

<https://doi.org/10.31891/csit-2026-3-9>

Oleksandr TYMCHENKO

Doctor of Technical Sciences, Professor,
Department of Virtual Reality Systems, Lviv
Polytechnic National University, Lviv,
Ukraine

alextymchenko53@gmail.com

<https://orcid.org/0000-0001-6315-9375>

Dmytro PALAMARCHUK

PhD student, Department of Virtual Reality
Systems, Lviv Polytechnic National
University, Lviv, Ukraine

dmytro.y.palamarchuk@lpnu.ua

<https://orcid.org/0009-0000-3934-5899>

Bohdana HAVRYSH

PhD, Docent, Associate Professor,
Department of Information Technologies in
Publishing, Lviv Polytechnic National
University, Lviv, Ukraine

dana.havrysh@gmail.com

<https://orcid.org/0000-0003-3213-9747>

Received: 19/08/2026

Accepted: 24/09/2026

Published: 30/09/2026

© Copyright
2026 by the author(s)



This is an Open Access article distributed
under the terms of the [Creative Commons
CC-BY 4.0](https://creativecommons.org/licenses/by/4.0/)

UDC 004.42:519.6:004.92

**SUMMARY OF THE RESULTS OF
INFORMATION TECHNOLOGY
DEVELOPMENT FOR OPTIMIZING VECTOR
GRAPHIC OBJECT PLACEMENT ON A
PLANE**

The article summarizes the results of developing an information technology for optimizing the placement of vector graphic objects within a limited area for automated cutting, contour cutting, and other manufacturing processes associated with efficient material utilization. The problem of placing arbitrarily shaped objects is formalized with consideration of workspace boundaries, allowable rotation angles, technological clearances, non-overlap constraints, and layout quality criteria. The developed technology combines SVG data preparation, discrete contour approximation, genetic optimization, clustering, caching, and parallel processing of placement variants. To reduce computational complexity, each contour together with its buffer zone is converted into a two-dimensional binary matrix, making it possible to replace complex geometric checks with comparisons of matrix masks using bitwise operations. The optimization core is implemented on the basis of a genetic algorithm in which solutions are represented as chromosomes, while new generations are formed using elitist selection, selection, crossover, mutation, and clustering. The technology was implemented as a desktop software tool developed in Kotlin on the JVM platform using Java Swing, which supports SVG file loading, parameter configuration, optimization execution, visualization, and layout saving. Experimental studies were conducted using synthetic and practically oriented sets of vector objects. The relationship between the discretization cell size, layout accuracy, and computation time was established. In the example involving the placement of paper model parts, the number of required A4 sheets was reduced from 10 to 6, corresponding to a 40% reduction. The obtained results confirm the effectiveness of the technology for automated cutting, contour cutting, and other vector object placement tasks.

Keywords: vector graphic objects, placement optimization, discrete approximation, genetic algorithm, clustering, caching, automated cutting.

Problem statement

The problem of efficiently placing graphic objects within a limited area arises in many manufacturing processes related to cutting, printing, engraving, milling, and contour cutting of parts [1]. Such problems occur in printing, advertising production, the textile and garment industries, packaging manufacturing, metalworking, woodworking, as well as in laser [2], plasma, and plotter cutting systems. In these processes, parts are often represented as two-dimensional vector contours of arbitrary shape that must be placed compactly within the boundaries of the available material.

One example of practical application is the production of stickers, labels, stencils, and decorative elements using desktop cutting plotters. Figure 1(a) shows the contour-cutting process [3], while Figure 1(b) shows the material waste remaining after the vector parts have been cut [4].

Insufficiently compact placement results in unoccupied areas between parts, which become material waste after processing is completed. This problem is characteristic not only of plotter cutting, but also of cutting fabric, paper, polymers, sheet metal, wood, and other materials. Inefficient arrangement increases raw material consumption, the amount of waste, and production costs [5].

Therefore, the development of an information technology for the automated search for an efficient placement of arbitrarily shaped vector graphic objects is a relevant problem.

Such a technology should ensure sufficient accuracy in contour representation, rapid verification of the admissibility of the relative positions of objects, consideration of technological constraints, and acceptable computation time. Its application should contribute to reducing the required material area and improving the efficiency of automated cutting and manufacturing processes. Solving this problem requires the integration of a formal problem model, methods for rapidly verifying geometric constraints, an optimization algorithm, and software tools for generating the final solution [6].

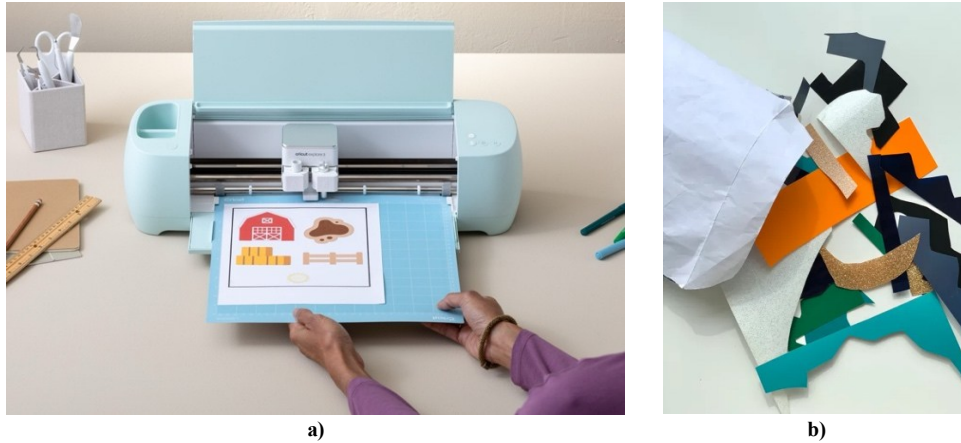


Fig. 1. Example of the application of contour-cutting technologies: a) cutting stickers using a desktop cutting plotter; b) material waste remaining after cutting vector parts

Analysis of recent research and publications

A considerable number of methods have been proposed for solving two-dimensional packing and layout problems. These methods differ in their solution-generation principles, computational complexity, and the quality of the results obtained. The simplest approaches include the Bottom-Left, Best-Fit, Shelf, and Guillotine heuristic algorithms, which place objects sequentially according to predefined rules. Their advantages include ease of implementation and relatively low computational costs. However, the result strongly depends on the initial order of the objects, and the resulting layout often does not provide sufficiently efficient use of the available area [6, 7].

A separate group comprises geometric methods and mathematical programming methods that formalize the relative positions of objects, non-overlap conditions, and workspace constraints. Such approaches can provide high accuracy and near-optimal solutions for small-scale problems. However, for arbitrarily shaped vector objects, intersection checking and the generation of feasible positions require complex geometric operations, while computational costs increase rapidly as the number of parts and possible orientations grows [5, 8].

Metaheuristic methods are widely used for complex combinatorial problems, including genetic algorithms, simulated annealing, tabu search, swarm-based methods, and other evolutionary approaches [6, 9]. They do not require exhaustive exploration of the entire solution space and make it possible to search efficiently for large sets of objects. However, the effectiveness of such methods depends on the solution representation, the evaluation function, the choice of search operators, and parameter settings. The use of a complex geometric model at each iteration can also significantly reduce their computational performance.

A separate group comprises geometric methods and mathematical programming methods that formalize the relative positions of objects, non-overlap conditions, and workspace constraints. Such approaches can provide high accuracy and near-optimal solutions for small-scale problems. However, for arbitrarily shaped vector objects, intersection checking and the generation of feasible positions require complex geometric operations, while computational costs increase rapidly as the number of parts and possible orientations grows [5, 8].

Metaheuristic methods are widely used for complex combinatorial problems, including genetic algorithms, simulated annealing, tabu search, swarm-based methods, and other evolutionary approaches [6, 9]. They do not require exhaustive exploration of the entire solution space and make it possible to search efficiently for large sets of objects. However, the effectiveness of such methods depends on the solution representation, the evaluation function, the choice of search operators, and parameter settings. The use of a complex geometric model at each iteration can also significantly reduce their computational performance.

A promising direction is the use of hybrid approaches that combine heuristic rules, metaheuristic optimization, discrete contour representation, clustering, and parallel computing. Preliminary grouping of objects with similar dimensions or shapes can reduce the dimensionality of the search space, while discretization can simplify the verification of placement feasibility [10, 11]. Similar principles are also used in industrial CAD/CAM systems for automated cutting; however, the algorithmic mechanisms of such systems are generally proprietary, and their adaptability to specific technological constraints remains limited. A review of existing approaches shows that scalability, high computational costs, and the complexity of practical integration remain the main challenges.

The conducted analysis showed that existing approaches do not simultaneously provide high computational performance, layout quality, scalability, and consideration of the technological constraints of automated cutting and contour-cutting processes. Individual aspects of this problem, including the application of evolutionary methods, discrete contour approximation, and clustering of vector objects, were considered in the author's previous publications [12, 14]. This article summarizes the main results of the dissertation research on the development of an information technology for optimizing the placement of vector graphic objects [13].

Objective of the study

The objective of this article is to summarize the results of the development and experimental evaluation of an information technology for optimizing the placement of vector graphic objects.

Presentation of the main research material

The developed information technology is intended for the automated placement of a set of arbitrarily shaped vector graphic objects within a specified workspace. The position of each object is determined by its coordinates and an allowable rotation angle. The resulting layout must ensure that all contours remain within the boundaries of the workspace, that objects do not overlap, and that the specified technological clearances are maintained. The set of configurations satisfying these constraints forms the feasible solution space [15].

Formally, the feasible solution space is defined as:

$$D = (x_i, y_i, \theta_i)_{i=1}^n \mid C_i(x_i, y_i, \theta_i) \subseteq \text{Container}, \text{dist}(C_i, C_j) \geq \delta, \theta_i \in \Theta_i, \quad (1)$$

where (x_i, y_i) - re the coordinates of object o_i on the plane; θ_i - is its rotation angle; Θ_i - is the set of allowable rotation angles for object o_i ; $C_i(x_i, y_i, \theta_i)$ - is the contour of the object after the corresponding geometric transformations have been applied; $\text{dist}(C_i, C_j)$ - is the distance between the contours of objects o_i and o_j ; δ - is the minimum allowable clearance between objects; *Container* - is the bounding rectangular area within which the objects are placed.

To evaluate placement efficiency, criteria characterizing layout density and material consumption were used. The main indicator is the area utilization ratio, defined as the ratio of the total area of the placed objects to the area occupied by them. For roll materials, the height of the actually used portion of the workspace is additionally taken into account. Thus, the problem involves maximizing the area utilization ratio while simultaneously minimizing the used area or the length of material.

The area utilization ratio is defined as:

$$\eta = \frac{\sum_{i=1}^n S_i}{S_{\text{container}}}, \quad (2)$$

where $O = \{o_1, o_2, \dots, o_n\}$ - is the set of objects, each having an area S_i . Maximizing of η is one of the key optimization criteria.

Formally, the problem can be represented as the following system:

$$\max \eta = \frac{\sum_{i=1}^n S_i}{S_{\text{container}}}, \quad (3)$$

$$\min H_{\text{used}} = \max_i (y_i^{\max}),$$

where η - is the container area utilization ratio; S_i - is the area of the i -th object; $S_{\text{container}}$ - is the container area; H_{used} - is the height of the area actually occupied after placement; $y_i^{\max}$ - is the maximum coordinate of the uppermost point of object o_i ; n - is the number of objects to be placed.

In the software implementation, the generated placement variants are evaluated using the ratio of the total area of the graphic objects to the area of the minimum bounding rectangle that encloses them. An increase in this indicator corresponds to a more compact arrangement and a smaller amount of unused material. It is subsequently used as the fitness function during the evolutionary search for the optimal solution [15, 17].

The proposed information technology comprises a sequence of stages for the preparation, processing, and optimization of vector graphic data. SVG files are supplied as input to the system, from which the object contours are imported and normalized. The parameters of the workspace, allowable orientations, technological clearances, and optimization search are also specified. The contours are then discretized to form matrix masks suitable for rapidly verifying the feasibility of the relative placement of objects.

The optimization core implements a genetic algorithm in which generations of possible layouts are created and successively improved. For each variant, geometric constraints are checked, quality criteria are calculated, and the best solutions are selected using selection, crossover, mutation, and clustering operators. The final stage involves selecting the best configuration, restoring its vector representation, visualizing it, and generating the final layout for subsequent use in the manufacturing process [12, 16]. The relationship between the main components of the technology is shown in Fig. 2.

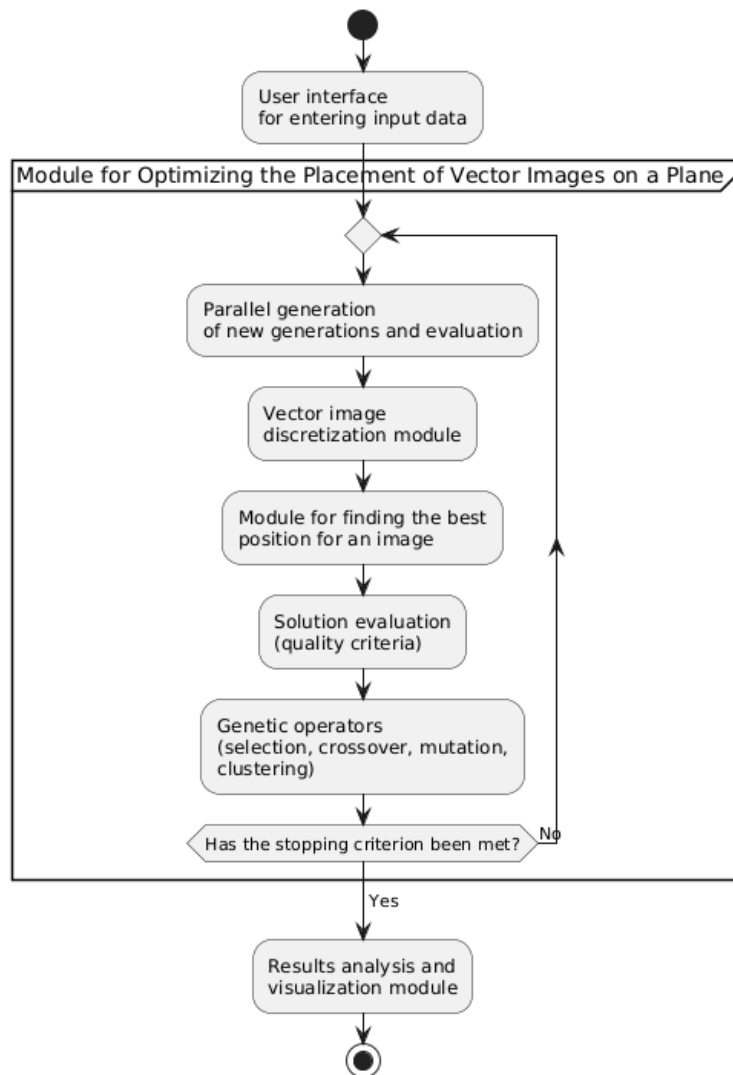


Fig. 2. Structural and functional diagram of the information technology for optimizing the placement of graphic objects

Unlike approaches in which geometric verification, optimization search, and layout generation are considered separately, the proposed technology integrates these processes into a unified sequence of vector data processing. Its main components include discrete contour representation, genetic search, methods for reducing repeated computations, and software-based generation of the final SVG layout.

To reduce the computational complexity of geometric verification, the developed technology employs discrete approximation of vector objects. Each contour, together with a specified buffer zone, is converted into a two-dimensional binary matrix in which occupied cells correspond to regions intersected by the object. As a result, placement feasibility verification is reduced to comparing matrix masks instead of repeatedly performing complex operations on vector contours [16]. This transformation process is illustrated in Fig. 3.

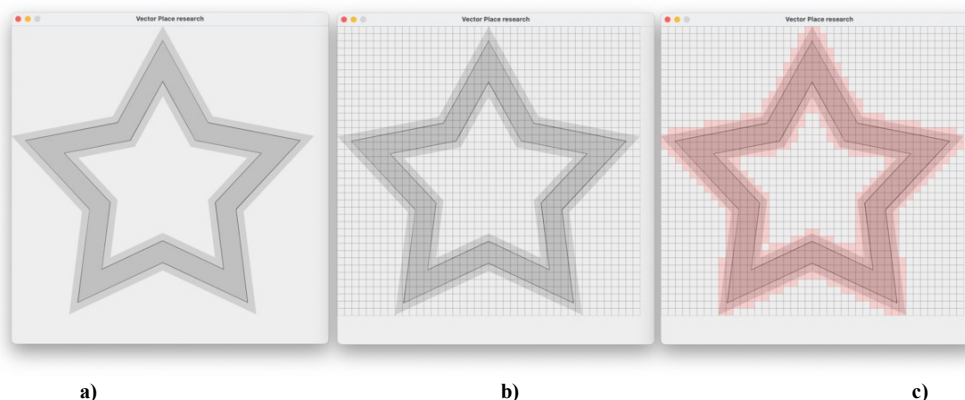


Fig. 3. Discretization of a vector image: a) original vector image; b) discretized image with an overlaid grid; c) discretization result with occupied cells indicated

The cell size determines the trade-off between contour representation accuracy and computational performance: a finer grid reproduces the object geometry more accurately but requires more memory and a greater number of operations, whereas increasing the cell size accelerates verification at the cost of a certain reduction in accuracy. To further improve performance, the matrices are stored in a compact bit-based representation, and intersection checks are performed using bitwise operations. Reusable discretization results for individual objects and their rotation angles are cached, eliminating the need to regenerate identical masks during genetic optimization.

A genetic algorithm is used to search for an efficient placement. In this algorithm, each possible solution is represented as a chromosome—an ordered sequence of genes describing the graphic objects, their positions, and allowable rotation angles. The initial population is formed from different layout variants, after which the fitness value of each chromosome is calculated to characterize the efficiency of workspace utilization [12].

The next generation is formed from several sources, as shown in Fig. 4. Some of the best chromosomes are transferred unchanged according to the principle of elitist selection. Other solutions are generated through the selection of parent chromosomes, crossover, and subsequent mutation. This approach simultaneously preserves the best configurations found and explores new regions of the search space.

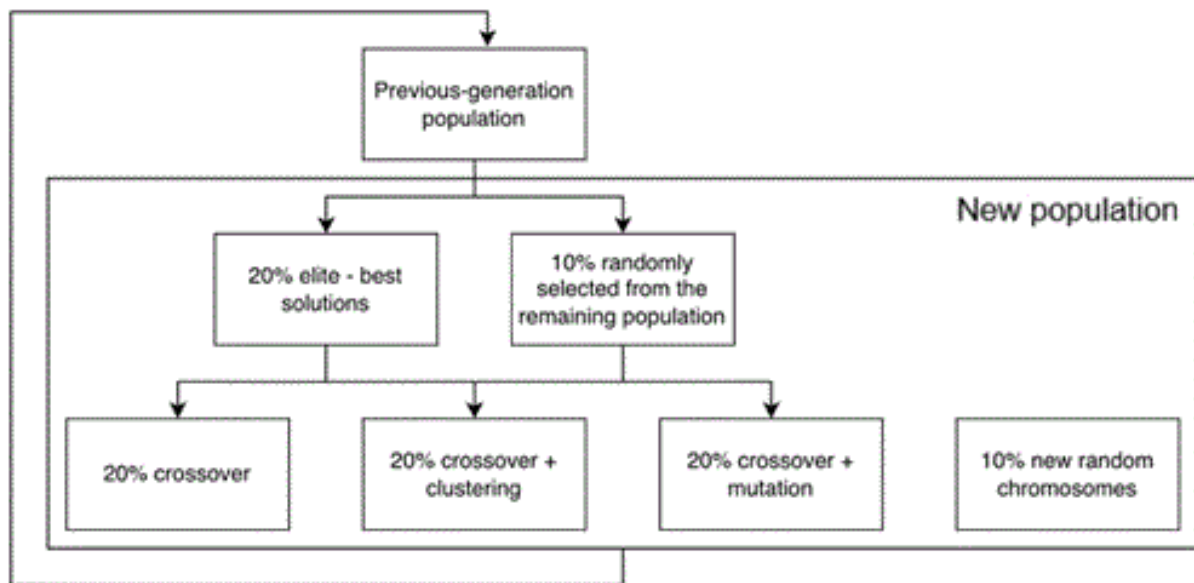


Fig. 4. Formation of a new population in the genetic algorithm for optimal placement of vector objects

During the evaluation of each chromosome, feasible object positions are determined using discretized matrix masks, thereby combining evolutionary search with rapid verification of geometric constraints [12, 17].

Repeated evaluation, selection, and generation of new chromosomes ensure a gradual improvement in layout quality. The computation is terminated when the specified number of generations is reached, the required quality level is achieved, or no further improvement in the result is observed.

To improve the quality and computational performance of the genetic search, the algorithm was enhanced through clustering vector objects according to their geometric characteristics, caching discretized representations for different rotation angles and intermediate placement verification results, and parallel processing of chromosomes during their generation and evaluation. The combined use of these techniques reduces the number of repeated computations, improves layout stability, accelerates the search, and enables more efficient processing of large sets of graphic objects [13, 17].

For the practical implementation of the developed information technology, a desktop software application was created. It supports SVG file loading, problem parameter configuration, execution of genetic optimization, visualization of the resulting layout, and saving of the final layout. The software was implemented in Kotlin on the JVM platform using Java Swing to build the graphical user interface.

The user can specify the dimensions of the workspace, discretization parameters, the number of chromosomes and generations, allowable rotation angles, and technological clearances. After the computations are completed, the program displays the generated layout together with its quality indicators and allows the result to be saved in SVG format. The main stages of working with the application are shown in Fig. 5: the input of initial data is presented in Fig. 5(a), while the visualization of the generated layout is shown in Fig. 5(b).

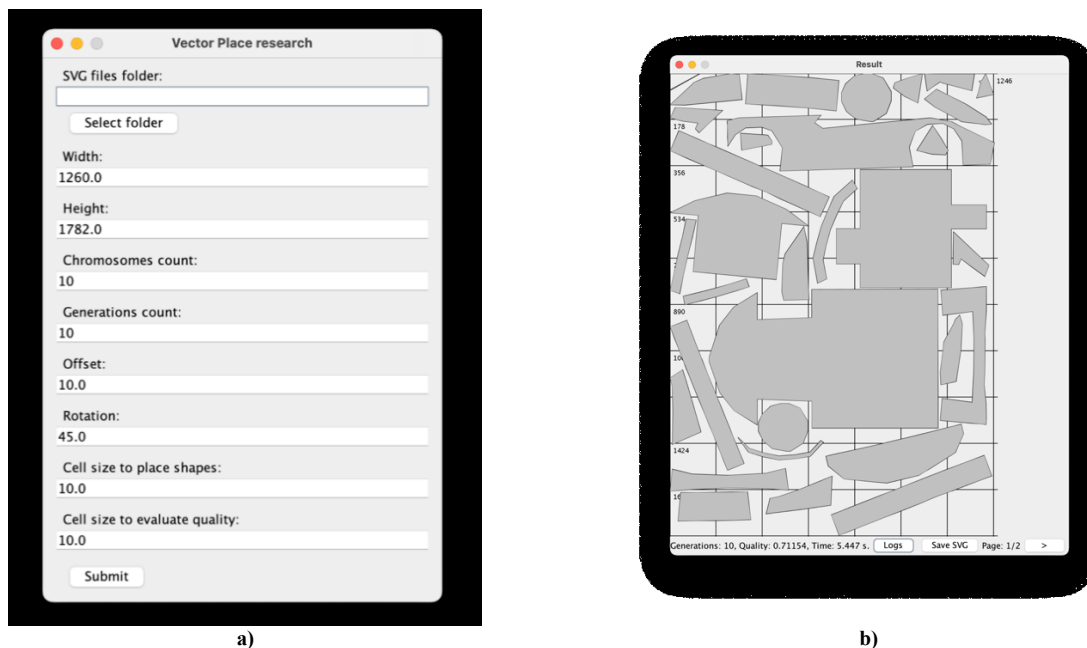


Fig. 5. Interface of the software tool for optimizing the placement of vector objects:
a) input of initial data and optimization parameters; b) visualization of the generated layout

Experimental studies

To verify the operability and effectiveness of the developed information technology, a series of experimental studies was conducted using synthetic and practically oriented sets of vector objects. The experiments examined the influence of the discretization cell size, allowable rotation step, number of objects, genetic algorithm parameters, and the use of clustering on layout quality and computation time. The results were evaluated using the workspace utilization ratio, algorithm execution time, and the amount of material required to place all parts. The following presents representative results of the study of discretization parameters using synthetic data, as well as an example of applying the technology to a real set of vector parts.

To evaluate the influence of discretization parameters, a series of experiments was conducted using a synthetic dataset of 108 vector objects composed of letters and digits of different sizes. The objects were placed within a workspace of 500×500 arbitrary units using fixed genetic algorithm parameters: a population of 50 chromosomes, 10 generations, a rotation step of 45° , and a minimum distance of 5 arbitrary units between objects. The discretization cell size was successively varied from 1×1 to 10×10 arbitrary units, which made it possible to determine its effect on layout quality and computation time.

The results showed that increasing the cell size substantially reduces the algorithm execution time; however, it simultaneously decreases contour approximation accuracy and reduces layout density. With a cell size of 1×1 , the area utilization ratio was 0.7220 and the computation time was 1543.1 s; for a cell size of 10×10 , the corresponding values were 0.5600 and 12.2 s. The relationships between computation time, area utilization ratio, and cell size are shown in Figs. 6 and 7.

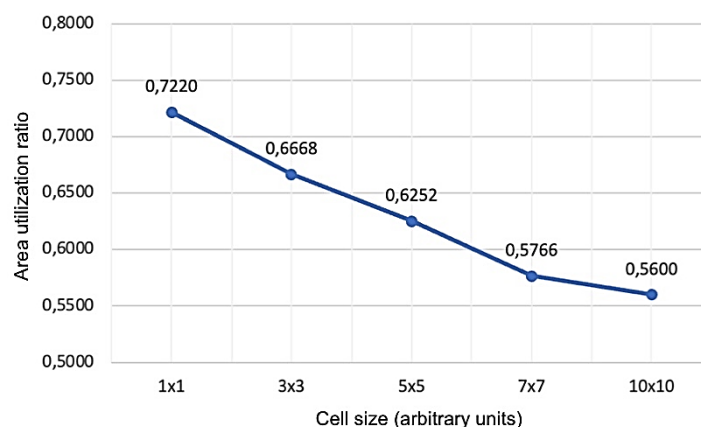


Fig. 6. Relationship between the area utilization ratio and cell size

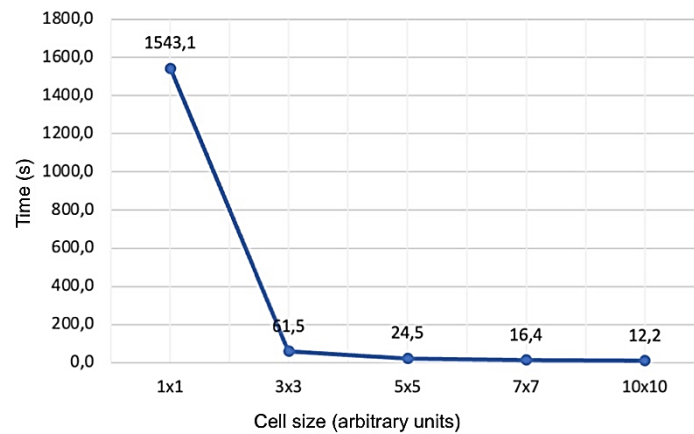


Fig. 7. Relationship between execution time and cell size

The visual comparison presented in Fig. 8 confirms the obtained numerical results: a finer grid makes it possible to represent contour shapes more accurately and produce a denser layout, whereas increasing the cell size creates additional unoccupied areas between objects. At the same time, an excessive reduction in cell size significantly increases computational costs; therefore, its value should be selected with consideration of the required trade-off between accuracy and computational performance.

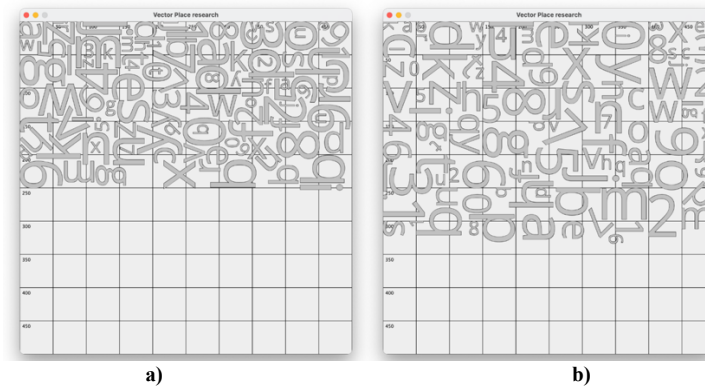


Fig. 8. Visualization of object placement for different discretization cell sizes: a) 1×1 arbitrary units (area utilization ratio 0.7220); b) 10×10 arbitrary units (area utilization ratio 0.5600)

Therefore, the cell size can be used as a controllable technology parameter depending on the priority of a particular task—layout accuracy or computational performance.

The effectiveness of the developed technology was also evaluated using practically oriented vector datasets generated from unfolded patterns of three-dimensional paper models. For datasets containing a large number of parts placed on A4 sheets, a coarser discretization of 50×50 arbitrary units and a minimum clearance of 50 arbitrary units were used. This choice of parameters reduced the computational load while maintaining sufficient accuracy for practical application.

As an example, a set of 66 parts forming a helmet model was used, the general appearance of which is shown in Fig. 9. The initial layout required 10 A4 sheets, whereas after optimization the number of sheets was reduced to 6, corresponding to a 40% reduction. The area utilization ratio was 0.7533, and the execution time was 2.569 s. A comparison of the initial and optimized layouts is presented in Fig. 10.

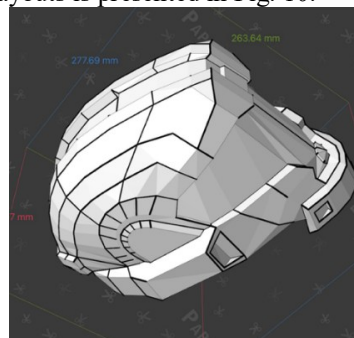


Fig. 9. Visualization of the 3D helmet model.

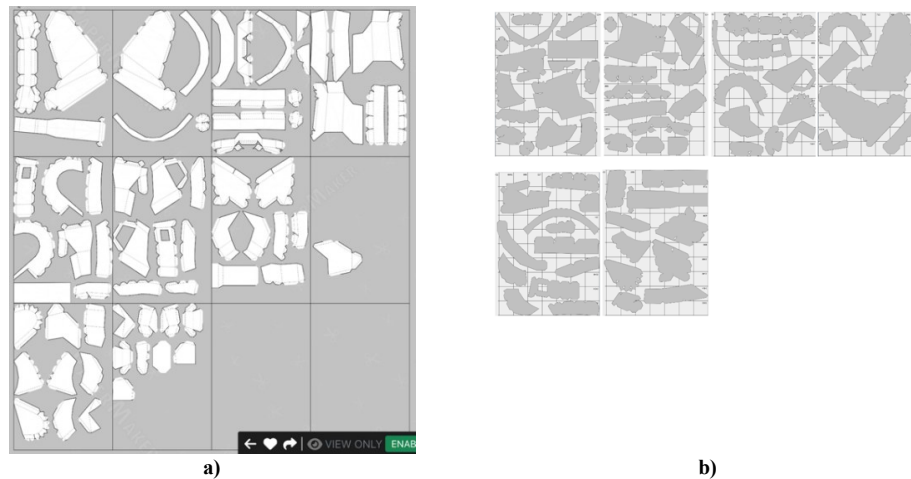


Fig. 10. Placement of parts on sheets: a) initial layout before optimization (10 sheets); b) optimized layout after execution of the algorithm (6 sheets).

The conducted studies confirmed the operability of the technology using both synthetic and practically oriented data. A controllable trade-off between layout accuracy and computational performance, depending on the discretization cell size, was established. For the real set of parts, the required number of A4 sheets was reduced from 10 to 6.

Conclusions

The article summarizes the main results of developing an information technology for optimizing the placement of vector graphic objects. An information technology was developed in which discrete contour approximation and genetic search are combined with clustering, caching, and parallel processing of placement variants. This made it possible to reduce the computational costs of intersection checking and to search for efficient configurations for sets of arbitrarily shaped objects.

The technology was implemented as a software tool for loading SVG data, configuring problem parameters, running the optimization process, visualizing the results, and saving the generated layouts. Experimental studies using synthetic data confirmed the relationship between the discretization cell size, layout accuracy, and computation time, making it possible to select parameters according to the requirements of a particular problem.

Evaluation using practically oriented datasets demonstrated the possibility of substantially reducing material consumption. In particular, for the set of helmet model parts, the number of required sheets was reduced by 40%, from 10 to 6. The obtained results confirm the effectiveness of the developed information technology and the feasibility of its application in automated cutting, contour cutting, and other processes involving the placement of vector objects within a limited area. Further research should focus on the automated selection of discretization and genetic algorithm parameters according to the geometric characteristics of the input dataset.

ADDITIONAL INFORMATION

AUTHOR CONTRIBUTIONS

Conceptualization, O.T.; methodology, D.P.; validation, B.H.; formal analysis, D.P.; investigation, D.P.; data curation, D.P.; writing-original draft preparation, B.H.; writing-review and editing, B.H.; visualization, D.P.; supervision, O.T.; project administration, O.T. All authors have read and agreed to the published version of the manuscript.

DECLARATION ON THE USE OF GENERATIVE ARTIFICIAL INTELLIGENCE TOOLS

In preparing this work, the author used DeepL Translate and Grammarly for: grammar and spelling checks, paraphrasing, and rephrasing. After using these tools/services, the author reviewed and edited the content and takes full responsibility for the content of this publication.

REFERENCES

1. Kokhanovskyi V. O. Publishing and Printing Equipment 2. Post-Press and Finishing Equipment. Kyiv: Igor Sikorsky Kyiv Polytechnic Institute, 2019. 37 p. URL: <https://ela.kpi.ua/handle/123456789/67067>
2. Peretiatko O. A. Desktop Gantry-Type Laser Machine: Bachelor's Diploma Project. Kyiv, 2021. 68 p. URL: <https://ela.kpi.ua/handle/123456789/53990>
3. Cricut Print Then Cut 101 [Electronic resource]. URL: <https://cricut.com/blog/gb/cricut-print-then-cut-101/>

4. Five Ways to Reduce Waste When Using Your Cricut [Electronic resource]. URL: <https://thelittlepomegranate.co.uk/2022/08/01/five-ways-to-reduce-waste-when-using-your-cricut/>
5. Havrysh B., Palamarchuk D., Tymchenko O., Kovalskyi B., Khamula O. Efficiency of Optimization Algorithms in the Problem of Graphic Objects Placement. CEUR Workshop Proceedings. Vol. 3899, 2025. Proceedings of the 1st International Workshop on Advanced Applied Information Technologies, Khmelnytskyi, Ukraine–Žilina, Slovakia, December 5, 2024, pp. 45–54. URL: <https://ceur-ws.org/Vol-3899/paper5.pdf>
6. Palamarchuk D. Yu., Tymchenko O. V. Modified Genetic Algorithm for Optimal Placement of Vector Graphic Objects. Scientific Notes of the Ukrainian Academy of Printing. No. 1 (70), 2025, pp. 11–20. URL: <https://www.doi.org/10.32403/1998-6912-2025-1-70-11-20>
7. S. Dou et al., "Hierarchically Recognizing Vector Graphics and A New Chart-Based Vector Graphics Dataset," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 46, no. 12, pp. 7556–7573, Dec. 2024, <https://doi.org/10.1109/TPAMI.2024.3394298>
8. J. Jiang, "Research and Application of Object Recognition Algorithm in Architectural Engineering Drawing Based on Deep Learning," 2023 Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC), Dalian, China, 2023, pp. 121–124, <https://doi.org/10.1109/IPEC57296.2023.00029>
9. C. Maxey, J. Choi, Y. Lee, H. Lee, D. Manocha and H. Kwon, "TK-Planes: Tiered K-Planes with High Dimensional Feature Vectors for Dynamic UAV-based Scenes," 2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hangzhou, China, 2025, pp. 3503–3510, <https://doi.org/10.1109/IROS60139.2025.11246343>
10. Palamarchuk D. Yu., Tymchenko O. V. Discrete Approximation in the Problem of Placing Vector Graphic Objects on a Plane. Computer-Aided Design Systems. Theory and Practice. Vol. 7, No. 1, 2025, pp. 221–228. URL: <https://doi.org/10.23939/cds2025.01.221>
11. E. J. Bayro-Corrochano, G. Altamirano-Escobedo, A. Ortiz-Gonzalez, V. Farias-Moreno and N. Chel-Puc, "Computing in the Conformal Space Objects, Incidence Relations, and Geometric Constrains for Applications in AI, GIS, Graphics, Robotics, and Human-Machine Interaction," in IEEE Access, vol. 10, pp. 112742–112756, 2022, <https://doi.org/10.1109/ACCESS.2022.3216266>
12. Palamarchuk D. Yu., Tymchenko O. V., Demchenko V. O. Application of Optimization Methods in the Problem of Placing Vector Graphic Objects on a Plane. Scientific Notes of the Ukrainian Academy of Printing. No. 1 (68), 2024, pp. 61–70. URL: <https://www.doi.org/10.32403/1998-6912-2024-1-68-61-70>
13. Z. Yanting, "Bitmap Vectorization Conversion Method in Plane Design," 2022 14th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA), Changsha, China, 2022, pp. 577–580, <https://doi.org/10.1109/ICMTMA54903.2022.00121>
14. S. Xu, R. Wang, H. Wang and R. Yang, "Plane Segmentation Based on the Optimal-Vector-Field in LiDAR Point Clouds," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 43, no. 11, pp. 3991–4007, 1 Nov. 2021, <https://doi.org/10.1109/TPAMI.2020.2994935>
15. Nikulina O. M., Severyn V. P., Kotsiuba N. V. Development of information technology for optimizing the control of complex dynamic systems. Bulletin of National Technical University "KhPI". Series: System Analysis, Control and Information Technologies. 2020. No. 2 (4). P. 63–69. <https://doi.org/10.20998/2079-0023.2020.02.11>
16. Isenberg T., Carpendale M. S. T., Sousa M. C. Breaking the pixel barrier. Proceedings of the First Eurographics Workshop on Computational Aesthetics in Graphics, Visualization and Imaging. Eurographics Workshop Series. Goslar: Eurographics Association, 2005. P. 41–48. <https://doi.org/10.2312/COMPAESTH/COMPAESTH05/041-048>
17. Riznyk V. Optimum vector information technologies based on the multidimensional combinatorial configurations. International Journal of Computational and Applied Mathematics & Computer Science. 2023. Vol. 3. P. 104–112. <https://doi.org/10.37394/232028.2023.3.12>

Олександр ТИМЧЕНКО, Дмитро ПАЛАМАРЧУК, Богдана ГАВРИШ
Національний Університет «Львівська Політехніка»

УЗАГАЛЬНЕННЯ РЕЗУЛЬТАТІВ СТВОРЕННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ОПТИМІЗАЦІЇ РОЗМІЩЕННЯ ВЕКТОРНИХ ГРАФІЧНИХ ОБ'ЄКТІВ НА ПЛОЩИНІ

У статті узагальнено результати розроблення інформаційної технології оптимізації розміщення векторних графічних об'єктів на обмеженій площині для задач автоматизованого розкрою, контурного різання та інших виробничих процесів, пов'язаних із раціональним використанням матеріалу. Формалізовано задачу розміщення об'єктів довільної форми з урахуванням меж робочої області, допустимих кутів повороту, технологічних відступів, умов відсутності перетинів і критеріїв якості компонування. Розроблена технологія поєднує підготовку SVG-даних, дискретну апроксимацію контурів, генетичну оптимізацію, кластеризацію, кешування та паралельне опрацювання варіантів розміщення. Для зменшення обчислювальної складності кожний контур разом із буферною зоною перетворюється на двовимірну бінарну матрицю, що дає змогу замінити

складні геометричні перевірки порівнянням матричних масок із використанням побітових операцій. Оптимізаційне ядро реалізовано на основі генетичного алгоритму, у якому рішення подаються у вигляді хромосом, а нові покоління формуються із застосуванням елітарного відбору, селекції, кросоверу, мутації та кластеризації. Практичну реалізацію виконано у вигляді настільного програмного засобу мовою Kotlin на платформі JVM із використанням Java Swing, що забезпечує завантаження SVG-файлів, налаштування параметрів, запуск оптимізації, візуалізацію та збереження макета. Експериментальні дослідження проведено на синтетичних і практично орієнтованих наборах векторних об'єктів. Встановлено залежність між розміром клітинки дискретизації, точністю компонування та тривалістю обчислень. На прикладі розміщення деталей паперової моделі кількість необхідних аркушів A4 зменшено з 10 до 6, тобто на 40 %. Отримані результати підтверджують ефективність технології для автоматизованого розкрою, контурного різання та інших задач розміщення векторних об'єктів.

Ключові слова: векторні графічні об'єкти, оптимізація розміщення, дискретна апроксимація, генетичний алгоритм, кластеризація, кешування, автоматизований розкрій.