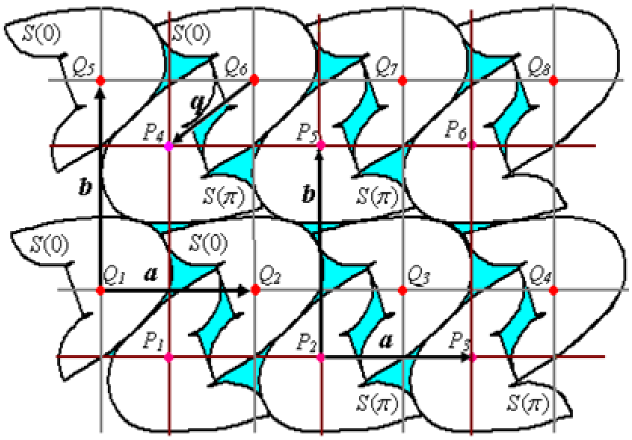

Fig.1. Dense lattice laying

A system consisting of two simultaneously defined on the plane identical but not coinciding lattices with nodes points Q_j, $j = 1,2..m$ and P_i, $i = 1,2..n$ one of which is a translation of the second to some vector, is called double lattice and is denoted $W=W(\vec{a},\vec{b},\vec{q})$. Here is the basis of each of the lattices of the system and the vector of their mutual displacement. ($\boldsymbol{Q_1Q_2} = =\boldsymbol{P_2P_3}$, $\boldsymbol{Q_1Q_5} = \boldsymbol{P_2P_5}$ and $\vec{q}=\boldsymbol{Q_6P_4}$ (Fig. 1).

The problem of generating a set of dense tabs is considered in detail in [11]. To reproduce dense lattice laying $\Lambda=\Lambda(\vec{a},\vec{b})$ on a single lattice it is necessary to determine two lattice vectors $\vec{a},\vec{b}$. To reproduce dense lattice laying on a double lattice $W=W(\vec{a},\vec{b},\vec{q})$ it is necessary to determine two lattice vectors $\vec{a},\vec{b}$ and the lattice shift vector $\vec{q}$.

3.2 Generating a set of sections

The source information for generating the set of allowable sections for a flat geometric object S^i will be the set of allowable dense stacks built on single

$$\Lambda_p = \Lambda(\vec{a}_p, \vec{b}_p),\ p = 1,2...p_0$$

and double lattices $W_r = W(\vec{a}_r, \vec{b}_r, \vec{q}_r),\ r = 1,2...r_0$.

To successfully solve the problem Section it is necessary to describe its structural components, namely:

- analytical description of the rectangular area Ω_i and the size *ShMat* x Dl_S_i, in which it is necessary to tightly place flat geometric objects S^i;
- determination of parameters that unambiguously determine the position of a flat geometric object S^i in a rectangular region Ω_i and given dimensions;
- conditions under which a flat geometric object S^i is in the middle of the rectangular region Ω_i;
- mathematical description of the set of admissible solutions;
- goal function.

We will connect the coordinate system with a material that has a rectangular shape (rolls or sheets). Let the origin be in the lower left corner of the material. Then the allowable area (Fig. 2), where flat geometric objects can be placed can be represented as a system of inequalities:

$$\begin{cases} 0 \le X \le Dl\_S_i \\ 0 \le Y \le ShMat_i \end{cases}, \tag{4}$$

where Dl_S_i, *ShMat* - respectively the length and width of the rectangular region Ω_i.

To unambiguously display a flat geometric object on the material, you need to know the following information:

i is the code of the flat geometric object S^i that is placed (in our case $i = 1,2..q$);

Xp_m, Yp_m, $m = 1,2..t$ - coordinates of the pole of a flat geometric object S^i relative to the coordinate system associated with the rectangular region Ω_i, on which is located

Pr - a sign of the position of the part (in our case: 0 - the main position; 1 - a flat geometric object rotated 180° relative to the main position)

We find the conditions under which a flat geometric object S^i is inside the rectangular region Ω_i. Obviously, if the pole of a flat geometric object S^i inside the rectangle *ABCD*, then this flat geometric object will be inside the rectangular region Ω_i (Fig. 2) Then it is obvious that if the inequality holds

$$\begin{cases} MX_i \le Xp_m \le Dl\_S_i - MX_i \\ MY_i \le Yp_m \le Sh - MY_i \end{cases}, \tag{5}$$

then a flat geometric object S^i that will be placed in a rectangular area Ω_i will never go beyond that area.

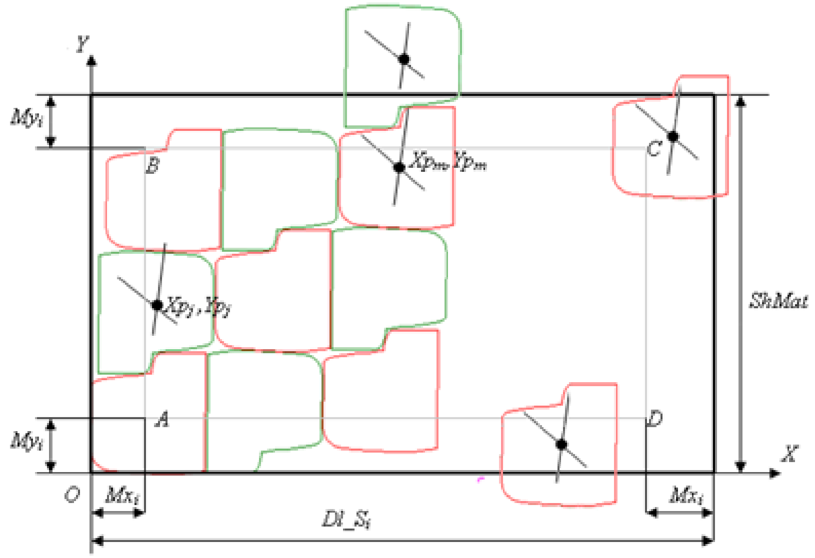

Fig. 2. Determining the mutual position of the plane geometric object S^i and rectangular area Ωi

For any lattice $W = W(\vec{a}, \vec{b}, \vec{q})$, the values of the functionals $N_\Omega^{(1)}(\vec{a}, \vec{b}, \vec{q})$, $N_\Omega^{(2)}(\vec{a}, \vec{b}, \vec{q})$ are equal to the number of pairs of integers (n, m) from the set $0, \pm 1, \pm 2, \ldots$ that satisfy inequalities (5). Using the *sign (x)* function

$$sign(x) = \begin{cases} 1, & if\ x \geq 0 \\ -1, & if\ x < 0 \end{cases},$$

the goal function can be written as: $N_\Omega(\vec{a}, \vec{b}, \vec{q}) = N^{(1)}{}_\Omega(\vec{a}, \vec{b}, \vec{q}) + N^{(2)}{}_\Omega(\vec{a}, \vec{b}, \vec{q})$, where

$$\begin{aligned} N^{(1)}{}_\Omega(\vec{a}, \vec{b}, \vec{q}) = \frac{1}{16} \sum_{n,m} & \left(1 + sign(x'_{nm} - Mx_i)\right) \cdot \\ & \cdot \left(1 + sign(Dl\_S_i - Mx_i - x'_{nm})\right) \cdot \\ & \cdot \left(1 + sign(y'_{nm} - My_i)\right) \cdot \\ & \cdot \left(1 + sign(Sh - My_i - y'_{nm})\right) \cdot \end{aligned} \tag{6}$$

$$\begin{aligned} N^{(2)}{}_\Omega(\vec{a}, \vec{b}, \vec{q}) = \frac{1}{16} \sum_{n,m} & \left(1 + sign(x''_{nm} - Mx_i)\right) \cdot \\ & \cdot \left(1 + sign(Dl\_S_i - Mx_i - x''_{nm})\right) \cdot \\ & \cdot \left(1 + sign(y''_{nm} - My_i)\right) \cdot \\ & \cdot \left(1 + sign(Sh - My_i - y''_{nm})\right) \cdot \end{aligned} \tag{7}$$

where
$$\begin{aligned} x'_{nm} &= n \cdot Xa + m \cdot Xb + Mx_i \\ y'_{nm} &= n \cdot Ya + m \cdot Yb + My_i \\ x''_{nm} &= n \cdot Xa + m \cdot Xb + Mx_i + Xq \\ y''_{nm} &= n \cdot Ya + m \cdot Yb + My_i + Yq \end{aligned} \quad \text{and} \quad \begin{aligned} \vec{a} &= (Xa, Ya) \\ \vec{b} &= (Xb, Yb) \\ \vec{q} &= (Xq, Yq) \end{aligned}, \quad n, m = 0, \pm 1, \pm 2 \ldots$$

4 Task B - Scheme

Often the cutting scheme consists of separate schemes, which we will call sections. These sections are combined as rectangles described around them when constructing the cutting scheme. And this is not always rational, as in this case the sections are not aligned tightly everywhere (Fig. 3).

4.1 Tight alignment of sections

Let the length of the *j*-th section be equal to Dl_S_j and the coordinates of the poles of the parts be where $k = 1,2..n_j$. To tightly align the *j*-th and *i-th* sections, it is necessary to find new coordinates Xp_k^i, where $k = 1,2..n_i$. Their initial value can be defined as $Xp_k^i = Xp_k^i + Dl\_s_j$, that is for alignment without taking into account the possibility of tight alignment of sections. For tight alignment of sections it is necessary to find the right boundary of the j-th section and the left boundary of the i-th section after the previous alignment. By the right boundary of a flat geometric object, t = 1,2… tR we mean the contours of this object, which are to the right of the reference line drawn from the right edge of the j-th section at a distance Dl_dj / 2 parallel to the axis 0Y , where Dl_dj is the length of the rectangle, which is described around a flat geometric object Sj (Fig. 3).

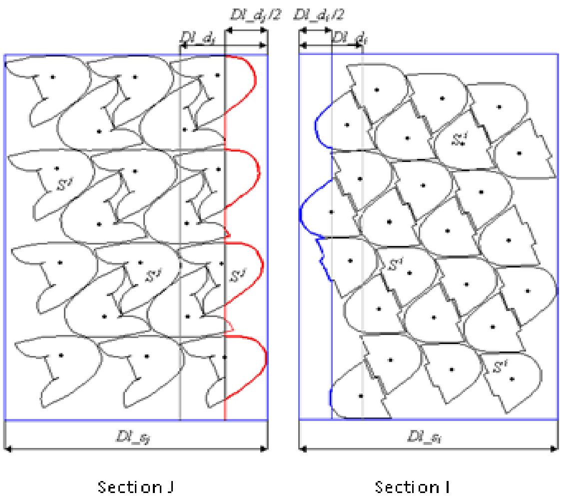

Fig. 3. Sections J and I before alignment

The sides of the rectangle are parallel to the sides of the *j*-th layout. By the left boundary G_{it}^{L}, $t=1,2...t_L$ of the flat geometric object, $t = 1,2... t_L$, we mean the contour of the flat geometric object which is to the left of the reference line, drawn from the left edge of the *i*-th section at a distance $Dl\_d_i / 2$ parallel to the axis $0Y$, where Dl_d_i is the length of the rectangle , which is described around a flat geometric object S^i (Fig. 3). The sides of the rectangle are parallel to the sides *i*-th layout. The right boundary of the *j*-th section consists of the right boundaries G_{jt}^{R}, $t=1,2...t_R$ $t = 1,2... t_R$ of flat geometric objects, for which the inequality holds $Xp_k^j > Dl\_s_j - Dl\_d_j$, where Dl_s_j is the length of the *j*-th section (Fig. 3). The left boundary of the *i*-th layout consists of the left boundaries G_{it}^{L}, $t = 1,2... t_L$ of flat geometric objects, ,for which the inequality holds $Xp_k^i < Dl\_d_i$ (Fig. 3). To closely match the *j*-th and *i*-th layouts, it is necessary to tightly align the right boundary of the *j*-th section and the left boundary of the *i*-th section (Fig. 4).

To do this first we select the left boundary G_{it}^{L}, $t = 1,2... t_L$ for each flat geometric object S^i of the *i*-th section, the poles of which satisfy the condition $Xp_k^i < Dl\_d_i$ and the right boundary G_{jt}^{R}, $t = 1,2... t_R$ for each flat geometric object S^j *j*-th sections whose poles satisfy the condition $Xp_k^j < Dl\_s_j - Dl\_d_j$.

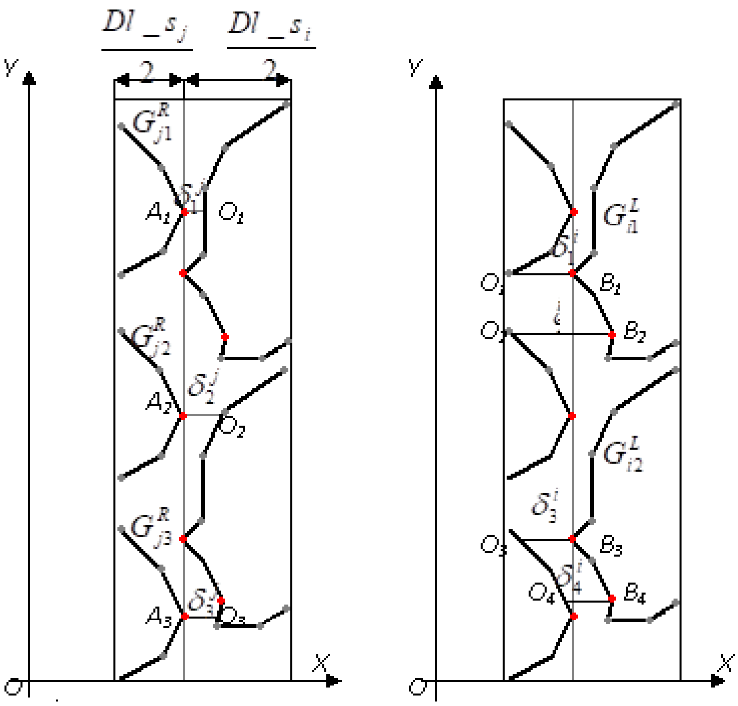

Fig. 4. The magnitude δ of the possible displacement of the sections

Then the left boundary G_i^L for the *i*-th section can be represented as the union of the left boundaries of flat geometric objects S^i, that is $G_i^L = \bigcup_{t=1}^{t_L} G_{it}^L$. Similarly, the right boundary G_j^R for the *j*-th section can be represented as the union of the right boundaries of flat geometric objects S^j, that is $G_j^R = \bigcup_{t=1}^{t_R} G_{it}^r$. For each left boundary G_i^L we find points for which the *X* coordinate reaches a local minimum. Let it be an array of points A_k (Xa_k, Ya_k), $k = 1,2..k_i$ (Fig. 4). For each right boundary G_j^R we find points for which the *X* coordinate reaches a local maximum. Let it be an array of points B_k (Xb_k, Yb_k), $k = 1,2..k_j$. Draw from each point A_k (B_k) a straight line parallel to the axis *OX* to the intersection with the left boundary G_i^L (right boundary G_j^R) *i*-th (*j*-th) section (Fig. 3-5). Find the length of the segments $A_kO_k = \delta k_1$, $k = 1,2..k_j$ ($B_kO_k = \delta k_2$, $k = 1,2..k_i$).

We will find $\delta_{ji} = \min(\delta^1, \delta^2)$, where $\delta^1 = \min_{k=1,2..k_j}(\delta_k^1)$, $\delta^2 = \min_{k=1,2..k_i}(\delta_k^2)$. The found value of δ_{ji} will be the value by which you want to shift the i-th layout so that it fits snugly with the *j*-th layout (Fig. 5). Then the coordinates of the poles of the parts in the *i*-th section after close alignment with the j-th section will take the following form:

$$Xp_k^{Hob_i} = Xp_k^i + Dl\_s_j - \delta_{ji} \text{ and } Yp_k^{Hob_i} = Yp_k^i,$$

where $k = 1,2, .. h_i$.

Now we can always closely match the two layouts (sections) $\hat{S}_i$ and $\hat{S}_j$ by calculating δ_{ij}, but we must remember that δ_{ij} determines the tight fit when $\hat{S}_i$ the layout is on the left and $\hat{S}_j$ the layout is on the right, δ_{ji} determines the tight fit when $\hat{S}_j$ layout is on the left, and $\hat{S}_i$ layout is on the right, as $\delta_{ij} \neq \delta_{ji}$.

4.2 Search for the optimal permutation of sections

The mathematical model of this problem can be represented as follows. You need to minimize the function:

$$L = \sum_{i=1}^{q} Dl\_s_i - \sum_{i=1}^{q}\sum_{j=1}^{q} \delta_{ij} \cdot x_{ij} \tag{8}$$

with the following restrictions:

$$\begin{cases} \sum_{i=1}^{q} x_{ij} = 1 \\ \sum_{j=1}^{q} x_{ij} = 1 \\ x_{ij} = \begin{cases} 0 \\ 1 \end{cases} \end{cases} . \quad (9)$$

As a result of solving this problem, we obtain the order of alignment of the sections in the cutting scheme, which, when the sections are tightly placed, will ensure the minimum length of the cutting scheme. This task can be reduced to the task of a salesman by entering the following notation

$$L_o = \sum_{i=1}^{q} Dl\_s_i \quad . \quad (10)$$

Then

$$L_k = \sum_{i=1}^{q} Dl\_s_i - \sum_{i=1}^{q}\sum_{j=1}^{q} \delta_{ij} \cdot x_{ij} = L_o - \sum_{i=1}^{q}\sum_{j=1}^{q} \delta_{ij} \cdot x_{ij} =$$
$$= \sum_{i=1}^{q}\sum_{j=1}^{q} (\frac{L_o}{q} - \delta_{ij}) \cdot x_{ij}) = \sum_{i=1}^{q}\sum_{j=1}^{q} \chi_{ij} \cdot x_{ij} \quad (11)$$

where
$$\chi_{ij} = \frac{L_0}{q} - \delta_{ij} . \quad (12)$$

After that, the mathematical model of the problem can be represented as follows:

$$L_k^* = \min_{\mu}(L_k) = \sum_{i=1}^{q}\sum_{j=1}^{q} \chi_{ij} \cdot x_{ij} \quad (13)$$

with the following restrictions:

$$\begin{cases} \sum_{i=1}^{q} x_{ij} = 1 \\ \sum_{j=1}^{q} x_{ij} = 1 \\ x_{ij} = \begin{cases} 0 \\ 1 \end{cases} \end{cases} \quad (14)$$

This is a mathematical model of the problem of the salesman.

5 Task C - Interactive adjustment

In many cases, it is not possible to automatically build cutting schemes that would meet the technological requirements. Therefore it is necessary to adjust the received schemes or to build new in an interactive mode. To successfully solve this problem, you need to solve the following problems:

- placement of parts on the material of the specified dimensions and not crossing the boundaries of the material details;
- removal of any previously placed part from the cutting scheme;
- not the intersection of parts when placing them.

Let's dwell in more detail on each of the above tasks. The problem of placing parts on the material of a given size and not crossing the boundaries of the material details was discussed in detail in the first section.

To remove any previously placed part from the cutting scheme, it is necessary to identify the part that needs to be removed. To do this, you must decide whether the point is inside a convex-concave polygon. To speed up the algorithm for determining the mutual location of the point $O(X_0, Y_0)$ and the polygon A, consider the problem of mutual placement of the point $O(X_0, Y_0)$ and the rectangle described around the polygon P. Let this rectangle be defined by a system of inequalities:

$$\begin{cases} X^a_{\min} \le x \le X^a_{\max} \\ Y^a_{\min} \le y \le Y^a_{\max} \end{cases} \tag{15}$$

Then the point $O(X_0, Y_0)$ is located outside the polygon A, if it does not satisfy the system of inequalities (15), otherwise the point O may or may not belong to the polygon A(рис. 5). To clarify this fact, we use the method of angles[12].

Consider the method of angles [12] to solve the problem of belonging to a point. In this approach, it is necessary to define the concept of an angle with a sign. Suppose we have a vector $\boldsymbol{OA_i}$ and a vector $\boldsymbol{OA_{i+1}}$. Denote the angle between them by $\varphi_i = \angle A_i O A_{i+1}$, where $i=1,2...n_p-1$. The angle φ_i will be with a plus sign, when rotating the vector $\boldsymbol{OA_i}$ around the point O the closest path to the vector $\boldsymbol{OA_{i+1}}$ will be when rotating the vector $\boldsymbol{OA_i}$ counterclockwise, otherwise this angle φ_i will be negative. The point O will be outside the polygon A, if $\sum_{i=1}^{n_p-1}\varphi_i = 0^o$ (fig. 5.a). The point O is inside the polygon A, if $\sum_{i=1}^{n_p-1}\varphi_i = 360^o$ (fig. 5.b).

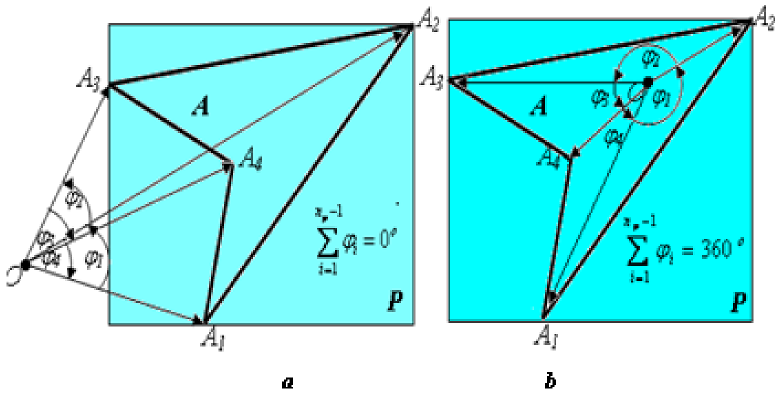

Fig. 5. Location of the point
a) outside the polygon b) in the polygon

To determine the total angle, it is necessary to find the elementary angles. Elementary angles will have a sign. To determine the sign of the elementary angle φ_i use the module of the vector product:

Determine the angle between the vectors $\boldsymbol{OA_i}$ and $\boldsymbol{OA_{i+1}}$. To do this, we find the modulus of the vector product and the scalar product of the vectors $\boldsymbol{OA_i}$ and $\boldsymbol{OA_{i+1}}$.. We introduce the notation: $\boldsymbol{a_i}= \boldsymbol{OA_i}=(Xa_i, Ya_i)=(Xi-X0, Y_i-Y_0)$; $\boldsymbol{b_i}= \boldsymbol{OA_{i+1}} =(Xa_{i+1}, Ya_{i+1})=(X_{i+1}-X_0, Y_{i+1}-Y_0)$;

Then $|[\boldsymbol{OA_i} \times \boldsymbol{OA_{i+1}}]| = |[\boldsymbol{a_i} \times \boldsymbol{b_i}]| = \begin{vmatrix} Xa_i & Ya_i \\ Xb_i & Yb_i \end{vmatrix} = Xa_i \cdot Yb_i - Xb_i \cdot Ya_i =$

$=|\boldsymbol{a_i}| \cdot |\boldsymbol{b_i}| \cdot \sin \varphi_i$,

$(\boldsymbol{OA_{i+1}} \cdot \boldsymbol{OA_{i+1}})=(\boldsymbol{a_i} \cdot \boldsymbol{b_i})=Xa_i \cdot Xb_i+ Ya_i \cdot Yb_i = |\boldsymbol{a_i}| \cdot |\boldsymbol{b_i}| \cdot \cos \varphi_i$.

From here:: $\sin \varphi_i = (Xa_i \cdot Yb_i- Ya_i \cdot Xb_i)/(|\boldsymbol{a_i}| \cdot |\boldsymbol{b_i}|)$,

$\cos \varphi_i=(Xa_i \cdot Xb_i- Ya_i \cdot Yb_i)/(|\boldsymbol{a_i}| \cdot |\boldsymbol{b_i}|)$,

where $|\boldsymbol{a_i}|=\sqrt{(X_i - Xo)^2 + (Y_i - Yo)^2}$ and $|\boldsymbol{b_i}|=\sqrt{(X_{i+1} - Xo)^2 + (Y_{i+1} - Yo)^2}$.

If $|[\boldsymbol{OA_i} \times \boldsymbol{OA_{i+1}}]| = |[\boldsymbol{a_i} \times \boldsymbol{b_i}]|>0$, then the angle will be positive.

If $|[\boldsymbol{OA_i} \times \boldsymbol{OA_{i+1}}]| = |[\boldsymbol{a_i} \times \boldsymbol{b_i}]|<0$, then the angle will be negative.

Then:

- if $\cos \varphi_i>0$, then $\varphi_j=arctg(|[a_i \times b_i]|/(a_i \cdot b_i)$;
- if $\cos \varphi_i=0$ та $\sin \varphi_i=1$, then $\varphi_i=\pi/2$;
- if $\cos \varphi_i=0$ та $\sin \varphi_i=-1$, then $\varphi_i=-\pi/2$;
- if $\cos \varphi_i<0$ та $\sin \varphi_i\geq 0$, then $\varphi_i=\pi+ arctg(|[a_i \times b_i]|/(a_i \cdot b_i)$;
- if $\cos \varphi_i<0$ та $\sin \varphi_i<0$, then $\varphi_i=-\pi- arctg(|[\boldsymbol{a_i} \times \boldsymbol{b_i}]|/ (\boldsymbol{a_i} \cdot \boldsymbol{b_i}))$.

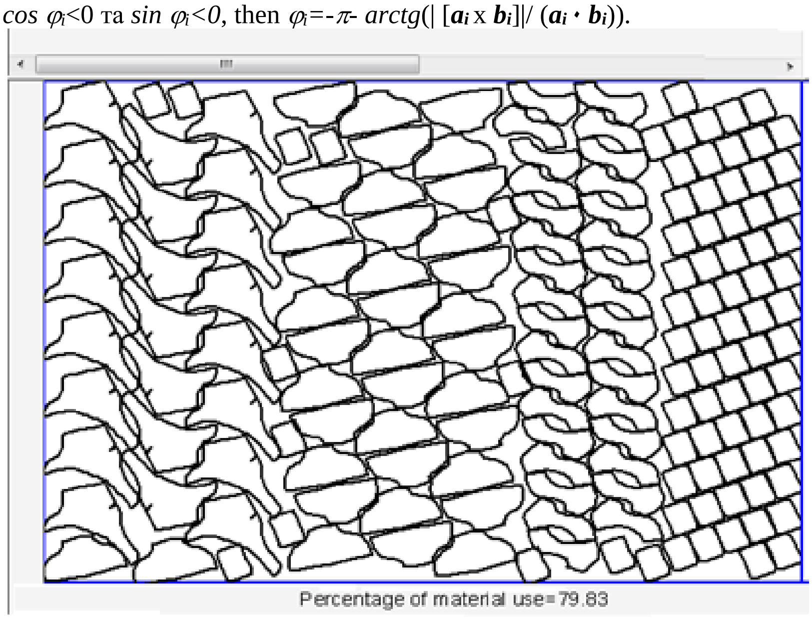

Fig. 6. An example of the designed rational scheme of cutting

To find out the intersection of two plane geometric objects, it is necessary to find out the intersection of the polygons *P* and *Q*, which approximate these objects.

The above-mentioned problems A-C were implemented in software for automated design of rational schemes of cutting rectangular materials into flat geometric objects of arbitrary shape of the outer contour, taking into account the need for these objects. An example of the designed rational scheme of cutting by means of the developed software is presented in fig. 6.

6 Conclusions

The article considers the task of computer-aided design of rational schemes for cutting rectangular materials onto flat geometric objects with a complex configuration of the external contour. For its successful solution, the task was divided into three consecutive tasks: task A – Section; task B – Scheme; task C - Interactive adjustment. For their tasks, methods and algorithms for solving them were proposed. The proposed mathematical models and algorithms allowed to develop software for automated design of rational schemes of cutting rectangular materials into flat geometric objects with a complex configuration of the outer contour. This software can be used in various fields, where it is necessary to rationally cut rectangular materials into flat geometric objects and will increase the efficiency of materials in cutting.

References

1. Guo, P., Takahashi, T., Cheng, C., Yoshimura, T.: Floor-planning using a tree representation. IEEE Trans. on Computer Aided Design of Integrated Circuits and Systems, 281, (2001)
2. 2. Chang, Y.C., Chang, Y.W., Wu, G.M., and Wu. S.W., B* - trees: a new representation for non-slicing floor plans, in *Proc. of the DAC*, 458(2000)
3. Sakanushi, K., Kajitani, Y., Mehta, D.P. The quarter-state-sequence floorplan representation. IEEE Trans. on Computer Aided Design of Integrated Circuits and Systems, 376, (2003).
4. Okano, H., A scanline-based algorithm for the 2D free-form bin packing problem, J. of the Oper. Res. Soc. Japan, 45, 145,(2002)
5. Lesh, N., Marks, J., McMahon, A., and Mitzenmacher, M., New heuristic and interactive approaches to 2D rectangular strip packing, ACMJ. of Experimental Algorithmics, 10(1-2), 1, (2005)
6. Imahori, S., Yagiura, M., and Ibaraki, T., Improved local search algorithms for the rectangle packing problem with general spatial costs,Eur. J. of Oper. Res., 167, 48, (2005)
7. 7. Deb, K.: An efficient constraint handling method for genetic algorithms. Computer Methods in Applied Mechanics and Engineering. **186**(2-4), 311-338 (2000)
8. Sherwani, N.: Algorithms for VLSI Physical Design Automation. Third Edition, Kluwer Aca-demic Publisher, USA, 338 (2013)
9. Valeyeva, A., Petunii, A, Fayzrakhmanov, R.: Primeneniye konstruktivnoy metaevristiki "murav'inaya koloniya" k zadache gil'otinnogo pryamougol'nogo raskroya. Vestnik Bashkirskogo universiteta. Razdel: Matematika. Ufa, **12**(3), 12-14, (2007)

10. Zhang, D., Chen, , C., Lin, Y.: An improved heuristic recursine strategy based on genetic algoritm for the strip rectangular packing problem. Acta Automatica Sinica. **33**(9). 911-916 (2007)
11. Informatsionnyye upravlyayushchiye sistemy. Problemy i resheniya. Monografiya. Odessa, 198-210 (2019)
12. Laszlo, M.: Computational Geometry and Computer Graphics in C++. Prentice Hall, New Jersey, 140-148(1997)