

The Use of Fuzzy Evaluation and Radical Cut-Off Strategy to Improve Apictorial Puzzle Assembly with Exhaustive Search Algorithm Performance

Stanisław Piotr Skulimowski^{1*}, Jerzy Montusiewicz¹, Marcin Badurowicz¹

¹ Department of Computer Science, Faculty of Electrical Engineering and Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

* Corresponding author's e-mail: s.skulimowski@pollub.pl

ABSTRACT

The paper presents an approach to solving the problem of assembling broken, flat elements using a letter notation of the elements' contours and checking their matching using linguistic methods. Previous studies with the use of exhaustive search have shown effectiveness in finding possible connections, but they are burdened with a large number of calculations and the time needed to carry them out. In order to accelerate the process of searching for solutions, the possibility of using a fail-fast method of fuzzy assessment of potential combinations of elements was checked, as well as the method of cutting off potential, but not effective connections. The numerical experiment carried out showed a significant reduction in the number of trials and total computation time while maintaining the quality of the potential solutions found.

Keywords: fuzzy logic, exhaustive search, reassembling, linguistic methods, puzzle, contour description, fail-fast design, cut-off strategy.

INTRODUCTION

The process of actual reconstruction of cultural heritage objects found in archaeological excavations in the form of individual elements is a tedious, ineffective activity, with a high chance of errors. Similar problems can be found in medicine (e.g., matching bone fragments), forensics (e.g., reproducing evidence) – a lot of small elements not distinguished by texture, yet unique in terms of shape. The sum of these features creates a paradox of a task of potentially significant importance and high priority, yet too costly in terms of the work to be conducted. In terms of the number of combinations to check, rectangle packing puzzles, square packing puzzles, jigsaw puzzles or polyomino packing puzzles, can be a computational challenge [1]. Additionally, apictorial puzzles require taking into account many non-obvious metrics[2]. Despite this, in the last fifty years, at least several dozen studies have been conducted using various, automatic and

semi-automatic, case-specific methods of arranging 2D puzzles[3], proving not only that it is an important issue, but a feasible task. Moreover, the methods that have already been described, like 2D edge matching[4], are still being developed, for example, to find relationships in the shape of relief elements[5], use new tools, like 3D scanners, and change the scope of operability, for example, to match 3D polygonal arcs[6,7].

The authors propose their approach to solve the problem of apictorial 2D puzzles, using the description of the edges of artefacts using the compass rose (CR) [8], consisting of unit vectors of the same length, differing in direction and sense. Each of the vectors is assigned a different letter of the alphabet [9]. After assigning successively determined vectors to the contour of the element, knowing their sequence and the letter belonging to them it was possible to define the contour entry in the form of a sequence of characters (Fig. 1). The process of creating the contour, its features, as well as the principle of operation

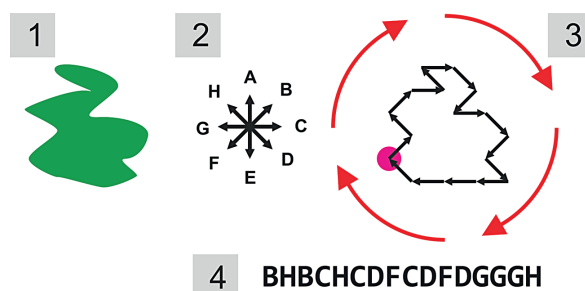


Fig. 1. An example of creating an abstract word that describes the outline of an element: (1) object, (2) the eight-point CR, (3) object edge notation using CR vectors (red point – start position, red arrows – the direction of contour drawing), (4) character code after changing vectors into characters

of the method have already been discussed in the article [10]. The description of the contour used differs from those that can be found in the literature, e.g. [11, 12] because it carries information about both the shape and length. The Levenshtein metric was used to compare the contours of the two analysed elements [13] and check the number of insertions, deletions, and substitutions needed to make strings identical.

When comparing two strings, Levenshtein metric returns zero if they are identical, a value of one is returned if the characters being compared at the same positions in the strings do not match, or when a single character in one of the strings must be added or removed to match. The comparison of strings is performed for the assumed length of the substring of characters describing the contour. When the returned value of a comparison using Levenshtein’s metric is zero, the analysed elements match the selected section. The connection that will be made on a substring as long as possible is the most desirable. The matched elements are combined, and a description of their common

contour is generated automatically, creating a new sequence of characters. The next element from the set of available elements is selected for the assembled object to check its fit and, if it meets certain matching criteria, it is added to the existing object.

The procedure of matching and assembling is conducted until no elements are left for comparison. The developed method based on the exhaustive search is able to determine all possible combinations of connections. However, this is burdened with a large number of calculations (including comparing sub-strings, of a certain length, from every two elements contours, shifted by one character) and a long execution time [14]. The complexity of calculations, as the number of computations and the time it takes to complete them, grows exponentially with each successive element added to the initial pool (Table 1) [10]. The obvious solution would be to implement vertical scaling (increasing the computing power of the calculating machine, concurrent processing), horizontal scaling (spreading calculations over many machines) or both. This action would lead to a general reduction in the duration of searching for all possible solutions, however it would not help to improve the quality, which can be interpreted, inter alia, as performing mainly processes that can bring the actual results. Unfortunately, without specifying the method for assessing potential combinations, there is no certainty whether the indicated elements will or should match (example Fig. 2).

For example, treating assemblies constructed of the largest number of elements from the initial pool cannot be considered a right or accurate composition due to the uncertainty about the quality of the initial elements pool (Does the initial set contain all the necessary elements? Does it contain duplicates? Does the set have pieces

Table 1. A list of the theoretical maximum number of comparisons needed to achieve possible combinations

The number of elements in the initial set ¹	The approximate number of comparisons needed ²	The number of possible assemblies by a number of elements included:			
		2	3	4	5
5	8.370*10 ⁵	10	30	60	60
7	6.116*10 ⁷	21	105	420	1260
9	6.634*10 ⁹	38	266	1596	7980
11	9.638*10 ¹¹	59	531	4248	29736
13	1.857*10 ¹⁴	84	924	9240	83160
15	4.627*10 ¹⁶	113	1469	17628	193908

¹ Average element length – 53.

² Approximation for the length of the substring being compared – 15.

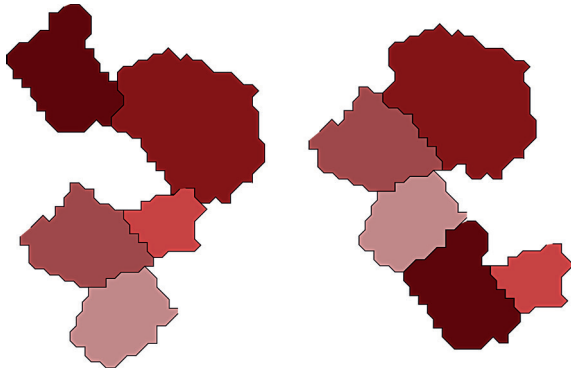


Fig. 2. An example of assembling five elements in two extremely diverse methods. Without a suitable grading method, both must stay as relevant up to the end of calculations

from only one puzzle?). It seems natural to limit the number of tested assemblies, leaving only the most promising ones. In this way, it would be possible to achieve both an increase in quality and a reduction in the time needed to perform the calculations. The results of the previous numerical experiments [10, 14] showed that in the constructed algorithm too many comparisons are made, even in situations that do not lead to obtaining solutions forming a compact system. The attempts to limit the number of potential connections, by implementing fail-fast components, so far have proved that it is possible to rapidly reduce the final number of comparisons made (Table 2). The methods used so far were:

- (e_v1) duplicate rejection mechanism and elimination of elements overlapping;
- (e_v2) dynamic window mechanism (DPL and DPL2 – described later in the paper) and elements from e_v1;
- (e_v3) dynamic comparison window as a list of values and elements from e_v1.

The introduction of strict rules so far could cause a decrease in the number of possible solutions. Notwithstanding, it does not have to

correlate with decrease in the number of connections necessary to make (Table 2).

Moreover, the connections determined in this way are not very revealing and according to expert judgment) do not differ significantly from the solutions found using a simpler method This confirms the need to find a better way to assess the suitability of potential connections, to implement a better, even more radical method of cutting off meaningless connections. The following questions arise: What indicators should be included in the aggregate evaluation of the connected parts in order to have additional information about the quality of the fitting and assemble performed?; is it worth continuing the matching and assemble process for each selected item from the set of available items? Looking for answers to the above questions, it was decided to conduct an experiment using the method of fuzzy evaluation of potential connections, due to various possible methods of assembling and many methods to evaluate them. Fuzzy logic mechanisms are now widely used in natural sciences, technical sciences and industry, from industrial manufacturing, automatic control, automobile production [15], through techniques of image recognition and grouping images [16] to even to bird migration predicting [17].

There have already been scientific papers that use fuzzy logic in the form of a modified Levenshtein algorithm – Fuzzy Levenshtein for linguistic calculations [18]. However in this work, the traditional, deterministic version of Levenshtein algorithm. The fuzzy logic system was used only to assess the quality of assemblies of elements due to the uncertainty regarding all the calculated indicators. Although the problem can be considered purely as a combinational one, the intention of using fuzzy logic element in the described algorithm, as well as the main motivation behind the research described below, is not to reduce the number of comparisons. The

Table 2. The number of possible assemblies and the number of accumulated comparisons

Evaluation	The number of comparisons needed	The number of assemblies of different number of elements ¹			
		2	3	4	5
Theoretical	837000	10	30	60	60
e_v1	312199	0	3	6	14
e_v2	149899	1	4	5	1
e_v3	617796	0	5	6	1

¹ Numbers from checking the different number of elements (from 2 to 5) and checking only one way of potential assembly.

motivation is to lead to a situation in which part of comparisons would not have to be made-by, for example, indicating a more precise adjustment of the elements' bond tolerances.

The goals of the study are: (G1) to define indicators to determine the suitability of the created matches for the further process of searching for joints; (G2) to determine the degree of optimisation of the method's performance, considering the reduction in the number of comparisons and the number of potential yet ineffective connections.

PREPARATION OF A FUZZY EVALUATION MECHANISM

Indicators for the fuzzy evaluation system of the degree of matching elements

The following indicators were defined to build mechanisms that would allow to automatically control the operation of the algorithm for matching elements and then assemble them:

1. (DPL) the substring length – the number of characters forming the abstract word taken from the contour describing one element, which in the given calculation phase is used to compare the correspondence when analysing the two considered elements.
2. (C1) object outline length is the number of characters describing the outline after assembling the two compared elements

$$C1 = N1 + N2 - 2 * DPL \quad (1)$$

where: $N1$ and $N2$ denote the number of characters of the first and second elements being compared, respectively.

3. (DPL2) the ratio of the length of the substring number of characters to the length of the shorter of the compared elements

$$DPL2 = \frac{DPL}{\min(N1,N2)} \quad (2)$$

For the typical assemblies of the analysed elements, the values of $C2$ range from 0.15 to 0.30. The program can be initiated such that the operator indicates the DPL value, and the DPL2 value will be calculated automatically. The reverse is also possible - the operator indicates the desired DPL2 directly and the DPL value is automatically calculated.

4. (C2) the contours of the object after assembling are the quantity expressing the relationship

between the original lengths of the contours of the compared elements (first and second) and the length of the contour after assembling.

$$C2 = \frac{(N1+N2)}{C1} \quad (3)$$

For the typical assemblies of the analysed elements, the values of $C2$ range from 1.15 to 1.30.

Fuzzy rules

The prepared DPL, DPL2, $C1$, and $C2$ indicators, were divided into three classes (“poor”, “average” and “good”) with the use of automatic division into three classes of membership by triangular functions, where the minimum and maximum of the range were always automatically calculated for a given set of compared elements. An example of such a division for the $C1$ parameter is shown in Figure 3. The fuzzy logic system was developed based on twelve rules [19, 20] where the $C1$ index was preferred first, then $C2$, DPL and DPL2, while forcing a low overall assessment in the case of a low value of the $C2$ index, independently of the others (Table 3). The created rules represent a fail-fast ruleset system design [21, 22] – only two rules describe the result potentially suitable for future evaluation (“good”), three rules describe the result at the limit of usefulness (“average”), and when the remaining seven rules describe the result considered to be weak or incorrect (“poor”).

The rule set has been designed to be consistent with the other testing components included in the developed method (e_v2) and to ensure that the poorly promising solutions will be screened out. The fuzzy system calculates the result as a numerical index on a scale from 0 to 9, using the three classes of result membership as well as centroid defuzzification. The numerical fuzzy connection rating index is used to sort the elements

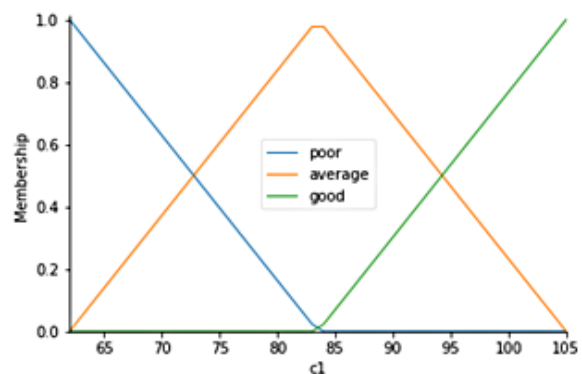


Fig. 3. Membership functions for the indicator $C1$

Table 3. Description of fuzzy rules used in the experiment

Rule	Results
C1, C2, DPL and DPL2 are GOOD	Good
C1 and DPL are GOOD, while C2 and DPL2 are AVERAGE	Good
C1, C2 and DPL are GOOD, while DPL2 is AVERAGE	Average
C1 and DPL2 are GOOD, while C2 and DPL are AVERAGE	Average
C1 AVERAGE and C2 POOR automatically classify the outcome to AVERAGE, independently of the remaining ones	Average
C1 and DPL are AVERAGE, even if the remaining ones are GOOD	Poor
All are AVERAGE, even if C2 is GOOD	Poor
C1 is POOR, and the remaining ones GOOD	Poor
C1 is POOR, C2 AVERAGE, and the remaining ones POOR	Poor
C1 is POOR, and C2 AVERAGE, even if DPL and DPL2 are GOOD	Poor
C1 is POOR, and C2 AVERAGE, even if DPL is AVERAGE	Poor
C2 is POOR, independently of the remaining ones	Poor

and therefore select the best assemblies in terms of compactness, the ratio of the length of the constituent elements to the length of the resulting element etc.. In such an approach and simple the scope of the use of fuzzy logic is not sufficient to conduct an exhaustive assessment. There is a lack of probabilistic hesitant fuzzy preference relations or multiplicative consistency analysis [23]. It was a conscious decision not to delve into these aspects of the fuzzy evaluation and preferred to focus on the very determination whether the very use of the fuzzy evaluation would positively affect the quality of the algorithm’s work.

Algorithm for match search with global selection and radical cut-off mechanism

So far, the program has been based only on crisp logic and consisted, not counting initialization step (Fig. 4. Original A) of the following steps:

- Searching for possibilities to combine two elements where the first element comes from the initial set, or it is the result of assembling other elements and the second element comes from the initial set (Fig. 4. Original B),
- Creating a temporary subset containing all possible connections for two selected elements, sorting them (Fig. 4. Original C), and selecting specific ones according to the adopted criteria (local selection),
- Creating a global set of sorted combines with crisp logic only (Fig. 4. Original D),
- Repeating the process from searching for possibilities to combine two elements for each newly discovered assembly considered as “the first

element” (Fig. 4. Original B) The process is repeated until the set of comparable elements is exhausted.

- The experiment described in this paper involved introducing a temporary global set for which re-selection of a chosen number of matches (cut-off) for further searches (global selection) is performed. The procedure’s other elements remained unchanged (Fig. 4. Modified A, B, C) The temporary global set involves two modifications:
- The fuzzy evaluation mechanism of the usefulness of element connections to global selection on solutions stored in the temporary set (Fig. 4. Modified E).
- The cut-off strategy from leaving only a few from the global temporary set (Fig. 4. Modified F).

The example in Figure 4 shows a scenario where no more than three matches are selected from each temporary subset. The novel approach is to check the quality of matches from all temporary subsets in the context of all found matches in one temporary global set. Potential combinations are ranked according to the fuzzy score value and the top five of them are selected, to identify the most promising solutions from the entire temporary set, regardless of what elements the connection concerns. The number of five was selected based on previous experience with verifying the usefulness of the solutions created. A preliminary analysis of the algorithm’s complexity showed that the version enriched with fuzzy logic is more complex. It is assumed that that performing a larger operation on a single potential assembly, which results in the rejection of incorrect or unpromising

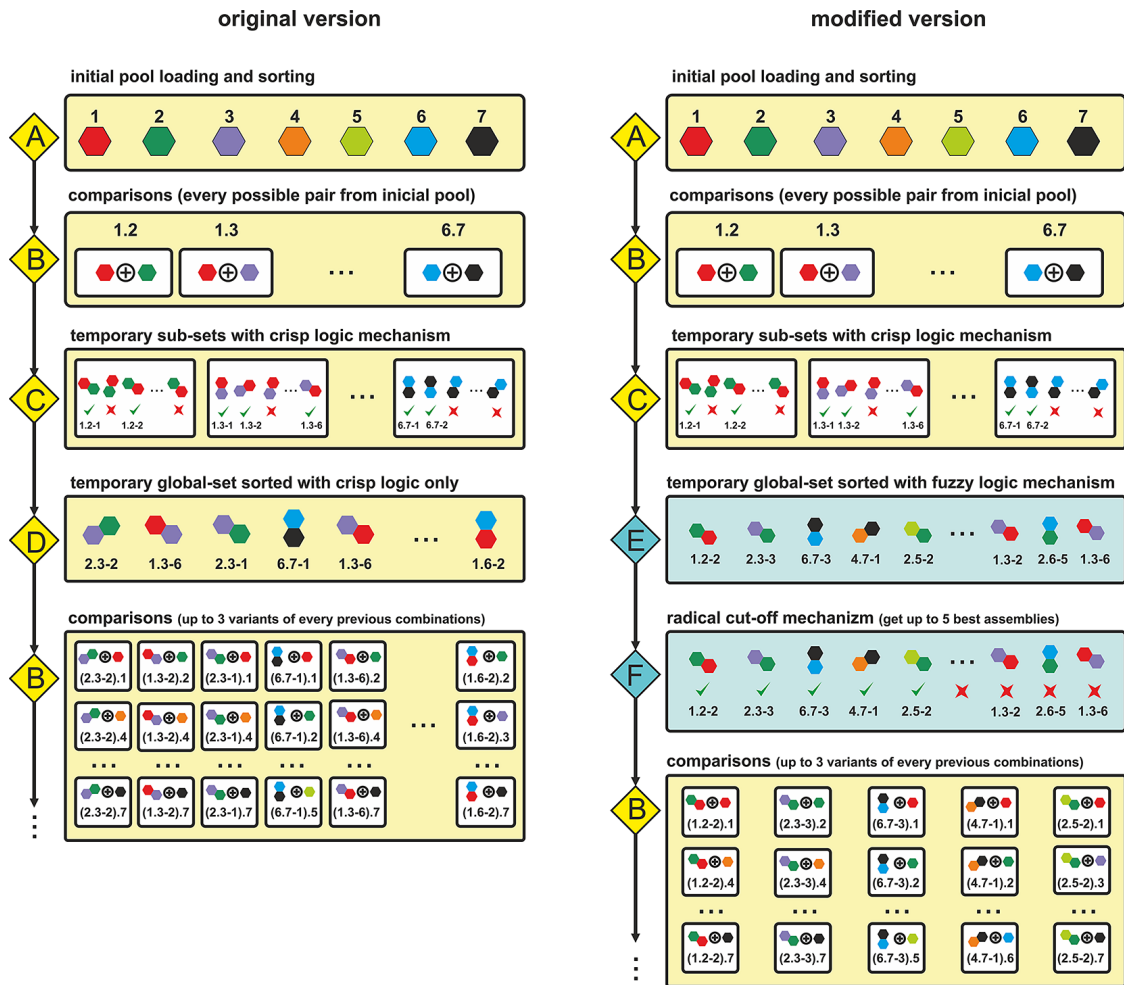


Fig. 4. Comparison of two algorithm versions: Original – the algorithm for match search used so far: green check icon –promising assembly that can be taken to the next step, red cross icon – a poor assembly that is deleted from sets; Modified – the algorithm for match search with a new approach: green check icon –promising assembly that can be taken to the next step, red cross icon – a poor assembly that is deleted from sets

bifurcations (local intensity), will eventually lead to a decrease in the overall number of computations (global extensiveness, low intensity).

NUMERICAL EXPERIMENT

To conduct the experiment, an additional script was created, dedicated to fuzzy evaluation purpose (Fig. 4. Modified E). The extracted partial results from the main application’s loop were used and were placed in the fuzzy logic system based on the scikit-fuzzy library in version 0.4, using the rule system described earlier. Tests were conducted in Python version 3.9 on the Windows platform. Numerical experiments were performed based on synthetic data of a defined set of elements for which a correct total solution is known [14]. A constant threshold in Levenshtein metrics

was assumed for all comparisons (Fig. 4. Original C and Fig. 4. Modified C), as well as the initial arrangement of elements (Fig. 4. Original A and Fig. 4. Original A). The initial elements were described in one fixed manner in terms of contour accuracy, selection of the starting point for creating contour description etc. The e_v2 method of limiting the number of potential connections was used for both not fuzzy and fuzzy evaluation tests. The limit values for all parameters (DPL, DPL2, C1 and C2) were determined on the basis of the minimum and maximum value from the temporary global set.

RESULTS OF THE EXPERIMENT

Table 4 presents the results of the experiment conducted and compared them with the previous results. The table describes, among others:

Table 4. List of initial parameters of the elements used for the experiment, with mid-process evaluation and comparison of current results (NFG) with new ones (FG)

Initial parameters		Mid-process evaluation parameters ¹				Final scores	
N1	N2	DPL	DPL2	C1	C2	NFG	FG
40	40	8	0.2	64	1.250	8	2.2971
		9	0.225	62	1.290	9	1.5366
61	45	9	0.2	88	1.205	7	3.7128
		10	0.223	86	1.233	8	3.7319
		11	0.244	84	1.262	9	3.7327
40	72	9	0.225	94	1.191	9	3.5722
		10	0.25	92	1.217	8	3.6358
60	72	13	0.217	106	1.245	9	4.3356
		14	0.233	104	1.269	8	4.4957
		15	0.25	102	1.294	7	4.4589

¹ DPL2 was calculated automatically based on DPL value.

- NFG – not fuzzy grade, the assessment used so far (9 – the best, 0 – the worst)
- FG – fuzzy grade (9 – the best, 0 – the worst)

As results of the calculations, the application of the system of limiting the composition of elements through their evaluation in the fuzzy logic system with special allowed for a better selection of elements for the search for further assumptions, reducing the number of possible comparisons by

95% (Fig. 5, 6). The introduced fuzzy logic system was used as a special selector and its compliance with the expert judgment allowed for the selection of the potentially most promising assemblies. To be able to perform such drastic cuts of complex elements not showing a worth continuation, the evaluation method must be precise enough to deal with the uncertainty. The proposed fuzzy system meets the expectations set for it, which results from the data obtained from the numerical experiment.

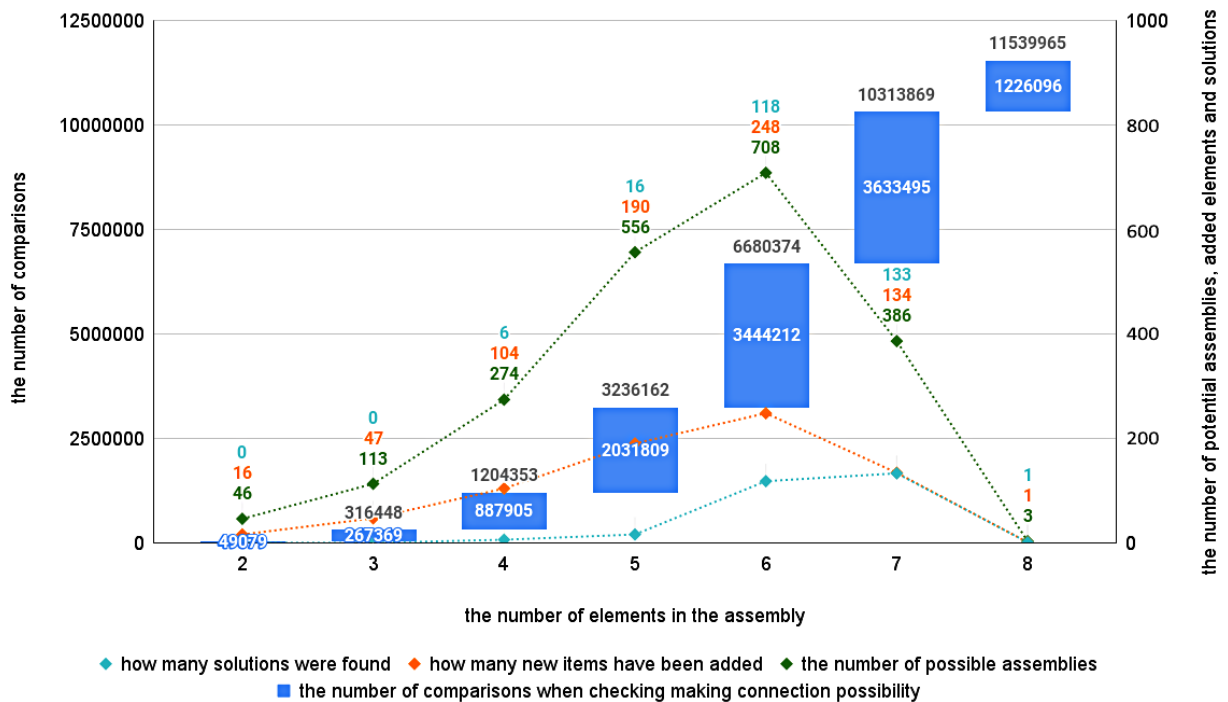


Fig. 5. The number of possible assemblies and the number of comparisons when checking the possibility of a different number of elements (from 2 to 8), before implementing the use of fuzzy evaluation mechanism (only crisp logic evaluation) and new method of cut-offs

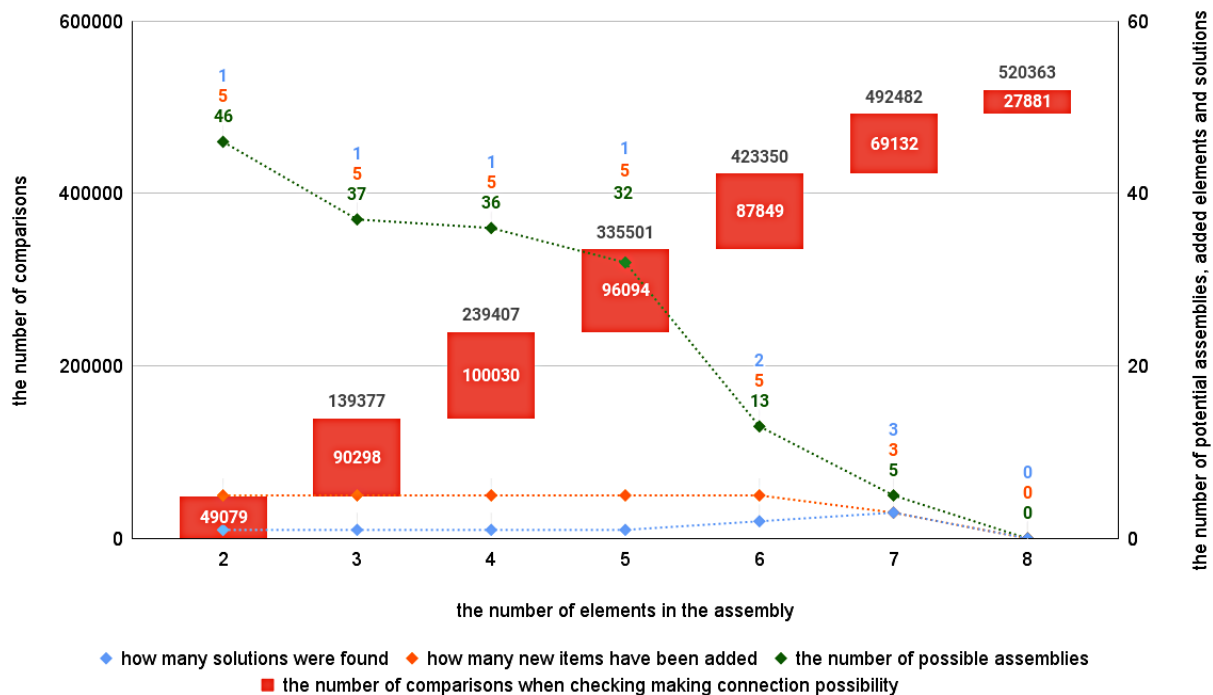


Fig. 6. The number of possible assemblies and the number of comparisons when checking the possibility of a different number of elements (from 2 to 8), after implementing the use of fuzzy evaluation mechanism and new method of cut-offs

CONCLUSIONS

The results obtained from the conducted research made it possible to achieve the set research objectives: the adopted indicators (DLP, DPL2, C1 and C2) turned out to be useful for selecting solutions that are stored in a subset of interim solutions(G1); It is possible to unambiguously define the degree of optimization of the method's performance, considering the reduction in the number of comparisons and the number of potential yet ineffective connections (G2). For an 8-item set, this may mean reducing the number of comparisons needed to find 8-item solutions, from over 11.5 million to just around 520,000. The experiment confirmed the potential of using fuzzy logic for the created algorithm. The fuzzy logic module provides an additional method of checking the qualitative assessment of the assemblies generated by the program. However, as the conducted experiments have shown, fuzzy logic mechanisms are not universal. The fuzzy evaluation is the more useful the larger the checked data set is. For small data sets, the fuzzy evaluation may not be reliable. The integration of both methods of elimination, fuzzy evaluation and radical cut-off strategy, guarantees both the high quality of potential solutions

and their small number. Additionally, in contrast to the previously used rigid elimination rules, the developed fuzzy assessment tool has a greater potential for future modifications with the use of expert and domain knowledge.

Acknowledgments

The work was co-financed by the Department of Computer Science Lublin University of Technology and Lublin University of Technology Scientific Fund FD-IT-001.

REFERENCES

1. Demaine E.D., Demaine M.L. Jigsaw puzzles, edge matching, and polyomino packing: Connections and complexity. *Graphs Comb.* 2007; 23(S1): 195–208. <http://dx.doi.org/10.1007/s00373-007-0713-4>
2. Freeman H., Garder L. Apictorial jigsaw puzzles: The computer solution of a problem in pattern recognition. *IEEE trans electron comput.* 1964; EC-13(2): 118–127. <http://dx.doi.org/10.1109/pgec.1964.263781>
3. Rasheed N.A., Nordin M.J. A survey of classification and reconstruction methods for the 2D archaeological objects. 2015 International Symposium on Technology Management and Emerging Tech-

- nologies (ISTMET), 2015, 142–147. <http://dx.doi.org/10.1109/ISTMET.2015.7359018>
4. Kong W., Kimia B.B. On solving 2D and 3D puzzles using curve matching. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR 2001. IEEE Comput. Soc; 2005.
 5. Willis A.R. Computational Analysis of Archaeological Ceramic Vessels and Their Fragments. In: Lukac R, editor. Digital Imaging for Cultural Heritage Preservation. Boca Raton, FL: CRC Press; 2011, 323–252.
 6. Zhou M., Geng G., Wu Z., Zheng X., Shui W., et al. A System for Reassembly of fragment Objects and Computer Aided Restoration of Cultural Relics. Virtual Retrospect 2007, Robert Vergnieux, Pessac, France 2007, 21-27.
 7. Andreadis A., Papaioannou G., Mavridis P. Generalized digital reassembly using geometric registration. In: 2015 Digital Heritage. IEEE 2015.
 8. Eureka!, 3 compass games that teach kids to use a compass [Internet]. Eureka camping.com. [cited 2022 Jan 22]. <https://www.eurekacamping.com/blog/article/3-compass-games-teach-kids-use-compass>
 9. Freeman D. A corner-finding algorithm for chain-coded curves. IEEE Trans Comput. 1977; C26(3): 297–303. <http://dx.doi.org/10.1109/tc.1977.1674825>
 10. Montusiewicz J., Skulimowski S. A search method for reassembling the elements of a broken 2D object. Adv Sci Technol Res J. 2020; 14(3): 49–56. <http://dx.doi.org/10.12913/22998624/122570>
 11. Grabowik C., Kalinowski K., Ćwikła G., Gwiazda A., Monica Z. The use of chain codes to describe the structure and identify structural elementary objects. In: Innovations in management and production engineering. Opole, Poland: Publishing House of the Polish Production Management Society. 2017; 180–90.
 12. Karczmarek P., Kiersztyn A., Pedrycz W., Dolecki M. An application of chain code-based local descriptor and its extension to face recognition. Pattern Recognit, 2017; 65: 26–34. <http://dx.doi.org/10.1016/j.patcog.2016.12.008>
 13. Buschmann T., Bystrykh L.V. Levenshtein error-correcting barcodes for multiplexed DNA sequencing. BMC Bioinformatics. 2013; 14(1): 272. <http://dx.doi.org/10.1186/1471-2105-14-272>
 14. Skulimowski S., Montusiewicz J. Optimization methods of searching algorithms for 2D elements matching. In: Katarzyna Falkowicz MS, editor. Modern Computational Methods and their Applications in Engineering Science. Lublin, Publishing house of the Lublin University of Technology; 2020; 35–47.
 15. Bai Y., Wang D. Fundamentals of fuzzy logic control — fuzzy sets, fuzzy rules and defuzzifications. In: Advances in Industrial Control. London: Springer London; 2007; 17–36.
 16. Korytkowski M., Scherer R., Szajerman D., Polap D., Wozniak M. Efficient visual classification by fuzzy rules. In: 2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE). IEEE 2020.
 17. Czerwinski D., Kiersztyn A., Lopucki R., Karczmarek P., Kitowski I., Zbyryt A. An application of fuzzy C-means, fuzzy cognitive maps, and fuzzy rules to forecasting first arrival date of avian spring migrants. In: 2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE). IEEE 2020.
 18. Chakrabarti T., Saha S., Sinha D. DNA multiple sequence alignment by a hidden Markov model and fuzzy Levenshtein distance based genetic algorithm. Int J Comput Appl. 2013; 73(16): 26–30. <http://dx.doi.org/10.5120/12826-0206>
 19. Mamdani E.H., Assilian S. An experiment in linguistic synthesis with a fuzzy logic controller. Int J Man Mach Stud. 1975; 7(1): 1–13. [http://dx.doi.org/10.1016/s0020-7373\(75\)80002-2](http://dx.doi.org/10.1016/s0020-7373(75)80002-2)
 20. Zhang Y., Ishibuchi H., Wang S. Deep Takagi–Sugeno–Kang fuzzy classifier with shared linguistic fuzzy rules. IEEE Trans Fuzzy Syst. 2018; 26(3): 1535–49. <http://dx.doi.org/10.1109/tfuzz.2017.2729507>
 21. Ekuse Eborn S.L. Fail-fast in the Design Process of an Interactive Voice Response System. Lund, Swedish: The Department of Design Sciences Faculty of Engineering, Lund University Box. 118, 221 00 Lund; 2019.
 22. Naina Gupta D.C. A fail-fast mechanism for authenticated encryption schemas. New Delhi: Indraprastha Institute of Information Technology; 2016.
 23. Jin F., Cao M., Liu J., Martínez L., Chen H. Consistency and trust relationship-driven social network group decision-making method with probabilistic linguistic information. Appl Soft Comput [Internet]. 2021; 103(107170): 107170. <http://dx.doi.org/10.1016/j.asoc.2021.107170>