МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ В ПРИРОДНИЧИХ НАУКАХ ТА ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

MATHEMATICAL MODELING IN NATURAL SCIENCES AND INFORMATION TECHNOLOGIES

DOI: 10.31319/2519-8106.2(53)2025.341873
UDC 519.254:004.021

Khuda Zhanna, Candidate of Physical and Mathematical Sciences, Assosiate Professor, Department of Mathematical Modeling and System Analysis
Худа Ж.В., кандидат фізико-математичних наук, доцент, кафедра математичного моделювання та системного аналізу
ORCID: 0000-0002-3451-5825
e-mail: khudazhanna@gmail.com

Nechyporenko Anton, master's degree student, Department of Mathematical Modeling and System Analysis
Нечипоренко А.Л., здобувач другого (магістерського) рівня, кафедра математичного моделювання та системного аналізу
e-mail: alnechyporenko@gmail.com

Dniprovsky State Technical University, Kamianske
Дніпровський державний технічний університет, Кам'янське

MODELING OPTIMAL CUTTING OF MATERIALS

МОДЕЛЮВАННЯ ОПТИМАЛЬНОГО РОЗКРОЮ МАТЕРІАЛІВ

The paper presents an effective approach to solving the rather urgent problem of optimizing the cutting of a rectangular sheet of arbitrary shape into rectangular parts of various shapes and sizes with waste minimization. Within the framework of the study, formal statements of the cutting problem were formulated, corresponding mathematical models were built, and a modified genetic algorithm adapted to the specifics of the rectangular object placement problem was implemented. The developed software allows you to set input parameters, visualize and analyze the results obtained.

***Keywords**: material cutting, discrete optimization problems, crossover, genetic algorithms, cutting and packaging problems.*

У контексті зростання вартості матеріалів, комплектуючих та підвищення тарифів на енергоносії, питання раціонального використання сировинних ресурсів набуває особливої актуальності. Це зумовлено необхідністю підвищення ефективності виробничих процесів та зниження витрат на виготовлення продукції. Одним із напрямів такої оптимізації є розв'язання задач плоского розкрою, що безпосередньо впливають на обсяг залишків матеріалу та загальну собівартість продукції.

Задача плоского розкрою формулюється як оптимізаційна проблема розміщення множини менших за розміром елементів на площині, що відповідає заготовці більшого розміру, з метою мінімізації залишків (незаповненого простору). Ефективне вирішення цієї задачі дозволяє знизити *матеріальні втрати та підвищити рентабельність виробництва. Навіть незначне покращення в схемі розміщення деталей може забезпечити суттєвий економічний ефект. Більшість задач даного типу належать до класу NP-складних, що унеможливлює знаходження точного оптимального розв'язку в обмежений проміжок часу для задач великої розмірності. У зв'язку з цим, доцільним є застосування евристичних та метаевристичних методів, зокрема генетичних алгоритмів, що забезпечують знаходження наближених, але практично придатних рішень.*

Метою дослідження є розробка ефективного метаевристичного підходу до розв'язання задачі двовимірного прямокутного розкрою з використанням генетичних алгоритмів. У межах дослідження сформульовано формальні постановки задачі розкрою, побудовано відповідні математичні моделі, а також реалізовано модифікований генетичний алгоритм, адаптований до специфіки задачі розміщення прямокутних об'єктів.

Запропонований алгоритм базується на імітації процесів природного відбору та еволюції — зокрема, на генерації популяції потенційних рішень (особин), механізмах селекції, кросоверу та мутації. Для інтерпретації генетичних даних у вигляді конкретних карт розкрою розроблено спеціалізований декодер, що забезпечує трансформацію генотипу в реальні просторові конфігурації.

На основі розробленого підходу створено програмне забезпечення, яке дозволяє задавати параметри вхідних даних, візуалізувати отримані рішення та здійснювати їх кількісну оцінку. Результати чисельних експериментів підтверджують ефективність запропонованого методу для широкого класу задач плоского розкрою. Отримані розв'язки демонструють високу якість заповнення площі заготовки та дозволяють зменшити витрати матеріалу, що сприяє підвищенню загальної економічної ефективності виробничих процесів.

***Ключові слова**: розкрій матеріалів, задачі дискретної оптимізації, кросовер, генетичні алгоритми, задачі розкрою та упаковки.*

Problem's Formulation

Given the increase in prices for materials and components, as well as the increase in energy tariffs, the issue of rational use of raw materials is gaining importance due to the need to optimize production processes and reduce production costs.

Flat cutting is an optimization problem of finding a dense placement of a set of smaller parts on larger objects. Solving this problem leads to the minimization of cutting losses, that is, the reduction of the volume of unoccupied space. Even small improvements in placement can lead to significant material savings and reducing the cost of production.

Since most of the cutting and packaging problems are classified as problems with NP complexity, finding exact solutions in an acceptable time is impossible. In this case, only the search for local optima is possible using approximate algorithms.

The cutting and packaging problem consists in determining the optimal position of a finite number of geometric objects in given areas, taking into account various constraints. Solving these problems contributes to reducing production costs, increasing efficiency and saving resources. The need is to create universal and effective algorithms that would work for different types of materials and different part geometries. Therefore, the development and implementation of new approaches to solving the problem of optimal cutting are extremely important and relevant.

Analysis of recent research and publications

Cutting and packing problems are important tasks in the field of optimization that arise in many industrial and commercial sectors. These problems involve the efficient use of resources, cost minimization, and productivity improvement in the manufacturing of products or the packaging of goods. They belong to the class of NP-hard problems (Nondeterministic Polynomial time), which means that solving them exactly requires the use of complex algorithms and numerical methods. The main goal of solving such problems is to maximize the use of available space while minimizing costs.

The need for the development of efficient cutting methods was recognized as early as the mid-20th century. A significant contribution to the research of this issue was made by Professor Yu.H. Stoyan [1], who developed effective heuristic algorithms for orthogonal cutting. The solution of various classes of optimization problems has been the subject of work by Ukrainian researchers, including O.O. Yemets, A.I.

Kosolap [2], L.F. Hulianytskyi [3], I.V. Serhiyenko [4], and S.I. Yaremchuk. Considerable attention has also been devoted to cutting and packing problems in foreign publications. For example, Dyckhoff proposed a typology of such problems. One notable example is the two-dimensional strip packing problem, which consists of placing a given set of rectangles onto a semi-infinite strip in such a way as to minimize the length of the used portion of the strip [5]. However, the application area of this problem is much broader. It is used to solve practical problems involving the allocation of two-dimensional resources, such as scheduling, workforce planning, as well as packing and placement problems.

Later, it became clear that the typology was insufficient to include current developments. Therefore, Wäscher, G., Haußner, H., Schumann, H. decided to present a new, improved typology, providing a consistent system of problem types that allows for a complete classification of all known cutting and packaging (C&P) problems. In [6], the authors reviewed the existing literature on uncertainty in cutting and packaging problems, proposed a classification framework, and highlighted numerous research gaps and opportunities for new research.

Most studies focus on approximate methods for solving such problems. Exact methods are rarely used, as they require an exhaustive search of the entire set of feasible solutions to find the optimal one. The efficiency of such exhaustive searches can be improved by enhancing algorithms in various ways [6]. For example, algorithms can be optimized by ordering items in decreasing order of volume (first-fit decreasing algorithm). However, this approach does not guarantee an optimal solution and leads to increased execution time for large input sizes. Improved exhaustive search is most commonly based on the branch and bound method [7].

In article [8], a solution to the problem of splitting a set of ordered elements using minimum-cost objects within a set of minimum-cost solutions is proposed, while maximizing the cost of usable residues. Since the concept of usable residues assumes that they can potentially be used to service new incoming orders, the problem extends to a periodic structure. Thus, the solution at each point in time does not minimize the cost of objects needed to service current orders, but is aimed at minimizing the total cost of objects.

However, existing methods do not always provide solutions close to optimal within an acceptable time frame, which creates the need to develop new methods for formulating and verifying the conditions for solving new classes of problems.

Formulation of the purpose of the research

The purpose of this work is to develop an algorithm and a software module based on a genetic algorithm for organizing the process of cutting a rectangular sheet of arbitrary shape into rectangular parts of various shapes and sizes with waste minimization.

Presenting main material

A rectangular sheet of a given width and length is provided, along with an ordered set of rectangular items. The sheet width, as well as the quantity and dimensions of the items, are specified. The task is to determine the parameters for the optimal placement of the items that will allow all elements from the order list to fit onto the sheet while minimizing the total material usage. Depending on the layout, the rectangular items may be of the same or different sizes.

In the one-dimensional case, packing of rectangle involves placing items of equal width. In this process, only one dimension is important. The use of items with different widths leads to the two-dimensional rectangle packing problem. Additionally, rectangle packing problems can be three-dimensional, for instance, in pallet or container loading tasks.

Many cutting and packing problems include additional constraints. These may be geometric constraints (e.g., the shapes to be cut or packed must conform to certain forms), technical constraints (e.g., the use of specific tools or technologies), or time constraints (e.g., the need to complete the task quickly).

Cutting problems can be divided into cutting problems for measurable material, where items have a predefined length, and cutting problems for unmeasurable material, where the length is random and unknown in advance. Accordingly, these are classified as deterministic and stochastic cutting problems.

The cutting problem can be represented in different ways, depending on the ultimate goal. Let us now consider several specific models.

We introduce the necessary notation:

J — material index, $j = \overline{1,n}$;

k — workpiece type index, $k = \overline{1,q}$;

i — index of the cutting method of a unit of material, $i = \overline{1,p}$;

a_{ijk} — number (integer) of blanks of type k obtained by cutting a unit of j-th material by i-th method;

b_k — number of blanks of type k in the set supplied to the customer;

d_j — amount of material of the j-th type;

x_{ij} — number of units of j-th material cut using i-th method (intensity of use of cutting method);

c_{ji} — the amount of waste obtained when cutting a unit of j-th material using the i-th method;

y — the number of sets of blanks of various types supplied to the customer.

Let's consider different types of cutting problem models.

Cutting model with minimal material consumption:

$$\sum\sum x_{ij} \rightarrow \min ; \tag{1}$$

$$\sum\sum a_{ijk} x_{ij} \geq b_k, \ k = \overline{1,q}; \tag{2}$$

$$x_{ij} \geq 0, \ i = \overline{1,p}; \quad j = \overline{1,n}. \tag{3}$$

Here (1) is the objective function (minimum amount of materials used); (2) — a system of constraints that determine the number of blanks required to fulfil the order; (3) — conditions for non-negativity of variables. Specific to this model are constraints (2).

Cutting model with minimal waste:

$$\sum\sum c_{ij} x_{ij} \rightarrow \min ; \tag{4}$$

$$\sum\sum a_{ijk} x_{ij} = b_k, \ k = \overline{1,q}; \tag{5}$$

$$x_{ij} \geq 0, \ i = \overline{1,p}; \quad j = \overline{1,n}. \tag{6}$$

Here (4) is the objective function (minimum waste when cutting materials); (5) is a system of constraints that determine the number of blanks required to fulfill the order; (6) is the condition for non-negativity of variables.

Cutting model taking into account the configuration:

$$y \rightarrow \max ; \tag{7}$$

$$\sum x_{ij} < d_j ; \tag{8}$$

$$\sum\sum a_{ijk} x_{ij} \geq b_k, \ k = \overline{1,q}; \tag{9}$$

$$x_{ij} \geq 0, \ y \geq 0, \ d_j > 0, \ \ i = \overline{1,p}; \quad j = \overline{1,n}. \tag{10}$$

Here (7) is the objective function (maximum of sets including blanks of different types); (8) is the restriction on the number of materials; (9) is the system of restrictions determining the number of blanks required to form sets; (10) is the condition of non-negativity of variables. Specific to this model are restrictions (9).

Due to the fact that exact methods based on exhaustive search cannot be implemented for large-scale data, more efficient reduced search methods are used instead [9]. The use of effective approximate methods of combinatorial optimization, which are applied in practice, is determined by several factors: practical problems are NP-hard, making exact solutions extremely difficult; the input data often contain inaccuracies; objective functions may have multiple local extrema; and approximate computational schemes allow the construction of algorithms that can solve not just one, but an entire class of closely related optimization problems.

Evolutionary computation methods are effectively used to solve cutting and packing problems [10]. These methods operate on a set of randomly generated solutions P, it is called the population. In evolutionary algorithms, potential solutions are encoded as vectors of elements resembling the genetic structure of chromosomes. New solutions, called offspring, are generated by applying genetic operators to existing solutions. The classical genetic operators used in evolutionary computation are crossover and mutation. Crossover facilitates the exchange of genetic information between chromosomes encoding potential solutions. Two parent chromosomes are selected either randomly or based on a specific rule and a crossover point is chosen at random. Then, the segments of the chromosomes to the right of the crossover point are exchanged between the parents. The mutation operator randomly alters any element of a chromosome, thereby increasing the diversity within the population. Over the past decade and a half, evolutionary computation methods have become some of the most widely used metaheuristic methods, as evidenced by a significant number of academic publications.

One variant of evolutionary computation methods is the genetic algorithm (GA) — an optimization and search technique that combines elements of deterministic optimization and stochastic approaches [11]. Genetic algorithms are often used in combination with analytical methods or other stochastic algorithms to achieve the best results. They belong to the class of adaptive methods, effective for solving problems in search, optimization, and learning.

In this work, an algorithm has been developed that aims to optimize the placement of templates (patterns) within given contours in order to minimize material waste.

Main steps of the algorithm:

1. Initialization of the algorithm. The program starts.
2. Input data acquisition. Data about the array of templates to be placed and the array of contours where they can be placed are loaded. This data is provided through an interactive subsystem.
3. Preliminary analysis of contours and templates. It is determined which templates can be placed in each contour. Contours where no template can be placed are considered waste and are removed from the array of contours.
4. Iteration for each genome. For each possible combination of criteria (genes), an iteration is performed to determine how to place templates within the contours.
5. Placement quality evaluation. For each placement option, the area of waste (contours without templates) is calculated. The top *m* options with the least waste area are selected.
6. Updating the list of placed templates. It is checked which templates have been successfully placed based on the selected optimal options, and a new placement list is generated.
7. Checking placement possibility in new contours**.** If no template can be placed in the newly formed contours, they are moved to waste. Otherwise, the algorithm proceeds to the next step.
8. Removal of unsuitable contours**.** Contours in which no templates can be placed are deleted from the array as waste.
9. Check for remaining templates. If all templates have been placed, the algorithm proceeds to completion. Otherwise, new possible genomes are generated.
10. Genetic operations. Genetic algorithm operators (crossover, mutation, etc.) are used to create new combinations of genes. Genomes that produce negative results are excluded from the next iterations. A new population is formed, and the process repeats from step 4.
11. Output of results. The results include:
    - the sequence of template placement within contours with coordinates;
    - the percentage of generated waste;
    - a list of unplaced templates (if any).
12. Completion of the algorithm. The algorithm finishes after generating and analyzing all possible placements. It demonstrates effective use of genetic algorithm principles for optimization tasks, confirmed by scientific research and testing.

Fig. 1 shows a block diagram of the genetic algorithm.

The main idea of the work is to develop and apply a genetic algorithm to solve the problem of placing rectangles on a two-dimensional rectangular sheet. According to the developed algorithm, an application interface with all the necessary elements was created.

Consider the following problem. Let us consider a rectangular sheet of material of size W×H, on which it is necessary to place a set of rectangular parts with fixed dimensions $\{(w_1,h_1);(w_2,h_2);..(w_n,h_n)\}$. The goal is to cut in such a way that: all parts are placed on the sheet without overlapping, and material waste (unused area) is minimal.

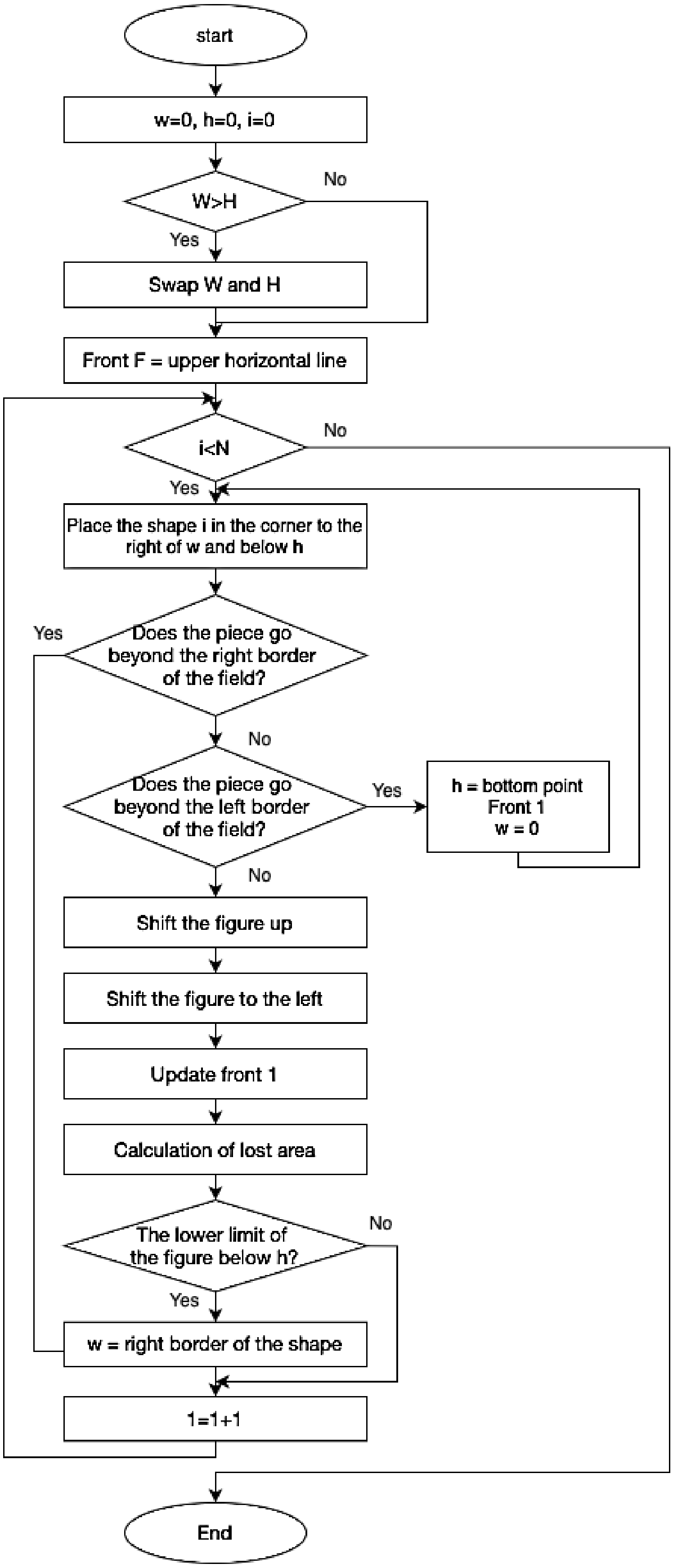

Fig. 1. Block diagram of the genetic algorithm

Let us denote:

S_i — the area of the *i*-th part, this area is equal to $w_i \cdot h_i$;

$S_{det} = \sum_{i=1}^{n} S_i$ — the total area of all placed details;

$S_{sheet} = W \cdot H$ — the area of the entire sheet;

S_{used} — the area occupied by the smallest rectangle (bounding box), which covers all the placed parts;

P — penalty function for overlap (for example, the number of pairs of overlapped parts);

$\alpha \in R_{+}$ — penalty weighting factor.

We use a specially constructed objective function (fitness function) that minimizes waste. Cutting waste is the difference between the area of the sheet (or its used part) and the total area of the placed parts without overlapping. Then the objective function has the form:

$$F = (S_{used} - S_{det}) + \alpha \cdot P \to \min, \qquad (11)$$

where $S_{used} - S_{det}$ is the waste estimate, and the second term is the overlap penalty.

The goal of optimization is to minimize F, that is, to minimize the amount of waste and fines.

It is also necessary to introduce the following constraint conditions:

1. Rectangles should not extend beyond the page boundaries:

$$x_i + w_i' \le W; \quad y_i + h_i' \le H,$$

where $(x_i; y_i)$ — coordinates of the upper left corner of the *i*-th detail, a w_i', h_i' — its dimensions taking into account the rotation.

2. Rectangles should not overlap:

$R_i \bigcap R_j = 0, \quad \forall i \ne j$, where R_i — area occupied by the *i*-th detail.

To solve the problem, a Genetic Algorithm (GA) was used, which simulates natural evolutionary processes. The main stages are:

1. Chromosome Encoding. Each chromosome represents the placement of all parts on a sheet:

- Part index;
- Placement coordinates (x, y);
- Orientation (rotation by 90°, if allowed).

2. Population Initialization. An initial population of randomly generated feasible solutions is created. For each part, random coordinates within the sheet and, if allowed, an orientation are generated. Overlapping of parts is not allowed (a simple greedy algorithm or constraints during generation can be applied).

3. Fitness Evaluation. The fitness function is calculated according to the objective function. It considers the area of placed parts, the bounding box size, and penalties for overlaps.

4. Selection. Parent selection is carried out using tournament selection or roulette wheel selection.

5. Crossover. One-point or position-based crossover is used: part of the part placements is inherited from one parent, the rest from the other, without duplication.

6. Mutation. Mutation operations include: changing a part's position (x, y), changing its orientation (if allowed), or swapping two parts. Some genes in the chromosome are probabilistically altered to maintain genetic diversity.

7. Generation Replacement. A new generation is formed from the best individuals of the previous generation (elitist strategy) and newly created offspring.

8. Stopping Condition. The algorithm terminates after a fixed number of generations or if there is no improvement for N consecutive generations.

The proposed genetic algorithm enables an efficient search for an approximately optimal solution to the rectangular sheet cutting problem while minimizing waste. Its flexibility allows the consideration of additional constraints, such as part rotation or quantity limits, and it can be scaled for larger input data. The algorithm checks for rectangle overlaps. If rotation is allowed, an additional variation (w ↔ h) is considered.

To verify the effectiveness of the proposed algorithm, a series of numerical experiments was conducted with various sets of parts. In all experiments, a rectangular sheet of 100×100 arbitrary units was used. The parts had dimensions ranging from 10×10 to 50×30 units. The total area of all parts varied from 30 % to 80 % of the sheet's area. Rotation of parts by 90 degrees was allowed.

The genetic algorithm parameters were chosen empirically and remained fixed for all test runs:

- Population size: 50 individuals;
- Number of generations: 100;
- Mutation probability: 20 %;
- Number of elite individuals: 5.

As a result of the experiments, it was found that the algorithm reliably placed all given parts without overlaps, with an average sheet utilization rate of over 85 %. In typical experiments, material waste ranged from 15 % to 18 %, depending on the configuration and orientation of the parts.

Here is an example of one of the experiments:

• Number of parts: 6;
• Dimensions of parts: (20×30), (50×20), (10×10), (25×25), (15×35), (30×30);
• Total area of parts: 5250 conventional units;
• Minimum rectangle covering all placements (bounding box): 80×80 = 6400;
• Waste: 6400−5250 = 1150;
• Area utilization factor: $\dfrac{5250}{6400} \cdot 100\% = 82{,}03\%$

It was also observed that using the bounding box as an approximate estimate of the actual occupied area allows for avoiding a rigid dependence on fixed sheet dimensions and provides greater flexibility for practical implementation of the algorithm.

Fig. 2 shows the curve of the dependence of the objective function value on the generation number. The graphs demonstrated stable convergence to a local minimum, typically within 30—50 generations. This indicates the effectiveness of the chosen chromosome encoding model as well as the appropriateness of using elitist strategy and mutation operations.

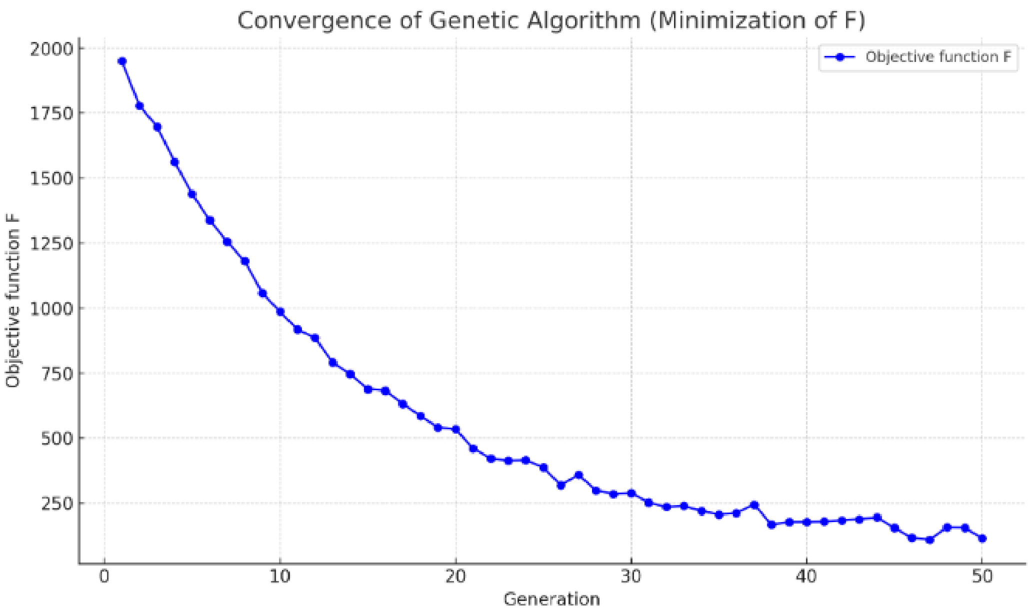

Fig. 2. Convergence of the genetic algorithm

The graph illustrates the convergence of the genetic algorithm: the value of the objective function gradually decreases with each generation, demonstrating the algorithm's successful convergence toward a solution with minimal area waste and minimal constraint violations. A plateau is reached around the 30th—35th generation, indicating the stabilization of the solution.

A comparison of the performance results of the developed algorithm based on the genetic algorithm with the results obtained using the greedy algorithm and exhaustive search is presented in Tabl. 1.

Table 1. Comparison of algorithm results

Method	Area of parts	Bounding box area	Used sheet area	Waste	Utilization rate (%)
Genetic algorithm	9425	10000	10000	575	94.25
Greedy algorithm	8900	10000	10000	1100	89.00
Full search	9425	9600	9600	175	98.18

Thus, it can be seen that the developed algorithm allows approaching the quality of a full enumeration with significantly lower computational load. The greedy method demonstrates the worst utilization ratio due to its local nature. The full enumeration ensures the highest efficiency, but in practice it is too computationally expensive as the number of parts increases.

The main features of the developed application include: support for inputting new material sheets with specification of their quantity and cost per linear meter; the ability to add parts with specified dimensions; generation of an optimized material cutting layout diagram based on a genetic algorithm; calculation of the total area of material sheets, parts, and waste; automatic creation of a report on the completed work on the project.

Conclusions

As the analysis shows, the cutting stock problem has a deep theoretical foundation and numerous practical applications in material-intensive industries. Therefore, there is an urgent need to develop specialized models and methods for solving the cutting problem. One promising direction is the creation of genetic algorithms for solving optimal cutting and packing problems.

Genetic algorithms differ from classical optimization methods in that they do not require the objective function to be smooth or differentiable, making them suitable for solving cutting problems, which are often multi-extremal and lack analytical models. Another advantage of genetic algorithms is their potential for parallel implementation, since each individual in the population can be evaluated independently. This contributes to a significant acceleration of the solution search process.

As a result of the work, a proprietary genetic algorithm was developed and tested, which allows optimizing the placement of rectangular parts of various shapes and sizes on a rectangular sheet of arbitrary shape, which allows minimizing material consumption.

The developed application is therefore useful for solving cutting optimization problems, and the implemented genetic algorithm demonstrates effectiveness. Additionally, software based on this algorithm was developed, enabling the user to input data, visualize cutting results as a layout map, and analyse quality indicators of the obtained solution. The program has shown the ability to handle problems of varying complexity and dimension effectively.

References

[1] Stoian Yu.H., Hil M.I. (2016) Metody i alhorytmy rozmishchennia heometrychnykh obiektiv[Methods and algorithms for placing geometric objects]. Kyiv: Naukova dumka. 247 s. [in Ukrainian].

[2] Kosolap A. (2020) Practical Global optimization. Dnipro: Publisher Bila K.O. 192 p. [in English].

[3] Hulianytskyi L. F., Dubina A. V. (2021) Rozviazuvannia zadachi rozmishchennia priamokutnykiv na napivneskinchennii strichtsi alhorytmamy lokalnoho ta tabuiovanoho poshuku. [Solving the problem of placing rectangles on a semi-infinite ribbon using local and tabu search algorithms] *Naukovyi visnyk Uzhhorodskoho universytetu*. *Seriia «Matematyka i informatyka».* Vol. 38. № 1. P. 123–136. [in Ukrainian].
[4] Sergienko I.V., Shylo V.P. (2016) Modern Approaches to Solving Complex Discrete Optimization Problems. *Automation and Information Sciences*. Volume 48, Issue 1. P. 15-24. [in English].
[5] Liu K., Zhang H., Wang C., Li H., Chen Y., Chen Q. (2023). Robust Optimization for the Two-Dimensional Strip-Packing Problem with Variable-Sized Bins. *Mathematics*, №11(23), 4781. https://doi.org/10.3390/math11234781 [in English].
[6] SalemK.H., Silva E., Oliveira J. F. (2023). Cutting and packing problems under uncertainty: literature review and classification framework. International Transactions in Operational Research. Volume30, Issue6. P. 3329-3360. [in English].
[7] Caggiano A., Seccia A., Di Sciascio F. (2020) A mixed integer linear programming model for optimising stock cutting in steel mills. *International Journal of Production Research*. No. 58 (16). P. 5063–5078. [in English].
[8] Birgin E. G., Romão O. C., Ronconi D. P. (2020) The multiperiod two-dimensional non-guillotine cutting stock problem with usable leftovers. *International Transactions in Operational Research*, Volume27, Issue3. P.1392–1418. [in English].
[9] Delorme M., Iori M., Martello S. (2016) Bin packing and cutting stock problems: Mathematical models and exact algorithms. *European Journal of Operational Research*, Volume 255, Issue 1. P. 1-20. [in English].
[10] Chen M., Wu C., Tang X., Peng Z., Liu S. (2019). An efficient deterministic heuristic algorithm for the rectangular packing problem. *Computers & Industrial Engineering*, 137, 106097. [in English].
[11] Yi-Bo Li, Hong-Bao Sang, Xiang Xiong, Yu-Rou Li (2021). An Improved Adaptive Genetic Algorithm for Two-Dimensional Rectangular Packing Problem. *Applied Sciences* (Multidisciplinary Digital Publishing Institute). Vol. 11, Iss: 1. 413. https://doi.org/10.3390/APP11010413 [in English].

Список використаної літератури

1. Стоян Ю.Г., Гіль М.І. Методи і алгоритми розміщення геометричних об’єктів. Київ: Наукова думка, 2016. 247 с.
2. Kosolap A. Practical Global optimization. Dnipro: Publisher Bila K.O., 2020.192 p.
3. Гуляницький Л. Ф., Дубіна А. В. Розв'язування задачі розміщення прямокутників на напівнескінченній стрічці алгоритмами локального та табуйованого пошуку. *Науковий вісник Ужгородського університету. Серія «Математика і інформатика»,* 38(1), 2021. С. 123–136. https://doi.org/10.24144/2616-7700.2021.38(1).123–136.
4. Sergienko I.V., Shylo V.P. Modern Approaches to Solving Complex Discrete Optimization Problems. *Automation and Information Sciences*. Volume 48, Issue 1, 2016. P. 15–24.
5. Liu K., Zhang H., Wang C., Li H., Chen Y., Chen Q. Robust Optimization for the Two-Dimensional Strip-Packing Problem with Variable-Sized Bins. Mathematics, №11(23), 2023. 4781. https://doi.org/10.3390/math11234781
6. SalemK.H., Silva E., Oliveira J. F. Cutting and packing problems under uncertainty: literature review and classification framework. *International Transactions in Operational Research*. Volume 30, Issue 6, 2023. P. 3329–3360.
7. Caggiano A., Seccia A., Di Sciascio F. A mixed integer linear programming model for optimising stock cutting in steel mills. *International Journal of Production Research*. No. 58 (16), 2020. P. 5063–5078.
8. Birgin E. G., Romão O. C., Ronconi D. P. The multiperiod two-dimensional non-guillotine cutting stock problem with usable leftovers. *International Transactions in Operational Research*, Volume27, Issue3, 2020. P.1392–1418.

9. Delorme M., Iori M., Martello S. Bin packing and cutting stock problems: Mathematical models and exact algorithms. *European Journal of Operational Research*, Volume 255, Issue 1, 2016. P. 1–20.
10. Chen M., Wu C., Tang X., Peng Z., Liu S. An efficient deterministic heuristic algorithm for the rectangular packing problem. *Computers & Industrial Engineering*, 2019. 137, 106097.
11. Yi-Bo Li, Hong-Bao Sang, Xiang Xiong, Yu-Rou Li. An Improved Adaptive Genetic Algorithm for Two-Dimensional Rectangular Packing Problem. *Applied Sciences* (*Multidisciplinary Digital Publishing Institute*). Vol. 11, Iss. 1. (2021). 413. https://doi.org/10.3390/APP11010413

Надійшла до редколегії 03.06.2025
Прийнята після рецензування 09.09.2025
Опублікована 23.10.2025