Applied Computer Science, vol. 19, no. 4, pp. 1–13
doi: 10.35784/acs-2023-31

Submitted: 2023-09-11 | Revised: 2023-09-24 | Accepted: 2023-10-01

Keywords: Levenshtein measure, apictorial, assembly, cut-off, fuzzy-logic

Stanisław SKULIMOWSKI [0000-0002-9049-9516]*,
Jerzy MONTUSIEWICZ [0000-0002-8571-3354]**,
Marcin BADUROWICZ [0000-0003-2249-4219]***

ENHANCING THE EFFICIENCY OF THE LEVENSHTEIN DISTANCE-BASED HEURISTIC METHOD OF ARRANGING 2D APICTORIAL ELEMENTS FOR INDUSTRIAL APPLICATIONS

Abstract

The article addresses the challenge of reconstructing 2D broken pictorial objects by automating the search for matching elements, which is particularly relevant in fields like archaeology and forensic science. The authors propose a method to match such elements and streamline the search process by detecting and filtering out low quality matches.

The study delves into optimizing the search process in terms of duration and assembly quality. It examines factors like comparison window length, Levenshtein measure margin, and number of variants to check, using theoretical calculations and experiments on synthetic elements. The experimental results demonstrate enhanced method effectiveness, yielding more useful solutions and significantly reducing the complexity of element comparisons by up to 100 times in extreme cases.

1. INTRODUCTION

Methods of reconstructing objects by searching for ways to connect their fragments are used, among others, to reconstruct artifacts of the past, such as broken ceramics, textile materials. Furthermore, they are used in the process of restoring destroyed or damaged elements of infrastructure, for example construction or water supply. The process of finding solutions can be tedious and is subject to the possibility of error, for example by oversight. The risk of error increases the more fragments have to be checked and combined. Therefore, any method of even partial automation significantly improves the overall results.

* Lublin University of Technology, Faculty of Electrical Engineering and Computer Science, Department of Computer Science, Poland, s.skulimowski@pollub.pl
** Lublin University of Technology, Faculty of Electrical Engineering and Computer Science, Department of Computer Science, Poland, j.montusiewicz@pollub.pl
*** Lublin University of Technology, Faculty of Electrical Engineering and Computer Science, Department of Computer Science, Poland, m.badurowicz@pollub.pl

There are many methods for classifying and reconstructing broken 2D and 3D objects (Andreadis et al., 2015; Rasheed & Nordin, 2015a, 2015b). Description methods most often refer to the color, texture and shape of fragments. These methods can be defined as a form of dimensions reduction. To reduce complexity, 3D objects are often projected and only the features of the selected cross-section (Chang & Chow, 1973) or depth buffer (Papaioannou et al., 2001) are examined. In the case of 2D objects, edges can be saved as a one-dimensional string containing data, for example, of the edge color (Oxholm & Nishino, 2013).

In terms of limiting number of possible connections to check, search methods used for fragments reassembly can be exhaustive (Brown, 2008) or hierarchical (Vendrell-Vidal & Sánchez-Belenguer, 2014). The process of testing and assembling of elements can be carried out with polynomial functions (Rasheed & Nordin, 2014), statistical methods, fuzzy logic, or even by an artificial neural network (Rasheed & Nordin, 2020).

The developed method of supporting the process of assembling apictorial 2D elements represent a group of polygon packing puzzle methods. In this method the contours of individual elements are represented in the form of a chain codes consisting of vectors of the same length and directions consistent with the assumed rose of directions with an even number of arms. The available directions are described by the letters of the alphabet, which causes the obtained string codes to form abstract words (Fig. 1).

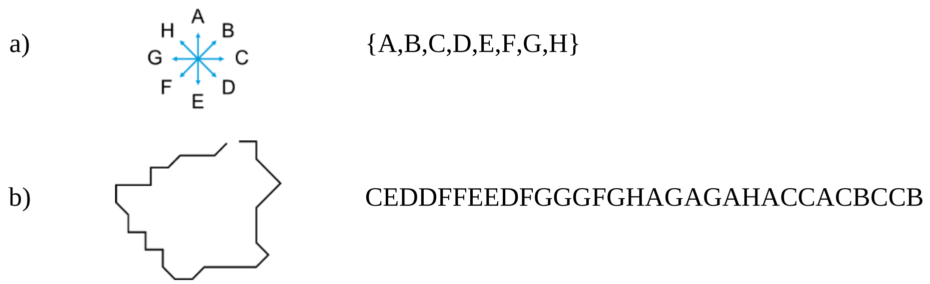

Fig. 1. The method of writing the contour of an element: a) an eight-way star and signs representing available directions, b) an example element and their chain code describing its contour.

The occurrence of code fragments in two compared elements that match after certain transformations is checked using the Levenshtein measure (Montusiewicz & Skulimowski, 2020; Skulimowski & Montusiewicz, 2020). What distinguishes the developed method from other existing puzzle-solving methods, is heuristic approach that allows recording many possible ways of combining elements in searchable and traversable graph structure, the so-called Great Puzzle Graph, or assembly graph. Moreover, the general structure of the method means that no domain knowledge is required at the stage of the search process.

It should be noted that the problem of comparing all elements in all possible settings of individual elements in relation to each other belongs to the so-called NP-complete problems (Demaine & Demaine, 2007). In practice, this means that with the available hardware and time resources, it is not possible to check all possible combinations. Moreover the cardinality, the uncertainty of the completeness of the set, the uncertainty of the element belonging to the set, the uncertainty of interpretation, the radiation uncertainty (Freeman & Garder, 1964) can affect the extension of the reassembly process or the decision to abandon such a process (Stanco et al., 2018). Thus, in addition to checking the current Levenshtein measure as the main method of evaluating the quality of the found potential bonds of 2D elements, there is a need to introduce ways to limit the number of considered potential bonds.

To limit the growth of the Great Puzzle Graph, it is necessary to determine:

- Where the selection can be made? – Selection of a group of potential assemblies that can be evaluated relative to each other.
- What can be cut? – Define the characteristics that will determine the quality of the potential gluing.
- How much can be cut? – Possibility to sort potential assemblies according to their quality and their selection.

2. METHOD AND MATERIALS

2.1. Selected methods to enhance the efficiency

The authors describe enhancing the efficiency of the puzzle-solving method by more accurately assessing correct potential assemblies and by reducing the calculations to be performed and the combinations to be checked in order to achieve the results. To increase the accuracy of the qualitative assessment of the potential assemblies, the authors developed a global fuzzy evaluation system. The assessment constructed based on nine classification rules (using, among others, compact factors) allows to change the multi-criteria sorting of results to single scalar sorting. The fuzzy evaluation system is described in the article (Skulimowski et al., 2022).

The developed method allows for reducing the number of potential assemblies considered and, consequently, reducing the calculations performed by limiting the growth rate of the Great Puzzle Graph in two ways:

- by limiting the number of considered variants that can be added to the assembly graph - MVN (*MaxVariantsNumber*),
- by limiting the number of potential assemblies that can be added to the assembly graph as assemblies of the next order - MLW (*MaxLocalWidth*).

Radical cut-offs strategy, RCO, determined with MVN and MLW, promote the retention and development of higher quality assemblies. The direct effect of using RCO is to reduce the number of comparisons necessary to perform in subsequent steps of the search, and to reduce the cardinality of all possible sets of potential higher assemblies. The number of possible variants of two objects can be expressed by the formula:

$$\left|{}^{RCO}K_{o,e}\right| = \min\left(\left|K_{o,e}\right|, MVN\right)$$

$$o \in (E \cup S) \quad , \; e \in E \quad , \quad o \neq e \tag{3}$$

where: $\left|K_{o,e}\right|$ – the number of combinations of combining objects o and e that meet the criteria of formal, qualitative and linguistic evaluation,

MVN – constant, specifying the maximum number of variants allowed,

E– a set of basic elements,

S – a set of assemblies.

The number of possible assemblies combinations that can be achieved using the RCO strategy can be expressed by a formula:

$$|{}^{RCO}S^{m}| = \begin{cases} min\left(\frac{(|E|-1)|E|}{2} * MVN \ \ , MLW\right) \\ \Leftrightarrow \\ \left|{}^{RCO}K_{e^2{}_i}\right|, \\ \text{for } m=2 \\ \\ min(|{}^{RCO}S^{m-1}| * (|E|-m+1) * MVN \ \ , MLW) \\ \Leftrightarrow \\ min\left(\sum_{i=1}^{|{}^{RCO}S^{m-1}|}\left(\sum_{e\in\left(E-{}^{RCO}S_i^{m-1}\right)}\left(\left|{}^{RCO}K_{\left({}^{RCO}S_i^{m-1},e\right)}\right|\right)\right), MLW\right), \\ \text{for } 2<m<|E| \end{cases} \tag{4}$$

where: MVN – a constant specifying the maximum number of variants allowed,
MLW – a constant specifying the maximum number of potential assemblies allowed to be added to the next row of the assembly graph,
$\left|{}^{RCO}K_{e^2{}_i}\right|$ – a radically limited number of combinations of the i-th pair of two primitives e,
$\left|{}^{RCO}K_{\left({}^{RCO}S_i^{m-1},e\right)}\right|$ – a radically limited number of combinations that can be made from i-th assembly (m-1) order with object e,
$E - {}^{RCO}S_i^{m-1}$ – the set of primitive elements not belonging to the i-th assembly (m-1) of the order.

The rationality of using the RCO must be justified by an appropriate and effective method of selecting potential assemblies, such as the previously described fuzzy evaluation method.

2.2. Detecting where the selection can be made

The developed method can be implemented using depth-first search, DFS and breadth-first search, BFS. The DFS used to expand the assembly-graph allows omitting the determination of all possible assemblies of each row. For the currently checked matches, the possibility of its extension with another original element is checked. When a solution is obtained, a return is made to the closest, lower assemblies row, for which it is possible to check the possibility of creating variants or linking with another primary element, Fig. 2. BFS, on the other hand, allows determining all assemblies of a given row. Adding higher assemblies is possible only when the current row is complete, Fig. 3. This strategy allows for comparing splines of a given row with each other, selecting and developing only selected branches.

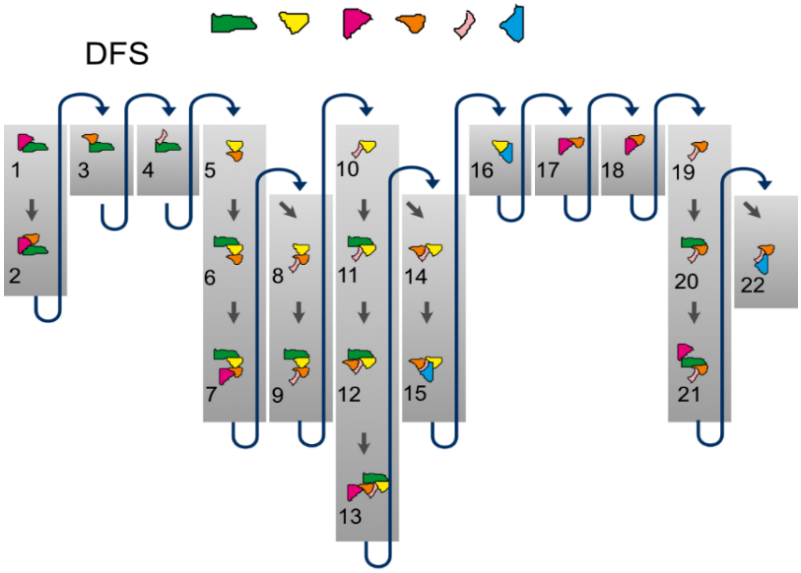

Fig. 2. An example presenting the sequence of adding splines to a graph using DFS. The assembly possibilities of the elements are checked in order from left to right. The numbers indicate the order in which assemblies are added to the Great Puzzle Graph.

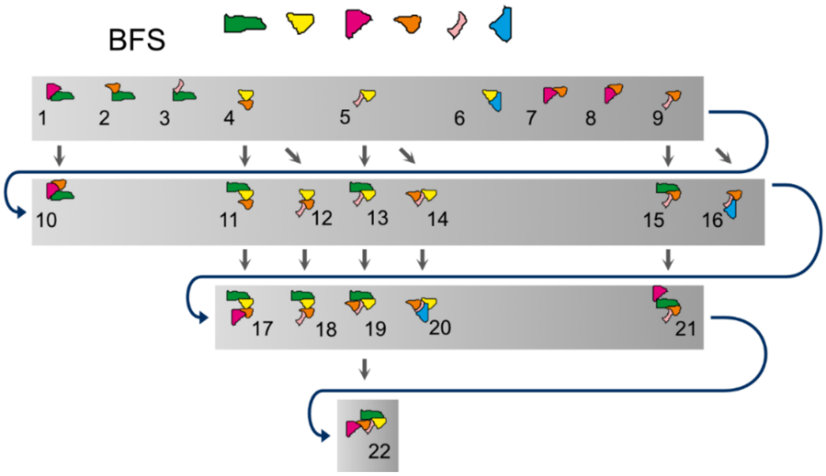

Fig. 3. An example presenting the sequence of adding splines to a graph using BFS. The assembly possibilities of the elements are checked in order from left to right. The numbers indicate the order in which assemblies are added to the Great Puzzle Graph.

The limitation of the growth of a graph built according to the DFS strategy can be made dependent on the maximum number of admissible assembly variants. The internal assessment of the quality of potential bonds is relative and can only be carried out in relation to the bonds of a given row selected for a given moment. Qualitative assessment using BFS can be performed on a set of all potential assemblies of a given order. This enables the absolute selection of the best potential bonds.

2.3. Deciding which elements can be discarded

The basic criteria for evaluating potential assemblies are the allowable margin of matching elements with respect to the Levenshtein method (Maximal Acceptable Value, MAV) and the allowable margin of difference between similar assemblies regarding the Levenshtein method (Minimal Acceptable Value, MIV). The MAV value is used to assess the degree of matching of the fragments of the checked elements. The MIV value is used to assess the degree of dissimilarity of assemblies and is used in the process of detecting and removing assemblies that are very similar to each other, in order to preserve the uniqueness of nodes in the assembly-graph.

Potential assemblies that meet the MAV and MIV margins are subject to further quality checks and selection based on their compactness factors. For the two compared elements 'o' and 'e', the coefficient of compactness of the first type C_S^I and the second type C_S^{II} are expressed by formulas (1) and (2):

$$C_S^I = cl(o) + cl(e) - 2 * cw_{o,e} \tag{1}$$

where: $cl(o), cl(e)$ – Contour length of objects **o** and **e**,
$cw_{o,e}$ – The length of the comparison window used when comparing objects o and e.

The lower the C_S^I the better the match. The C_S^{II} is expressed by the formula:

$$C_S^{II} = \frac{cl(o)+cl(e)}{C_S^I} \tag{2}$$

Markings as in formula (1). The higher the C_S^{II} the better the match.

Tables 1 and 2 show examples of possible values of coefficients C_S^I and C_S^{II} for a set of elements consisting of the same number of characters.

Tab. 1. An example of calculating the value of C_S^I for various assumed lengths of the cw, when the length of the contours of the elements of the original set is equal.

$\|E\| = 10$ $cl(e) = 30$	p_{dcw} value, corresponding to current cw	The number of elements included in the assembly								
		2	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
$cw = 5$	16,7	50	70	90	110	130	150	170	190	210
$cw = 6$	20	48	66	84	102	120	138	156	174	192
$cw = 8$	26,6	44	58	72	86	100	114	128	142	156
$cw = 9$	30	42	54	66	78	90	102	114	126	138
		The value of C_S^I								

When the comparison window is extended, it is clear that the value of C_S^I decreases significantly for a certain number of glued elements, e.g. for 2 elements and the value of cw = 5, and then cw = 9, it decreases by 16%, and in the case of assembling 10 elements, it decreases by as much as by 34%. The obtained results indicate the preference for using windows of greater length.

Table 2. An example of calculating the value of C_S^{II} for various assumed lengths of the cw, when the length of the contours of the elements of the original set is equal.

$\|E\| = 10$ $cl(e) = 30$	p_{dcw} value, corres-ponding to current cw	The number of elements included in the assembly								
		2	3	4	5	6	7	8	9	10
$cw = 5$	16.7	1.2	1.286	1.333	1.364	1.385	1.4	1.412	1.421	1.429
$cw = 6$	20	1.25	1.364	1.429	1.471	1.5	1.522	1.538	1.552	1.563
$cw = 8$	26.6	1.364	1.552	1.667	1.744	1.8	1.842	1.875	1.901	1.923
$cw = 9$	30	1.429	1.667	1.818	1.923	2	2.059	2.105	2.143	2.174
		The value of C_S^{II}								

The presented results indicate that the C_S^{II} index prefers assemblies carried out with an extended comparison window, e.g. for 2 elements and cw = 5 and cw = 9, an increase of about 19% was obtained, and for assembling 10 elements by as much as 53%. Thus, the C_S^{II} well reflects the quality of the assembly.

2.4. Deciding how many items will be discarded

An exemplary comparison of the increase in the number of possible assemblies without and with the use of RCO is presented in Table 3.

Table 3. Comparison of the number of assemblies possible to create with and without the use of RCO.

	1	2	3	4	5	6
Initial values	**Number of possible pairs control strategy**	**All possible pairs $m = 2$**	**All possible threes $m = 3$**	**All possible fours $m = 4$**	**All possible fives $m = 5$**	**Sum of all assemblies**
$\|E\| = 5$ $cl(e) = 15$ $cw = 6$	none	640	1.690e5	3.785e7	5.147e9	5.185e9
	$MVN = 3$ $MLW = \infty$	45	540	6.804e4	1.094e8	1.095e8
	$MVN = \infty$ $MLW = 5$	5	5	5	5	20
	$MVN = 3$ $MLW = 5$	5	5	5	5	20
	$MVN = 5$ $MLW = 10$	10	10	10	10	40
	$MVN = 5$ $MLW = 2000$	75	1500	2000	2000	5575

The use of the MLW mechanism allows to significantly reduce the growth rate of the assembly-graph, which is shown in Table 3. The number of possible combinations has been reduced dramatically by 2 orders of magnitude for column No. 3 (possible threes) by even 6 orders of magnitude for column No. 6 (possible fives). In addition, the use of the second MVN mechanism allows maintaining a higher variety of matches by limiting the number of possible branches of one node.

Due to the design assumptions of the method of creating an assembly-graph, the legitimacy of using the MLW parameter can be justified only in the case of using BFS. This is due to the need to create a cutoff against the set of potential assemblies found for all currently checked nodes of the graph. It also implies the need to sort potential assemblies.

The MVN parameter can be used for both BFS and DFS because it is used when considering possible branches of a single node of the graph, not branches of a group of nodes. Restricted options may or may not be subject to additional cumulative assessment at a later date, based on which they will be re-ranked and re-restricted - this time against the MLW.

3. NUMERICAL EXPERIMENT

The experiment used a set of synthetic data, characterized by repetitive fragments of contours, and thus enabling the creation of many correct potential assemblies (Montusiewicz & Skulimowski, 2020). The experiment was carried out using a proprietary application LiMePuRe-2D (Linguistic Methods for Puzzle Reassemble in 2D) that enables: calibrating parameter values affecting the process of searching for solutions, automating the process of searching for possible matches, filtering and browsing the found solutions, and step-by-step building instruction for each solution (Fig.4). The experiment was carried out using the BFS mechanism to take advantage of the MLW and MVN capabilities.

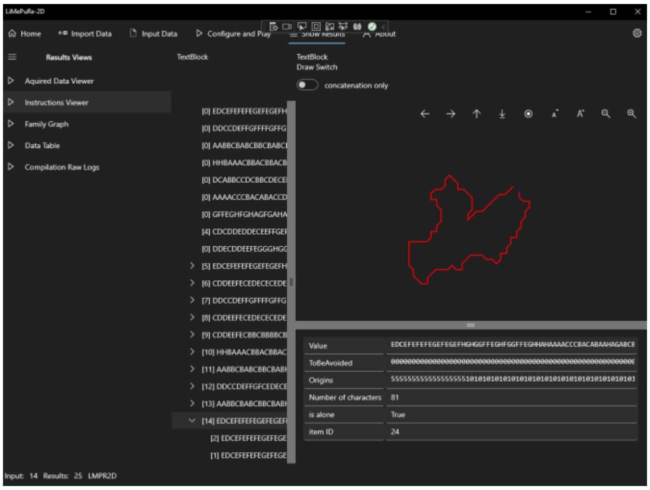

Fig. 4. An example of using the LiMePuRe-2D UWP graphic application, view of the interactive list of found solutions in the form of a list of assemblies.

The research was carried out on a synthetic primary subset with the number of 8 elements, taking elements with the numbers: 1, 2, 3, 4, 5, 11, 12, 13 (Fig. 5). In the experiment, a dynamic window was used, calculated as 20% of the shorter string describing the outline of the compared elements, which resulted in a primary window of 9 to 13. For the same parameter values, the search for assemblies was repeated after applying the fuzzy evaluation mechanism and radical cutoff.

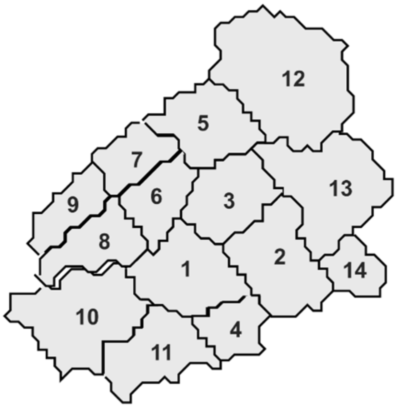

Fig. 5. An example of assembling elements of a synthetic set. The elements are signed with a starting number, used for identification in the conducted experiments.

3.1. Results of the experiments

The experiment defined a way to directly control the number of generated intermediate bonds and solutions, and indirectly control the time available for the search process. The result of the comparisons is a set of potential matches, i.e. those that meet the specified Lev MAV criterion, but have not yet been subjected to a formal, qualitative and quantitative assessment. As a result of filtering the set of potential assemblies through formal, qualitative and quantitative assessment, some of the assemblies are rejected.

Expansion of the assembly-graph using only the local evaluation of potential assemblies is conducive to finding all possible ways to connect elements, but does not take into account the verification of the quality of the connections themselves. The selection of the best assemblies of the kth order was done by selecting the q first potential assemblies from the list sorted according to the given criteria. Among the solutions selected, there were assemblies of 8 elements (Fig. 6) resembling the arrangement of elements in Fig. 5.

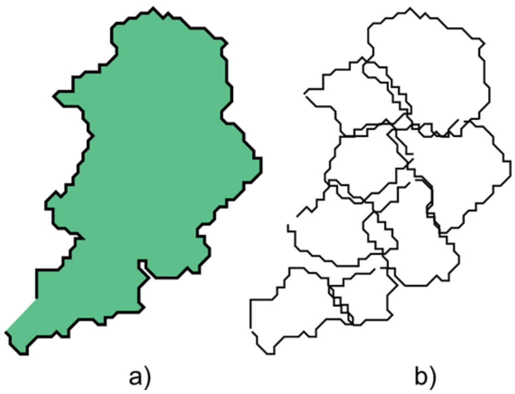

Fig. 6. The solution consisting of 8 elements obtained during the experiments: a) - uniform contour of the obtained solution, b) - arrangement of the primary elements included in the solution.

Among the selected solutions, there are those that can be described as "snake-like", i.e. solutions in which all or almost all of its elements form a string, cannot be combined with more than two elements, and the shape of the assembly itself resembles a curve rather than a compact object, Fig. 7. The C_S^{II} turned out to be important for rejecting this type of assembly.

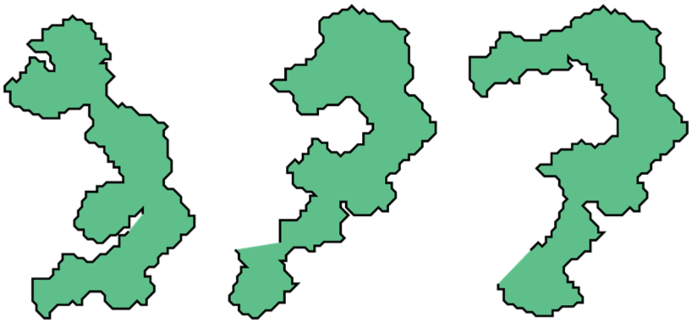

Fig. 7. Images of solutions consisting of 8 elements obtained during experiments with a low value of C_S^{II}.

The total number of comparisons made exceeded 16.4 million, and the maximum number of comparisons made in a given row reaches its maximum only for the 7th order gluing. This means that almost the entire search process required more time and calculations necessary to determine potential assemblies of higher and higher degree. This makes it difficult to predict the allocation of memory resources or time. That should be allocated to the implementation of a given test scenario, and to predict the duration of calculations.

The use of new mechanisms resulted in a significant reduction of the obtained potential assemblies, and the number of new splines stabilized at a constant level and corresponds to the specified maximum number of new assemblies that could be added to the next row in the Great Puzzle Graph - value 5. Only when creating assemblies 7 and 8. This number decreases to 4 and to 1. The qualitative ratio of rejected assemblies to new assemblies and solutions has increased. Without the use of fuzzy evaluation mechanisms and radical cut-offs, this ratio ranged from 1.24 to 1.941, however, after the implementation - from 2 to 9. As a result of the applied MLW constraints, not only the number of new assemblies decreased significantly (to a value not greater than MLW in each row of the graph), but also led to a reduction in the pool of potential assemblies that can be considered in subsequent search iterations from several dozen or several hundred to several.

The number of assemblies in a given degree reached its maximum value at an early stage of the search. The cumulative data sets before and after the introduction of the fuzzy assessment and the RCO strategy are presented in Table 4.

The difference in the number of comparisons made in a given degree of assemblies for different algorithms is 2 orders of magnitude. After the use of fuzzy evaluation mechanisms and RCO, both the number of comparisons and the number of potential assemblies changed from rapidly increasing to systematically extinguishing as the search process progressed. Furthermore, it can be noticed that the increase in cumulative number of comparisons before the implementation of the RCO mechanisms resembles the exponential function, while the increase after the introduction of the RCO mechanisms resembles the logarithmic function. Determining the limit for the latter is seemingly more intuitive and simpler. The ability to predict the time needed to complete the search process can be considered a valuable factor in starting the search process for a given set of parameter values.

Table 4. The number of comparisons made in a given row of assemblies for different algorithms.

Assembly degree	The number of comparisons made in a given degree of assembly		Difference [%]
	Before adding of fuzzy evaluation and the RCO strategy	After adding of fuzzy evaluation and the RCO strategy	
2nd degree	4.636e4	4.636e4	0
3rd degree	2.760e5	8.857e4	212
4th degree	9.844e5	9.595e4	926
5th degree	2.459e6	9.118e4	2597
6th degree	4.399e6	7.759e4	5570
7th degree	5.368e6	5.638e4	9421
8th degree	2.878e6	2.599e4	10973

Relationships in the growth rate of the number of comparisons between successive degrees of assemblies were also observed, Fig. 8. Both in the variant using fuzzy logic and RCO as well as in the variant without additional evaluations and criteria, the largest difference in the number of comparisons (proportionally) occurs in the initial phase of the comparison process - for 3rd degree assemblies. The number of comparisons after 3rd order assembling continues to increase for the scenario without fuzzy evaluation, and for the scenario using fuzzy evaluation - the number of comparisons systematically decreases after 4th degree assemblies.

The number of performed comparisons reached its maximum value already for the 4th degree assemblies. This means that the assemblies of all higher orders were determined using fewer comparisons and in a shorter time.

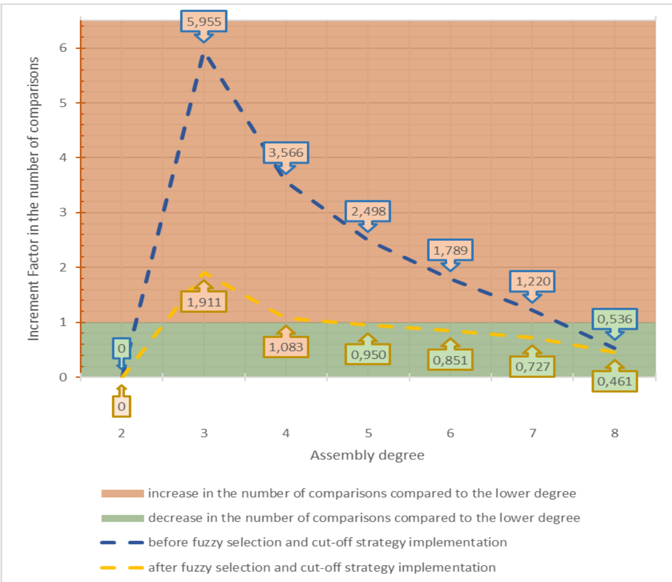

Fig. 8. Comparison of the increase in the number of comparisons carried out for assemblies of successive degrees (ratio of k-order to (k-1)-row) before and after applying the fuzzy and RCO evaluation mechanisms.

A significant reduction in calculation time was also achieved - it was able to quickly interrupt the function checking the suitability of a questionable gluing and proceed to checking the next one. However, due to the increase in computational complexity caused by the use of a script implementing fuzzy logic, the change in time is not linear. Nevertheless, the overall reduction in computation time reliably compensates for the additional computational overhead. On the computer set used, it was about a threefold reduction in the calculation time, from 66 seconds to 21.5 seconds.

4. CONCLUSIONS

Comparing the assemblies of a given degree to each other made it possible to narrow down the list of potential correct expansions of the graph more precisely, and consequently to reduce the number of considered cases, the number of comparisons and calculations performed.

The change of the search strategy from the assumed maximum number of variants of assemblies of the kth order to the maximum number of unrolled assemblies of the kth order turned out to be a compelling tool for optimizing the operation of the method, directly reducing the number of checked joints. The advantage of using such a strategy can be achieved only if the criteria for evaluating the quality of k-th degree assemblies are good enough. Otherwise, it may happen that the only unrolled merges are those in doubt. Overly stringent calibrations led to distortion and rejection of most or even all potential assemblies.

The authors are aware that adding a fuzzy evaluation module increases the computational complexity for each assemblies degree. The authors consider this to be an appropriate price for a more accurate selection of candidate assemblies and a reduction in the cumulative number of computations. The actual reduction in complexity must be based on the balance between the additional complexity of the fuzzy evaluation process and the number of comparisons that could be avoided by RCO. A detailed analysis of this phenomenon may be the subject of another article.

Author Contributions

- Stanisław SKULIMOWSKI – the main research, test environment design,
- Jerzy MONTUSIEWICZ – research planning and progress verification,
- Marcin BADUROWICZ – content correction, consistency of content.

REFERENCES

Andreadis, A., Papaioannou, G., & Mavridis, P. (2015). Generalized digital reassembly using geometric registration. *2015 Digital Heritage International Congress* (pp. 549–556). IEEE. https://doi.org/10.1109/DigitalHeritage.2015.7419572

Brown, B. J. (2008). *Registration and matching of large geometric datasets for cultural heritage applications*. Princeton University.

Chang, S. K., & Chow, C. K. (1973). The Reconstruction of three-dimensional objects from two orthogonal projections and its application to cardiac cineangiography. *IEEE Transactions on Computers*, *C–22*(1), 18–28. https://doi.org/10.1109/T-C.1973.223596

Demaine, E. D., & Demaine, M. L. (2007). Jigsaw puzzles, edge matching, and polyomino packing: Connections and complexity. *Graphs and Combinatorics*, *23*, 195–208. https://doi.org/10.1007/s00373-007-0713-4

Freeman, H., & Garder, L. (1964). Apictorial jigsaw puzzles: The computer solution of a problem in pattern recognition. *IEEE Transactions on Electronic Computers*, *EC-13*(2), 118–127. https://doi.org/10.1109/PGEC.1964.263781

Montusiewicz, J., & Skulimowski, S. (2020). A search method for reassembling the elements of a broken 2D object. *Advances in Science and Technology Research Journal*, *14*(3), 49–56. https://doi.org/10.12913/22998624/122570

Oxholm, G., & Nishino, K. (2013). A flexible approach to reassembling thin artifacts of unknown geometry. *Journal of Cultural Heritage*, *14*(1), 51–61. https://doi.org/10.1016/j.culher.2012.02.017

Papaioannou, G., Karabassi, E. A., & Theoharis, T. (2001). Virtual Archaeologist: Assembling the past. *IEEE Computer Graphics and Applications*, *21*(2), 53–59. https://doi.org/10.1109/38.909015

Rasheed, N. A., & Nordin, M. J. (2014). A polynomial function in the automatic reconstruction of fragmented objects. *Journal of Computer Science*, *10*(11), 2339–2348. https://doi.org/10.3844/jcssp.2014.2339.2348

Rasheed, N. A., & Nordin, M. J. (2015a). A Survey of computer methods in reconstruction of 3D archaeological pottery objects. *International Journal of Advanced Research*, *3*(3), 712-714.

Rasheed, N. A., & Nordin, M. J. (2015b). A survey of classification and reconstruction methods for the 2D archaeological objects. *2nd International Symposium on Technology Management and Emerging Technologies (ISTMET)* (pp. 142–147). IEEE. https://doi.org/10.1109/ISTMET.2015.7359018

Rasheed, N. A., & Nordin, M. J. (2020). Classification and reconstruction algorithms for the archaeological fragments. *Journal of King Saud University - Computer and Information Sciences*, *32*(8), 883–894. https://doi.org/10.1016/j.jksuci.2018.09.019

Skulimowski, S., & Montusiewicz, J. (2020). Optimization methods of searching algorithms for 2D elements matching. *Modern Computational Methods and Their Applications in Engineering Science* (pp. 35–47). Wydawnictwo Politechniki Lubelskiej.

Skulimowski, S., Montusiewicz, J., & Badurowicz, M. (2022). The use of fuzzy evaluation and radical cut-off strategy to improve apictorial puzzle assembly with exhaustive search algorithm performance. *Advances in Science and Technology Research Journal*, *16*(2), 179–187. https://doi.org/10.12913/22998624/147024

Stanco, F., Battiato, S., & Gallo, G. (2018). Digital reconstruction and mosaicing of cultural artifacts. In F. Stanco, S. Battiato, & G. Gallo (Eds.), *Digital Imaging for Cultural Heritage Preservation* (pp. 353–384). CRC Press.

Vendrell-Vidal, E., & Sánchez-Belenguer, C. (2014). A discrete approach for pairwise matching of archaeological fragments. *Journal on Computing and Cultural Heritage*, *7*(3), 15. https://doi.org/10.1145/2597178