ACTA SCIENTIARUM POLONORUM *Geodesia et Descriptio Terrarum 4(2) 2005, 15-23*

A GENETIC ALGORITHM HOW TO SOLVE A PUZZLE AND ITS USING IN CARTOGRAPHY

Dalibor Bartoněk

University of Technology of Brno

Abstract: Genetic algorithms represent an up-to-date method of process optimization, where other solutions have failed or haven't given any satisfactory results. One of these processes is puzzle solving, where fragments have to be placed into the defined shape in such a way so that no fragment should mutually overlay and the whole shape area will be filled with all of these fragments. A genetic algorithm solving this task including an exact formulation and a definition of the initial conditions based on cluster analysis has been described in this paper. The algorithm efficiency will be tested in diploma works in Institute of Geodesy, Faculty of Civil Engineering, University of Technology, Brno. The results will be used in the application for cartograms creation.

Key words: Genetic algorithm, cluster analysis, shape, fragments, shape boundary, string code, optimization, fitness function, cartograms

1. INTRODUCTION

Puzzle belongs to the combinatorial tasks, the solving of which is the most difficult one. If you use the computer, the process is both storage and time consuming. Example of this is the Eternity problem – see Figure 1.

Eternity is a puzzle introduced by Christopher Monckton and consists of 209 fragments, each of which is a 12-polydrafter (i.e., a compound of 30-60-90 triangles). The puzzle was presented in Britain in June 1999. The goal of the puzzle is to arrange the fragments in the shape of a slightly non-regular dodecagon. A one-million-pound award was offered for the first solution to the puzzle, which was found by Alex Selby and Oliver Riordan in May 15, 2000. Their solution is illustrated in Fig. 1 (puzzle pieces © 1999, Christopher Monckton). The second solution was subsequently found by Guenter Stertenbrink. Interestingly, neither of these solutions matches the six clues given by the puzzle's creator Christopher Monckton for his solution, which remains unknown. Para-

Corresponding author – Adres do korespondencji: Dalibor Bartoněk, Institute of Geodesy, University of Technology of Brno, Veveří 95, 662 37 Brno, Czech Republic, e-mail: bartonek.d@fce.vutbr.cz.

doxically, a *harder* puzzle is possible using a *smaller* number (about 140 fragments) of 12-polydrafters. This fact accounts for the reason that the discoverers of the two known solutions deliberately discarded Monckton's clues in order to open the way for possibly easier solutions.

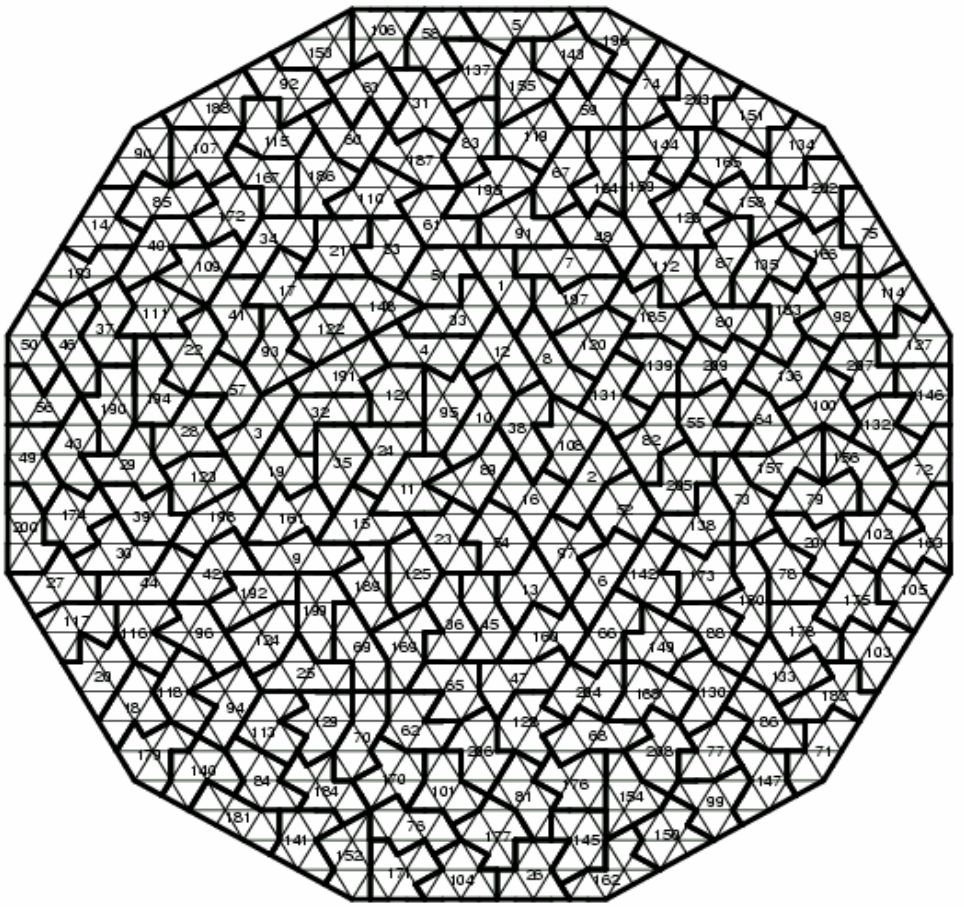


Fig. 1. Eternity problem
Rys. 1. Problem „Eternity”

2. TASK FORMULATION

The task formulation is very simple. There is a set of fragments and a certain shape (area). The goal of the solution consists in arrangement of the fragments into the shape in such a way, so that none of the fragments should mutually overlay. As an example of this problem you can see tangram on Figure 2. The tangram is a combination of plane polygonal fragments in such a way, so that the edges of the polygons can be coincident. There are 13 convex tangrams (where a "convex tangram" is a set of tangram fragments

arranged into a convex polygon). Curiously, a tangram with colored squares is the logo for the utility company *Illinois Power*. When solving the puzzle problem on computer, the shape and every fragment have to be suitably described. One of these methods uses string codes.

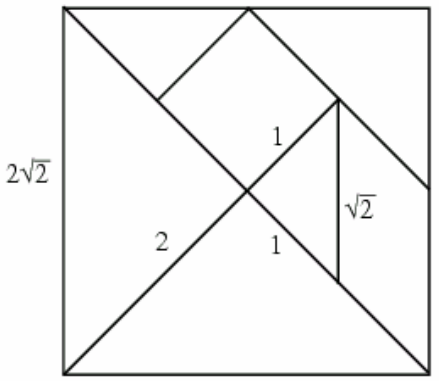
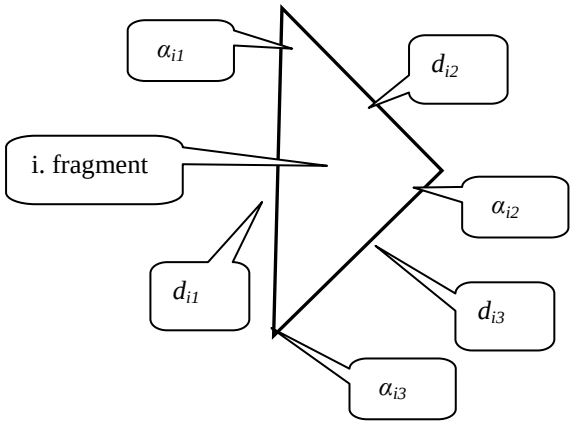


Fig. 2. Tangram
Rys. 2. Tangram

String code is shown on Figure 3. Every shape (whole area) and each edge of the shape or fragment is denoted by integer number. The edge configuration of *i.* shape/fragment is described by sequence $d_{ij}\ \alpha_{ij}$, where d_{ij} is length of *j*. edge and α_{ij} [º] is angle between edges d_{ij} and d_{ij+1}. Direction of sequence must be with every fragment the same e. g. clockwise. String code for the triangle fragment of tangram is shown on Figure 3. The string codes are independent on the geometric transformation.



String codeof i. fragment:
$d_{i1,}\ \alpha_{i1,}\ d_{i2,}\ \alpha_{i2,}\ d_{i3,}\ \alpha_{i3}$ =
√2,45º,1,90 º,1,45 º

Fig. 3. Example of the string code
Rys. 3 Przykład kodu łańcuchowego

3. USED METHOD OF SOLUTION

The used method is based on the theory of cluster analysis. The idea consists in hierarchic arranged clusters created during the iteration process. Every fragment is assigned with a certain cluster according to similarity with others fragments, which are stored in the same cluster. The value of the fragment similarity is computed on the common length of string codes (length of coincident string codes) between the processed (actual) fragment and new created (current) shape boundary.

Disadvantage of this method is great number of combinations how split *n* fragments into *k* clusters, given by Stirligs number of second order:

$$G^{(k)} = \frac{1}{k}\sum_{i=1}^{k}(-1)^{k-1}\binom{k}{i}i^{n} \tag{1}$$

Therefore this process must be optimized. In the cluster theory there are most heuristic methods e. g. McQueen's or Forgy's algorithm [Hebák, Hustopecký 1987]. Let us try to solve this problem by genetic algorithm. The fragments splitting into clusters will be done according the fitness function:

$$F_i = max\ F_{i1},\ldots,F_{im} \tag{2}$$

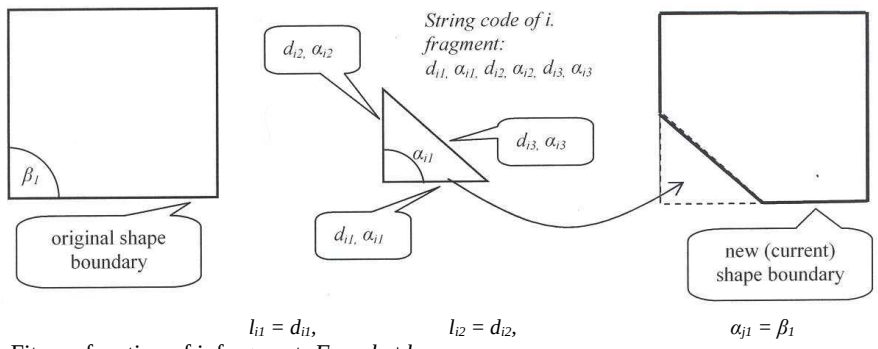
where *m* is number of possible coincident string codes between *i*. fragment and shape boundary and F_{ij} is total length of coincident boundaries (sequence of edges and angles) between *i.* fragment and shape boundary given by string codes:

$$F_{ij} = \sum_{j=1}^{n} l_{ij} \tag{3}$$

where *n* is number of edges in *i*. fragment, l_{ij} is length of *j*. edge of *i*. fragment computed by formula:

$$l_{ij} = \begin{cases} d_{ij} \text{ if } (d_{ij} <= e_k) \text{ and } (\alpha_{ij} = \beta_k) \text{ in string code of } i. \text{ fragment.} \\ 0 \text{ - in other cases.} \end{cases} \tag{4}$$

where e_k is coincident edge of shape boundary related to d_{ij} and α_{ij} is angle between d_{ij-1} and d_{ij} and β_k is angle between e_{k-1} and e_k. Example of computing l_{ij} is shown on Figure 4.



Fitness function of i. fragment: $F_{i1} = l_{i1}+l_{i2}$.

Fig. 4. Evaluation of fitness function of *i.* Fragment
Rys. 4. Ocena funkcji dopasowania fragmentu i

4. DESCRIPTION OF THE ALGORITHM

The main idea consists in iteration process of 2 steps:

1. A consequent selection of suitable fragments in such a way, that their boundaries were coincident right with the whole shape boundary – see Figure 5.
2. If all fragments are used, the algorithm is over, otherwise the previous process creates a new shape boundary and iteration follows from the step No 1.
3.

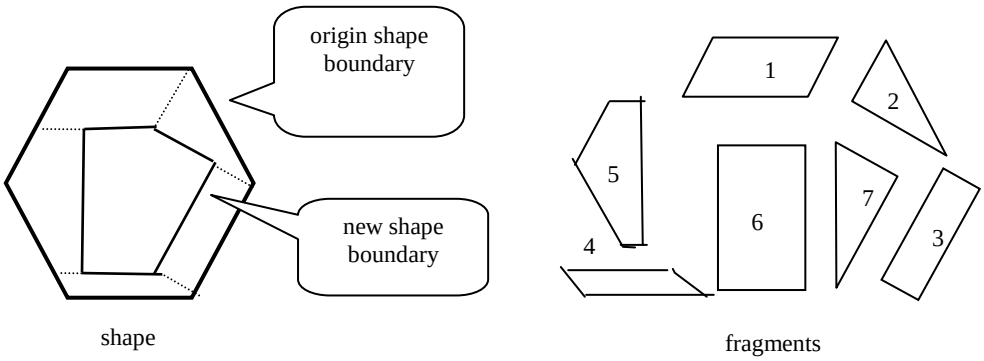


Fig. 5. The main idea of the algorithm
Rys. 5. Idea algorytmu

A selection of the suitable fragment in step 1 is a very complicated problem, because there are lots of possibilities. One of the methods how solve this problem is using of genetic algorithm. Fitness function is used as a selection criterion – see equation (2).

Algorithm:

1. (Initialization). Mark all fragments as unused, evaluate string code for origin shape boundary.
2. Affinity evaluation of all free (unused) fragments in relationship to the current shape boundary using the string codes according to the formula (3). For each *i.* fragment create a list of functions F_i (3) ordered according to their values.
3. Select current fragment from the list for the first time with maximum value of affinity according to the fitness function – equation (2).
4. Applicate previous step on newly created shape boundary so that two next adjacent fragments can be coincident both with shape and current fragment boundary – see figure 6.
5. If the selected fragment is suitable, create a new shape boundary and selected fragment with his both adjacent fragments mark as used.
6. If there are no adjacent fragments of this type in previous step, you must try
   a) to select another string value in step 3 in the ordered list of the same fragment (in fact it concerns rotation of this fragment to be coincident with another part of shape boundary),
   b) to select another fragment in step 3.
7. If the whole shape boundary is covered with the selected fragments, go to the step 2. If there are no free (unused) fragments, the algorithm is over.

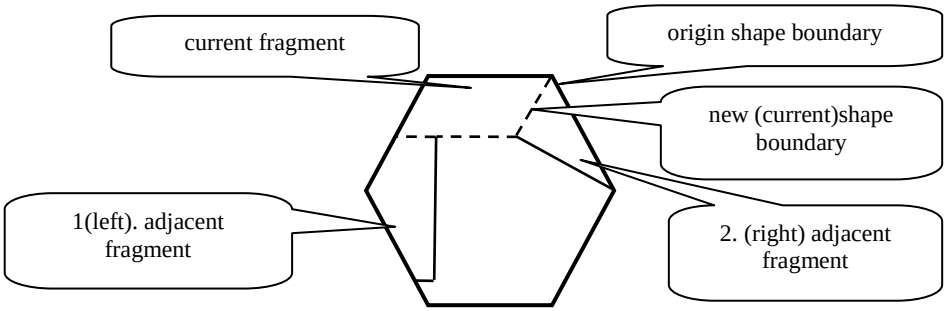


Fig. 6. Fundamental selection of next fragments adjacent to the previous fragment
Rys. 6. Zasada selekcji następnych fragmentów dla fragmentu bieżącego

5. EVALUATION OF THE ALGORITHM

For algorithm assessment we usually use the function $O(n)$ which expresses its time durability depending on extent of input data *n*. As, in our case, the problem concerns nondeterministic algorithm where every step can differ in difficulty. Let's set maximum time durability with the algorithm:

$$O(n)_{max} = T_{max} = \sum_{i-1}^{k} \left(T_{L_i} + T_{SEL_i} \right) \tag{5}$$

k – is number of iteration steps,
T_{L_i} – is time of ordered string codes list creation in *i.* step of the algorithm,
T_{SEL_i} – is time of segments selection list creation in *i.* step of the algorithm.

Let's try to estimate the time of ordered string codes list creation:

$$T_{L_i} = n_{ASC_i} n_{SH_i} t_c + t_{ORD_i} \tag{6}$$

n_{ASC_i} – is number of string codes of all segments in *i.* step of the algorithm (see Fig. 5):
n_{SH_i} – is number of string codes of new shape boundary in *i.* step of the algorithm (see Fig. 5),
t_c – is time duration of comparison of two different segments,
t_{ORD_i} – is time needed to order string codes of all segments in *i.* step of the algorithm.
The value of n_{ASC_i} we can get from following formula:

$$n_{ASC_i} = \sum_{j=1}^{n_{SEGi}} n_{SC_j} \tag{7}$$

n_{SEG_i} – is number of segment in in *i.* step of the algorithm and

n_{SC_j} – is number of string codes of individual segments in *i.* step of the algorithm.

In [Wirth 1989] was derived formula for one of the best Quick sort method that can be used:

$$t_{ORD_i} = 28 n_{ASC_i} + 22 \log_2 n_{ASC_i} + 38 n_{ASC_i} + 7 \tag{8}$$

The T_{SEL_i} can be computed according to the formula:

$$T_{SEL_i} = n_{ASC_i} n_{SH_i} t_c \tag{9}$$

When we express the formula (5) by all formulas (6) – (9) we get:

$$T_{max} = \sum_{i=1}^{k} \left(2 n_{SH_i} t_c \sum_{j=1}^{n_{SEG_i}} n_{SC_j} + 28 \sum_{j=1}^{n_{SEG_i}} n_{SC_j} + 22 \log_2 \sum_{j=1}^{n_{SEG_i}} n_{SC_j} + 38 \sum_{j=1}^{n_{SEG_i}} n_{SC_j} + 7 \right) \tag{10}$$

If we suppose, that the genetic algorithm would be successfully in 50% (test according to the fitness function) we can correct the last formula as follow:

$$T_{max} = \sum_{i=1}^{k} \left(n_{SH_i} t_c \sum_{j=1}^{n_{SEG_i}} n_{SC_j} + 28 \sum_{j=1}^{n_{SEG_i}} n_{SC_j} + 22 \log_2 \sum_{j=1}^{n_{SEG_i}} n_{SC_j} + 38 \sum_{j=1}^{n_{SEG_i}} n_{SC_j} + 7 \right) \tag{11}$$

From the formula (11) it is obvious that the complete time durability of the algorithm depends not only on the number of segments but also on their variability e. t. the number of string codes. As with the increasing number of iteration steps the number of sting codes considerably decreases we suppose that this algorithm will converge.

6. USAGE OF THE ALGORITHM IN APPLICATION

The algorithm was used in application for cartograms creation. In the first phase only 9 cadastral boundaries in Znojmo district were tested. These cadastral areas were from neighbouring districts, therefore theirs coordinates came from different sources. The output in shown on Figure 7. In order to test the efficiency of the algorithm we would need a greater number of areas. This is a task in the future e.g. as theses for diploma and PhD. works.

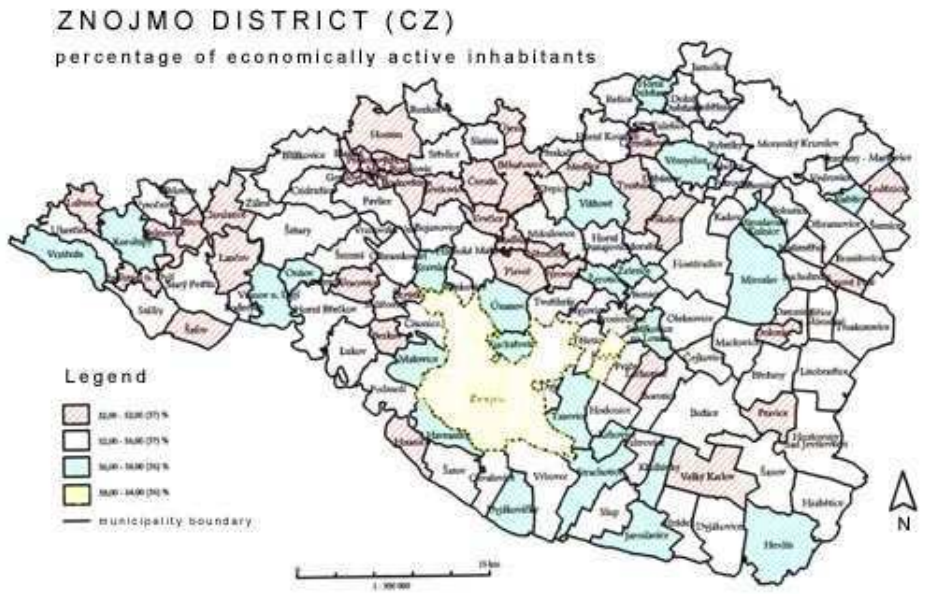


Fig. 7. Percentage of economic active inhabitants in Znojmo district (CZ)
Rys. 7. Udział procentowy ekonomicznie aktywnych mieszkańców powiatu Znojmo (Czechy)

7. CONCLUSION

The advantage of the above described algorithm unlike the others consists in selection of only those fragments having the suitable evaluation (the fitness function value). This method reduces the computing time. Substance of this algorithm consists in consequent selection of fragments to engage the current shape boundary. The selection is controlled by fitness function value according to the similarity among the current fragment, its left and right adjacent fragments and current shape boundary. If all the fragments were selected at random to fit each other and covered the total shape area, the number of possible combinations would be so enormous, that the iteration process may be diverged. Avoiding this situation it would be the most important contribution of presented genetic algorithms.

Algorithm presented in this paper will be verified in diploma works in Institute of Geodesy, Faculty of Civil Engineering, University of Technology, Brno. The results will be used in the application for cartograms creation. Most of the cadastral boundaries are in various cartographic coordinates and the transformation key is unknown, therefore to create the municipal boundaries out of these cadastral boundaries is commonly very difficult. One of possible solutions is usage of this algorithm.

REFERENCES

Goldberg D. E., 1989. Genetic Algorithms in Search, Optimisation and Machine Learning. Addisson – Wesley.

Ošmera P., 2001. Genetic algorithms and theirs application. Habilitation thesis. Faculty of Mechanical Engineering, University of Technology, Brno.

http//www.mathworld.com.

Hebák P., Hustopecký J. 1987. Vícerozměrné statistické metody s aplikacemi. SNTL Praha, 456 s.

Wirth N. 1989. Algoritmy a štruktúry údajov. ALFA Bratislava, ISBN 80-05-00153-3.

ALGORYTM GENETYCZNY DO SKŁADANIA POWIERZCHNI Z FRAGMENTÓW I JEGO ZASTOSOWANIA W KARTOGRAFII

Streszczenie. Algorytmy genetyczne reprezentują nowoczesne metody optymalizacji procesów, dla których inne rozwiązania zawiodły lub nie dały satysfakcjonujących rezultatów. Jednym z takich procesów jest rozwiązywanie układanek – puzli, w których fragmenty muszą być wstawione w zdefiniowany kształt w ten sposób, aby żadne się nawzajem nie nakładały, a kształt zawierał wszystkie zadane fragmenty. Praca niniejsza zawiera opis algorytmu genetycznego rozwiązującego takie zadanie wraz ze ścisłą formułą rozwiązania oraz definicją warunków początkowych, bazującą na analizie skupień. Skuteczność algorytmu będzie testowana w pracy dyplomowej w Instytucie Geodezji na Wydziale Budownictwa, Politechniki w Brnie. Rezultaty zostaną wykorzystane przy tworzeniu kartogramów.

Słowa kluczowe: Algorytm genetyczny, analiza skupień, kształt, fragmenty, granice figur, kod łańcuchowy, optymalizacja, funkcja dopasowania, kartogramy

Accepted for print – Zaakceptowano do druku: 28.12.2005