

Sequence Matching Genetic Algorithm for Square Jigsaw Puzzles

Josef Hynek

► **To cite this version:**

Josef Hynek. Sequence Matching Genetic Algorithm for Square Jigsaw Puzzles. 10th IFIP International Conference on Artificial Intelligence Applications and Innovations (AIAI), Sep 2014, Rhodes, Greece. pp.317-324, 10.1007/978-3-662-44654-6_31 . hal-01391329

HAL Id: hal-01391329

<https://hal.inria.fr/hal-01391329>

Submitted on 3 Nov 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Sequence Matching Genetic Algorithm for Square Jigsaw Puzzles

Josef Hynek

Faculty of Informatics and Management, University of Hradec Králové,
Rokitanskeho 62, 500 03 Hradec Králové, Czech Republic
Josef.Hynek@uhk.cz

Abstract. Our paper presents a new method for solving the rectangle piece jigsaw puzzle problem. The puzzle image is RGB full color and because of uniform shape of the individual pieces the process of puzzle assembly is based on information of the pixel values along the border line of the piece only. We have utilized a genetic algorithm that searches for the optimal piece arrangement using dissimilarity between adjacent pieces as the measure of progress. Unlike the previous attempts to utilize genetic algorithms to solve the problem, we have proposed a new heuristic asexual operator that aims at identification of points of fraction within partially assembled picture, extraction of supposed sequence of correctly joint pieces, and its insertion into a new position in such a way that, if possible, the segment is enlarged. Our approach has been successfully tested and the algorithm is capable of solving puzzles consisting of several hundred pieces.

Keywords: genetic algorithms, image reconstruction, pattern matching, puzzle solving.

1 INTRODUCTION

Solving puzzles is usually viewed as a special example of pattern matching recognition problem. At the first sight it might be viewed as a pleasant and very popular widespread game for players of all ages but we should not forget the complexity of jigsaw puzzles. It is a well-known fact that jigsaw puzzles and their equivalents belong to NP-complete class [1] and thus provide serious obstacles for various attempts to solve them automatically. Moreover, there are many practical applications ranging from image mosaicking up to the reconstruction of archaeological artifacts and assembly of shredded documents [2]. The first jigsaw puzzle was produced in 1760 in London [3] and then the game of puzzle gradually expanded all over the world. There are various types of jigsaw puzzle and an integrated classification terminology is missing. However, there are some basic classification terms that are commonly used and we will mention here at least the most important of them.

The term standard jigsaw puzzle is usually used to describe puzzles created by cutting pictures printed on firm stiff paper into interlocking patterns of pieces. The pieces

have a distinctive geometrical shape that provides an important piece of information that is used together with image information in the process of puzzle assembly. It is necessary to point out that there are also so called apictorial types of puzzle that do not have any picture nor chromatic information and the shape of the pieces is the only clue to solve the problem.

We are interested in square jigsaw puzzles where the individual pieces lack characteristic curvilinear shape because all the pieces have straight borders and uniform rectangular size. In this case pictorial information provides us only information and guidance that is available there. This type of puzzle is sometimes called as edge-matching puzzle [1] in order to emphasize that the adjacent pieces match along their respective edges. These puzzles are very challenging as there is no guarantee that two pieces that fits together should be together and the only way to proof the correctness of individual matches is the completion of the entire solution.

Of course, the simple categorization of different kinds of puzzle above is very superficial and definitely incomplete, but for the sake of this paper we do not need to go further in this direction. For more details concerning different kinds of puzzle and its classification please refer to [3].

The rest of the paper is organized as follows. Section 2 summarizes various approaches to the automatic solution of jigsaw puzzles with the particular focus on utilization of genetic algorithms. Section 3 is the key part of our paper and the newly proposed solution is described there. Our experimental results are briefly described in Section 4 and it is followed by discussion and conclusions.

2 Previous Work

Many attempts to utilize various problem solving techniques were made since 1964 when the first paper describing computer solution of apictorial jigsaw puzzle was published [4]. It has been pointed out that the difficulty encountered when attempting to program a computer to solve jigsaw puzzles relates to three different aspects of the problem: 1) the description of the pieces, 2) the manipulation of the pieces (rotating and matching), and 3) the evaluation of the correct matching of the individual pieces. The complexity of the problem implies that a brute-force approach, which may work for small puzzles, becomes impractical as the number of pieces is increased [4].

It is necessary to emphasize that majority of attempts that dates more than fifteen years back focused on apictorial puzzles. It means that the relevant experiments are concerned with geometric shape information of the puzzle pieces and chromatic (pictorial) information is neglected (see, for example, [5] or [6]). These articles brought up gradually improved methods and techniques for representation of boundaries of puzzle pieces and their subsequent efficient pertaining and correct matching.

Perhaps the most efficient algorithm for automatic solution of jigsaw puzzles was presented by Goldber et al. [7]. It is based on the similar ideas as the already mentioned algorithm described in [5], but it utilizes better and more global matching techniques. In particular, use of fiducial points to align adjacent pieces of puzzle together

with optimization of partial solutions based on the method of global relaxation worked surprisingly well and allowed authors to solve the puzzles consisting of more than two hundred pieces [7].

An obvious way to improve efficiency and robustness of the algorithm is a suitable utilization of additional pieces of information on the top of a distinctive geometric shape of the puzzle pieces. Colours or textures on the boundaries of these pieces could provide the algorithm further clues and direction if such information is available (see [8] or [9]). However, as we have mentioned above, there are also so called edge-matching puzzles where all the pieces have straight borders and uniform rectangular size and pictorial information provides us the only information and guidance that can be used in order to solve it.

Toyama et al. [10] proposed a method for solving the rectangle piece jigsaw puzzle problem using a genetic algorithm. The picture of their puzzle is painted only in black and white and so the pieces of the puzzle are represented as binary images. The assembly of the puzzle is then performed using information of the pixel value on the border line of the adjacent pieces. They utilised the populations of 200 randomly generated individuals representing candidate solutions and employed two genetic operators (2-point crossover and self-crossover) to produce offspring and run the evolutionary process. They reported that the method described there correctly assembled all pieces in the 8x8-piece puzzle.

This team of researchers collaborated on another interesting paper [11] and developed an improved method for solving the rectangle piece jigsaw puzzle assembly problem, but no evolutionary algorithm is used in this process any more. The assembly of the puzzle is once again performed only using information of the pixel value on the border line of the pieces and this time a puzzle image is RGB full colour. Pieces are connected by a matching function between two pieces and a simple method connects together a single piece to a block that is defined as a group of already connected pieces. According their results the proposed method correctly assembled all pieces in 16x12-piece puzzles.

Alajlan [12] experimented with gray picture puzzles and he used dynamic programming to facilitate non-rigid alignment of border pixels for local matching of the puzzle pieces. Moreover, instead of the classical best-first search, his algorithm simultaneously positioned the neighbours of a puzzle pieces during the search using the so-called Hungarian procedure. This procedure begins with a starting piece of the puzzle and then locates four adjacent pieces of puzzle to the initial piece in the way the sum of border distances is minimal. In order to make the algorithm robust, every puzzle piece was considered as starting piece at various starting locations. Experiments using several images demonstrated that the proposed algorithm correctly assembled puzzles up to 8x8 pieces. However, the author of this article acknowledged that its performance deteriorates as the number of pieces exceeds 64.

Finally, we would like to mention one completely different approach that was described by Gindre et al. [3]. Their algorithm is a part of the development of Intelligent Robotic System that solves an unknown jigsaw puzzle. The system uses pattern recognition techniques as edge and feature detection in conjunction with genetic algorithms. The novelty of this as approach is based on a completely different encoding

scheme. The candidate solutions are represented by relevant graphs describing the interconnections between puzzle pieces and that is why the corresponding chromosomes are defined by the relevant adjacency matrix.

The common limitation of the above presented approaches is that just small scale puzzles could be tackled and solved. That is the fact that fostered our interest in this area and based on a detailed analysis of the methods that were already tested we have designed a hybrid genetic algorithm for jigsaw puzzle problem.

3 Problem Definition and Representation

We have shown that there are various types of jigsaw puzzles and therefore it is necessary to start with its definition. We have decided to experiment with rectangular pictorial puzzle as it is described in [10]. Therefore, we assume that it consists of $N \times M$ puzzle pieces that do not rotate and the assembly of the puzzle is performed by using the pixel values on the border lines of the individual pieces.

Whereas Toyama worked with black and white pictures only where the pixel values are 0 and 1 respectively, we have decided to use color 24-bit RGB images. Despite the fact that the comparison of border lines is very easy and straightforward, there is no guarantee that there is an absolute correspondence between border lines of the neighboring pieces. It depends on the particular picture and irregularities along the border make the problem difficult. It is clear that algorithm could be even misled in the situation when wrong piece of the puzzle is regarded as more suitable from the point of borders matching than the right one. Hence, it is evident that the problem cannot be solved by the simple local piece matching. The example of the puzzle used in our experiments is shown on Fig. 1.

Having defined the problem we can decide what kind of representation of partial (or candidate) solutions would be suitable for our algorithm. We have realized that graph representation used by Gindre et al. [3] is rather inconvenient for genetic operators and the pertinent manipulations and that is why we have chosen simple $N \times M$ matrix for each individual and each cell $c_{i,j}$ corresponds to one piece of puzzle. The matrix depicts the relevant arrangement of puzzle pieces in a very straightforward manner that the left/right/up/down adjacency in the matrix means the same interrelationship between the relevant puzzle pieces. This representation is very convenient for fitness evaluation as well as for utilization of genetic operators.

Each piece of the puzzle is represented by matrix $L \times L \times 3$, where L is the width and length of the piece (i.e. the number pixels along the border). The fitness function of our algorithm is based on two measures of dissimilarity. Horizontal dissimilarity between two pieces x_i and x_j , where x_j is placed to the right of x_i is defined as

$$D_h(x_i, x_j) = \sqrt{\sum_{l=1}^L \sum_{k=1}^3 (x_i(l, L, c) - x_j(l, 1, c))^2}, \quad (1)$$

where c stands for intensity of red, green, and blue color and each value is integer from 0 to 256. Likewise, vertical dissimilarity between two pieces x_i and x_j , where x_i is placed on the top of x_j is defined as

$$D_v(x_i, x_j) = \sqrt{\sum_{l=1}^L \sum_{k=1}^3 (x_i(L, l, c) - x_j(1, l, c))^2}, \quad (2)$$

where square difference between pixels of the last row of x_i and the first row of x_j is computed. These measures of dissimilarity will be calculated many times during the process of evaluation of thousands of individuals and therefore it is advantageous to create a lookup table covering all the relevant values at the beginning of run.

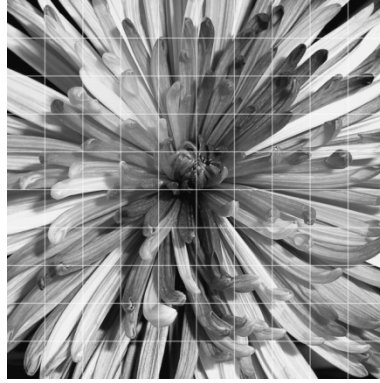


Fig. 1. Example of the puzzle used for the algorithm testing (10x10-piece puzzle).

The fitness function of each individual is calculated as the reverse value of the sums of horizontal and vertical differences of the relevant puzzle pieces. Providing that individual a is represented by $N \times M$ matrix C as we have described it above, then its fitness is defined as

$$f(a) = \frac{1}{1 + \sum_{i=1}^N \sum_{j=1}^{M-1} D_h(c_{i,j}, c_{i,j+1}) + \sum_{i=1}^{N-1} \sum_{j=1}^M D_v(c_{i,j}, c_{i+1,j})}. \quad (3)$$

It is evident that the important feature of fitness function $f(a)$ is that individual with smaller sum of dissimilarities is fitter and thus will have a higher chance of being selected. We have employed standard biased roulette wheel selection method together with an elitist approach when the best individuals (10%) from current population are copied without any change to the newly created population.

The novelty of our approach is based on the utilization of problem specific operator. Besides the common partially-matched crossover (PMX) that assures legitimacy of created offspring we have employed a new operator that helps to preserve the already created segments of the picture being assembled. The main drawback of PMX

is that crossover points are selected randomly and therefore it is easy to split a sequence of correctly assembled pieces. Nevertheless, it provides us necessary variability, exchanges pieces of information between different individuals and thus it is important for the whole process of evolution.

We have experimented with asexual heuristic operator that is designed to enlarge the already created segments of the picture. These segments are not only scattered over the whole candidate solution but they are usually located at incorrect locations. We are unable to assign them to their correct place within the picture as the algorithm does not use this piece of information, nevertheless we can try to connect them with other segments believing that larger segments will finally settle at the right positions.

First of all, we identify likely points of fraction where the horizontal measure of dissimilarity is relatively large. For $N \times M$ pieces of puzzle we identify N points of fraction with the greatest value of D_h function using (1). We select one of these points of fraction randomly using the biased roulette wheel selection mechanism so the point of fraction with the greatest dissimilarity has the highest probability to be selected. When the point of fraction has been determined we follow the horizontal direction till the next point of fraction is identified and the sequence of pieces between these two points of fraction constitutes the segment. An example of distribution of points of fraction as well as the relevant segments is on Fig. 2.

The final phase of this procedure is an attempt to allocate the segment into a new position within the picture using D_h function value (1) for the very first or very last piece of the segment. Moreover, we must not forget that albeit the point of fraction cannot occur at the beginning or at the end of rows (it results from the definition of D_h), the right place of segment could be at the beginning of the row and such a placement will be indicated by D_v according to (2).

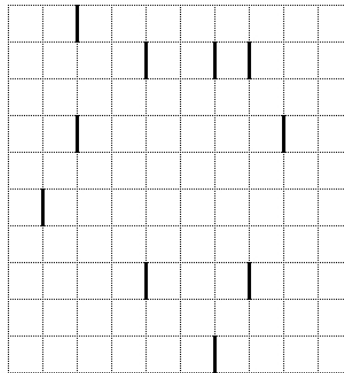


Fig. 2. Points of fraction and prospective segments (10x10-piece puzzle).

Of course, there is no guarantee that the sequence selected is a real and correct segment of the picture. However, as the points of fraction present very likely some faults within current arrangement of pieces, there is a chance to improve the overall composition.

It is clear that the above described asexual operator is efficient especially in latter phases of evolution when suitable building blocks (segments) exist within relevant individuals. That is why we have started the evolution process with 1000 individuals and carried it on for 50 generations. Then we employed decimation operation and discarded most of the population. Decimation is a secondary genetic operation that is normally performed at the beginning of a run and it is used to increase the proportion of fitter individuals in the population. It has been shown [13] that decimation provides improvement in performance that justifies the additional computation at the beginning of the run. Hence, taking advantage of the decimation operation only 100 individuals survived (we made sure that these individuals are unique in order to provide enough diversity) and then we start using the newly devised operator.

4 Experiments

We have experimented with three different pictures that were divided into $N \times N$ pieces. The size of the pieces was 80×80 pixels and so the size of the picture changed accordingly the number of pieces taken into account. The algorithm run for 200 generations, the first 50 generations using PMX operator with $p_c=0.8$ and simple mutation (changing mutually places of two randomly selected pieces) with $p_m=0.05$. For the next 150 generations the fraction point operator was applied to 50% of population whereas frequency of PMX slowed down to $p_c=0.3$. Taking into account that the best individuals (10%) are copied automatically to the new population, the size of it remains constant for the rest of the evolution process.

We ran our algorithm 50 times for each image and the results are given in Table 1. We can see that correct solution has been found for all the cases that were tested and the efficiency of the algorithm clearly deteriorates with increasing number of pieces. We have not included puzzles with more than 625 pieces (25×25) into our results table as we were unable to get a correct result within the given size of population and limited number of generations.

Table 1. Experimental Results.

No. of pieces	Test 1	Test 2	Test 3
15x15	100%	94%	100%
20x20	92%	74%	86%
25x25	56%	42%	52%

5 Discussion and Limitations

Our results show that the method using points of fraction is promising. There are still some opportunities to improve the way of points of fraction location as well as the way of matching the individual segments. Nevertheless, using this approach we were able to assemble puzzles of larger sizes than our predecessors.

In several cases our algorithm got stuck when it reached a partial solution in a form of horizontally cut picture that is glued together in a completely wrong way. We have analyzed this malfunction and we have realized that the algorithm was simply misled by a low level of dissimilarity along the cutting line. Because our algorithm does not make any use of the correct order and/or location of the individual pieces within the original picture, there is no way to avoid this kind of fault.

6 Conclusions

We have described the new algorithm for solving the rectangle piece color puzzle problem. Genetic algorithm has been augmented by new heuristic asexual operator that aims at identification of points of fraction within partially assembled picture, extraction of supposed sequence of correctly joint pieces, and its insertion into a new position in such a way that, if possible, the segment is enlarged. This method promotes the gradual enlargement of correctly connected blocks and the search for an optimal position of the block within the frame of the assembled picture is enabled and supported at the same time too. Our approach has been successfully tested and the algorithm is capable of solving puzzles consisting of several hundred pieces.

7 References

1. Demaine, E. D., Demaine, M. L.: Jigsaw Puzzles, Edge Matching, and Polyomino Packing: Connections and Complexity. *Graphs and Combinatorics*, vol. 23, pp. 195–208 (2007).
2. Richter, F., Ries, C. X., Cebon, N., Lienhart, R.: Learning to Reassemble Shredded Documents. *IEEE Transactions on Multimedia*, vol. 15, no. 3, pp. 582 – 593 (2013).
3. Gindre, F., Trejo Pizzo, D. A., Barrera, G., Lopez De Luise, M. D.: A Criterion-based Genetic Algorithm Solution to the Jigsaw Puzzle NP-Complete Problem. *Lecture Notes in Engineering and Computer Science*, vol. 2186, no. 1, pp. 367-372 (2010).
4. Freeman, H., Garder, L.: Apictorial Jigsaw Puzzles: The Computer Solution of a Problem in Pattern Recognition. *IEEE Transactions on Electronic Computers*, vol. EC-13, no. 2, pp.118-127 (1964).
5. Wolfson, H., Schonberg, E., Kalvin, A., Lamdan, Y.: Solving jigsaw puzzles by computer. *Annals of Operation Research*, vol. 12, no. 1-4, pp. 51-64 (1988).
6. Webster, R. W., LaFollette, P. S., Stafford, R. L.: Isthmus critical points for solving jigsaw puzzles in computer vision. *IEEE Transactions on Systems, Man and Cybernetics*, vol.21, no.5, pp.1271-1278 (1991).
7. Goldberg, D., Malon, C., Bern, M.: A global approach to automatic solution of jigsaw puzzles. *Computational Geometry*, vol. 28, no. 2-3, pp. 165-174 (2004).
8. Chung, M. G., Fleck, M., Forsyth, D.: Jigsaw puzzle solver using shape and color. In *Proc. of 4th International Conference on Signal Processing*, vol.2, pp. 877-880 (1998).
9. Makridis, M., Papamarkos, N., Chamzas, C.: An innovative algorithm for solving jigsaw puzzles using geometrical and color features. In *Proc. of 10th Iberoamerican Congress on Pattern Recognition*, Havana, Cuba, *Lecture Notes in Computer Science*, vol. 3773, pp. 966-976 (2005).

10. Toyama, F., Fujiki, Y., Shoji, K., Miyamichi, J.: Assembly of puzzles using a genetic algorithm. In Proc. of 16th International Conference on Pattern Recognition, Quebec, Canada, vol.4, pp.389-392 (2002).
11. Murakami, T., Toyama, F., Shoji, K., Miyamichi, J.: Assembly of puzzles by connecting between blocks. In Proc. of 19th International Conference on Pattern Recognition, Tampa, FL, pp. 1-4, (2008).
12. Alajlan, N.: Solving Square Jigsaw Puzzles Using Dynamic Programming and the Hungarian Procedure. American Journal of Applied Sciences, vol. 6, no. 11, pp. 1941-1947 (2009).
13. Nanduri, D. T., Ciesielski, V.: Comparison of the effectiveness of decimation and automatically defined functions. In Proc. of the 9th Int. Conference on Knowledge-Based Intelligent Information and Engineering Systems, Lecture Notes in Computer Science, vol. 3683, pp. 540-546 (2005).