# The Combination of Morphological and Evolutionary Algorithms for Graph-based Pathfinding on Maps with Complex Topologies

I. A. Savostin[1], A. O. Trubakov[1]
iansav7@gmail.com|trubakovao@gmail.com
[1]Bryansk state technical university; Bryansk, Russia

*One of the difficult problems to solve has always been and still remains the problem of finding a path either in a graphic chart or a graphic maze of large size. The main problem is that traditional algorithms require a lot of time due to combinatorial complexity. At the same time, both classical algorithms based on the search of variants (such as Dijkstra's algorithm, A*, ARA*, D* lite), and stochastic algorithms (ant algorithm, genetic), alongside with algorithms based on morphology (wave) are not always able to achieve the goal. The article proposes a new modification of the path-finding algorithm, which is a hybrid of the following: the morphological operations on graphic chart approach and genetic algorithm having a useful property of elasticity in time. The experiments (both synthetic and real data) have shown the feasibility of the proposed idea and its comparison with the most commonly used algorithms of contemporaneity.*

***Keywords:*** *path search, genetic algorithms, morphological transformations of images.*

## 1. Introduction

Finding a path in the maze has been and remained rather a complex computational task [10]. The solution of this problem is very important for different fields of science. For example, when driving autonomous vehicles, more often than not it is critical to construct a path in conditions of large map and limited resources [6]. Besides, this algorithm can be used in automation theory, graph-based image segmentation, transport problems and other fields. At the same time it is worth noting that very often the source data is a certain image that needs to be processed, formalized and only after that finding of the best route should be reasonable. Today, there are many problem-solving approaches. For example, in [3] we consider a variant-based approach-A *, which is a classical algorithm that uses heuristics to speed up the search. This approach fits for problems of small dimension with a topology that is advantageous from the heuristic point of view. However, it has a number of disadvantages: with large size maps the algorithm operation time increases significantly. Besides it, this algorithm implies very unfavorable topologies, in which time increases extremely strongly due to “errors” of heuristics, which prevents the fulfilment of objectives [10]. The paper [8] proposes algorithm ARA*, which allows varying the time of the algorithm with respect to the quality of the path, however, like other algorithms based on brute force and heuristics (for example, D * lite [7], the most widely used in practice today), shares the common disadvantages of the class. Another class of algorithms is stochastic-algorithms based on working with random and pseudo-random variables. The ant algorithm [4] copes well with the task of finding a path, however, has a propensity to falling into local maxima comparing to the genetic algorithm [11], which is partially deprived of it, but in its turn, has some problems with convergence in case of a large genotype length and sequence nature of the path [12]. Algorithms based on morphology (wave [2], filler-algorithms [9]), as a rule, are unable of both: effective reusing of the previous runs results and map formalizing. However, this class is well suited for cases where the source data is given graphically, which are the subject of this work. Therefore, despite the abundance of approaches, this problem cannot be considered to have been completely solved. Most modern approaches do well with small-sized maps, but their performance falls heavily with the number of forked roads, which can be critical for autonomous mobile objects having limited resources. In such situations, a time-elastic algorithm that is able to find a good enough solution in a short period of time and then, if it is necessary to make it specific, can be considered acceptable.

## 2. Statement of the problem

In this paper, we consider the problem of finding the optimal route of movement on the map, given graphically. Let there be given a map as a matrix:

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix}, \qquad (1)$$

where $a_{ij}$ = {*wall*, *free*, *start*, *finish*}; *n* – image width; *m* – image height.

It is required to find a path *P* of length *l* such that:

$$P = \{k_1, k_2, \dots, k_n\}, l \rightarrow min,$$

where $k_i$ = {*left*, *right*, *up*, *down*}; *n* – number of turns; *l* – geometric path length.

Graphical representation, complex topology, and large map size result in the following problems, in the context of autonomous robot path-finding algorithms:

As a rule, autonomous robots are used to work in conditions that are dangerous for human life and health, which often implies danger also for the robot itself. When the values of environmental parameters are beyond the permissible values for the operation of the robot, it is necessary to plan an evacuation route as quickly as possible. Accordingly, the algorithm must have the property of elasticity in time that is the ability to vary the location time, even if it decreases the quality of the discovered path.

Graphical representation means a checkered field having marks and implying the possibility of movement through the checker squares. The problem is that each free square is a decision point concerning the direction of further movement, which greatly increases combinatorial complexity in the case of large free spaces.

The non-formalized representation imposes restrictions on both: the use of graph-based algorithms and the reuse of the previous algorithm runs results: since the map is an image and you can just use methods based on morphology, but on large maps in case of reruns, they are much slower than graph-based algorithms.

A large number of nodes of the graph is a classical problem of algorithms based on the search of variants, since combinatorial, and as a consequence, computational complexity increases, at best, in proportion to the number of nodes. The use of heuristics based algorithms improve, in general, the efficiency of algorithms based on the enumeration of possibilities, however, in case of unfavorable for heuristics topology of the maze, these algorithms work extremely inefficiently.

To solve these problems, this paper proposes a hybrid algorithm based on morphological operations in image maps for pre-optimization, formalization of the map to the graph-based one as a transition table of a finite-state automation with a preliminary estimate of the nodes and calculation of arcs weights, and the application of a modified genetic algorithm to search the paths in the graph.

## 3. Use of morphological operations for preliminary optimization of the map

For preliminary optimization of the map it has been decided to use morphological transformations [5]. The key-note of morphology is focused on convolution kernel run over the image in order to transform it according to the given rules.

Suppose $A$ is the map image (1), $W$ is the convolution kernel, in such case:

$$W = \begin{pmatrix} w_{(i-1,j-1)} & w_{(i-1,j)} & w_{i-1,j+1)} \\ w_{(i,j-1)} & w_{(i,j)} & w_{(i,j+1)} \\ w_{(i+1,j-1)} & w_{(i+1,j)} & w_{(i+1,j+1)} \end{pmatrix},$$

where *w* = {*wall, free, no matter*}; *i, j* are coordinates relative to the center point.

Suppose $R(A, W_i)$ is a convolution operator, so:

$$R(A, W_i) = \begin{cases} \text{wall, if } (\sum_{i=1}^{n} w_i \& a_i = 9 \\ free, if\ \sum_{i=1}^{n} w_i \& a_i < 9 \end{cases},$$

where $w \in W, a \in A$.

In this case, standard operations such as erosion, dilation, make-and-break are ill-adapted for solving the problem. Therefore, to optimize the maze, we’ve developed our own kernel packages, which are created to fill with “inefficient path” labels. Their main task is to reduce the number of paths and branches that can be discarded in advance and thereby downscale the dimension of the problem. The main kernels were chosen as follows: elimination of nonconvex areas, going across which is not advantageous for geometric reasons (Fig. 1). This allows us to eliminate from consideration those paths that will be obviously longer (for example, diagonally).

There is an optimization of both: rectangular areas with multiple inputs and cubes of 2x2 square (bottom row in Fig. 1-3).

In all figures, the black color is the indicator of the presence of obstacles, light gray presents free spaces, dark gray indicates that the state of the square does not matter in this convolution operation. Mathematically, the kernel is expressed by the following convolution matrix (for example, for the kernel in Fig. 1, in the upper left corner):

$$W = \begin{pmatrix} no\ matter & wall & no\ matter \\ wall & free & no\ matter \\ wall & no\ matter & free \end{pmatrix}.$$

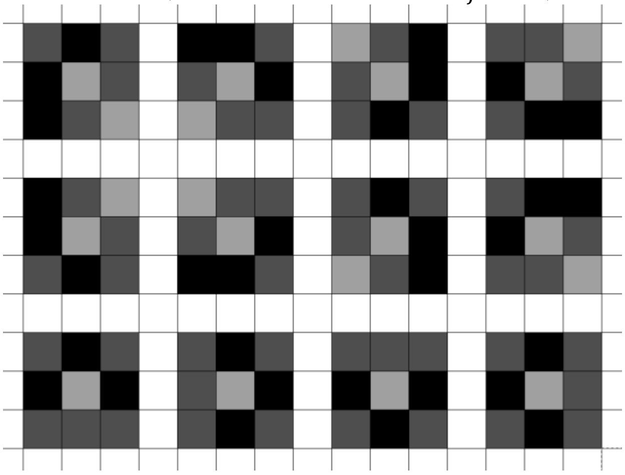

**Fig1.** Convolution kernels to fill non-convex regions

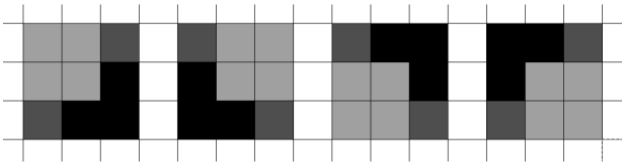

**Fig2.** Convolution kernels to remove the 2x2 cubes

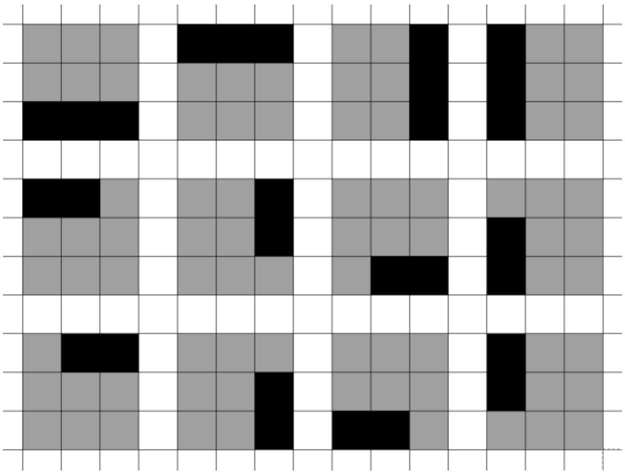

**Fig3.** "Greedy" convolution kernels to fill large empty areas

After these convolution kernels’ run over all maps dead-end branches would shrink and turn to be completely full, wide paths and large spaces are compressed either to the optimal paths or road forks of single width and that results in the map image being formalized into the graph. In this case, the paths are treated as arcs because of the single width, while the forks are regarded as nodes of the graph. In addition, the nodes are estimated by the farness distance from the target point, transition table is constructed, weights are counted and sequences of commands to move along arcs are recorded. In other words, after this step the problem is put into a formalized finite state bot.

An example of the original map and the results of the morphology are shown in Fig. 4 and 5. For illustrative purpose a map of the Bezhitsky district of Bryansk was taken, the image of which was obtained from the Yandex Map Service (people’s map). In the original map, white indicates areas, which are good for moving, while black is the indicator of unfitted areas.

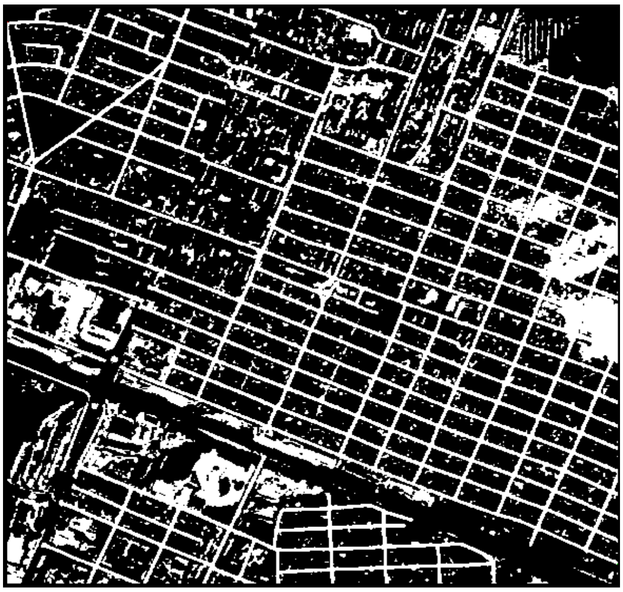

**Fig. 4.** The original map

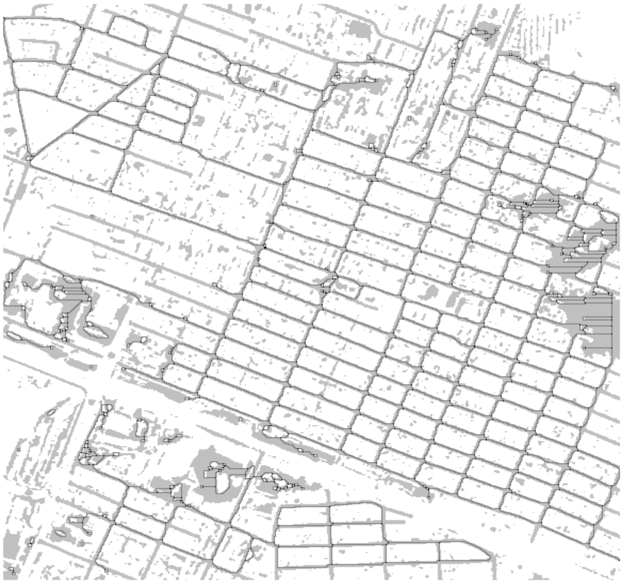

**Fig 5.** Marked paths and road forks

The image of the result has been inverted in brightness for better readability, where white presents not available for traffic areas, light grey indicates available ones, dark grey specifies the ways of single width and black is the indicator of single width road forks.

In its original state, the map contained 61951 locations with possible movements in more than one direction (road forks). The use of morphological operations described above reduced the dimension of the problem to 647 road forks.

## 4. The use of evolutionary algorithms

Even after reducing the dimension of the problem, the use of algorithms based on the search is impractical for reasons of graphical representation and a large number of the transition graph nodes. Morphological algorithms are suitable only for initial optimization and are not good for being the only means in solving the problem of finding paths. The proposed solution is connected with stochastic algorithms. However, the ant algorithm tends to hit local maxima and according to studies [1] it loses in speed to the genetic one. Nevertheless, the classical genetic algorithm does not cope with the task perfectly. It has some very serious problems.

The first and the most serious disadvantage is the” fragility” of the genotype. From the point of view of the current task the way can be estimated either by attaining the goals or in case of covering the distance. The consistent nature of the pathway, shifted to the chromosomes with which standard genetic operators work, leads to the fact that random changes in the genotype violate the integrity and correctness of the paths resulting in the impossibility to evaluate their optimality in the context of the active task.

In terms of implementation it is easier to use approaches that involve a fixed chromosome length. In this case, it is convenient to represent the genotype in the form of chromosomes, in which genes are forks and loci are respectively directions of movement at the forks. This raises the following problem: with a large number of forks very long chromosomes are obtained, that increases the combinatorial complexity and “fragility " of the genotype significantly.

These problems make the use of the classical genetic algorithm almost impossible in the context of the above-mentioned task. However, the work has developed a modification of the algorithm which is permissive for the successful purposes of the problem. The basic principles of modification are as follows.

Using a greedy algorithm to correct an individual's genotype is a way to counter destructive mutations. The use of this mechanism is designed to reduce the search space by creating obviously integral parts of the path, to align the population.

The use of stigmergy mechanism, which is aimed at indirect interaction of individuals in order to share knowledge about the world. Stigmergy is the basic principle of the ant colony optimization algorithm, however, in respect to genetics, it is advantageous to use this approach not in the framework of the population, but through many generations of one and the same individual, i.e. learning cycles.

When using automatic adjustment of the genotype on the basis of the greedy algorithm and stigmergy, the classical mutation and cross are useless, since their effect is offset by the described above algorithms. To avoid this, allele fixation and elitism mechanisms have been introduced. Elitism gives the possibility to disable the actions of all modified genetic operators, in order not to lose good genotypes for retraining under the action of stigmergy. Fixation of alleles allows blocking within a certain number of generations the effect of greedy algorithm on genes having been under mutation and cross operations. The result of the algorithm is shown in Fig. 6. The obtained path is shown in red.

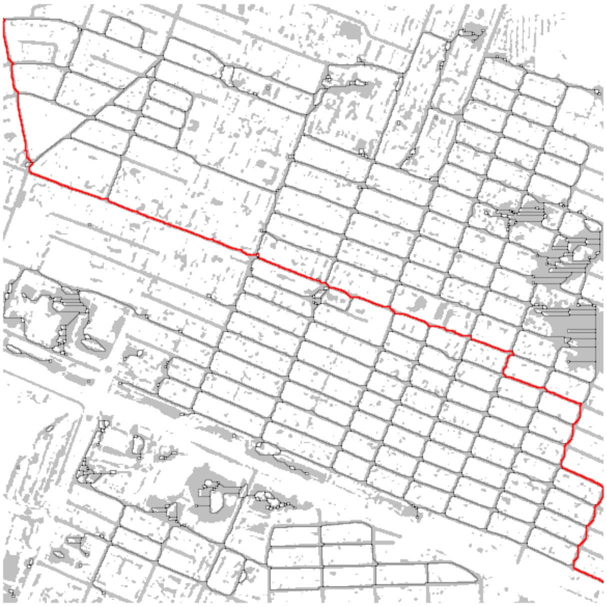

**Fig 6.** The result of the algorithm

## 5. Testing and results

For the combined algorithm testing the images of mazes generated in Daedalus were used, as well as images of maps in the areas with selected roads.

At first there were some tests, which were carried out to compare the full-time operation of the genetic algorithm and then in the mode with disabled operators of crossing and mutation. The main idea of the test is to illustrate that using greedy algorithm with stigmergy solely is less optimal than applying a combination of genetics. The result is an acceleration of convergence by using crossing and mutation operators.

Then a series of comparisons of the developed algorithm with the algorithm D * lite took place, taking into account the fact that D* lite is the most popular in the control devices of autonomous robots. These comparisons were made by using large mazes with the best algorithms based on the variety of being tried topologies, average cases, unfavorable topology (multiple nested horseshoe-shaped obstacles –fig. 7), as well as by using the terrain map small image.

The main interest lies in the card with bad topology. The image of the maze is generated by Daedalus, then it is manually amended: horseshoe-shaped walls have been added and besides it, some passages have to be also made because of the walls, which create unattainable areas in some places making impossible to find the way to the goal.

In the case of an advantageous topology D* lite made a quick work of it, for the average case and on the terrain map it dispatched the developed hybrid algorithm, but if it were disadvantageous topology D* lite would fail because of 2 GB memory. The exact time of the algorithms is given in table 1.

**Table 1**

Time taken by an algorithm

| | Complex topology | Average case | Advantageous topology | Terrain map |
|---|---|---|---|---|
| D* lite | OOM | 225 | 129 | 6 |
| Developed algorithm | 74 | 35 | 153 | 2 |

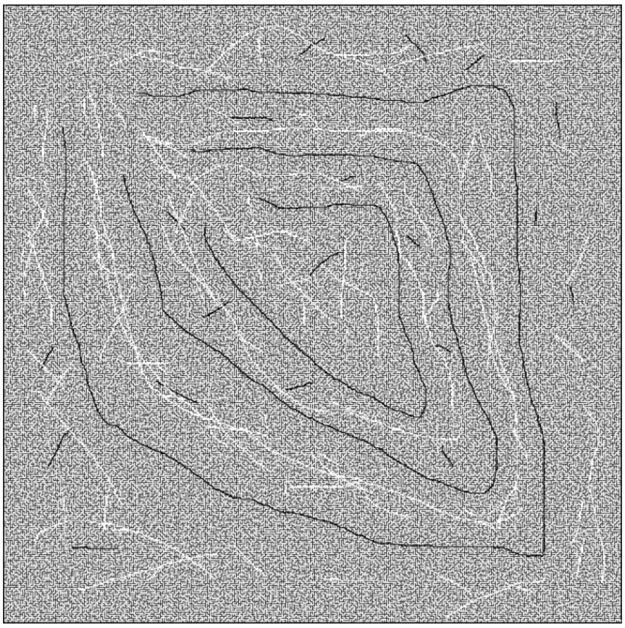

**Fig. 7.** Unfavorable topology

## 6. Conclusion

In the course of investigation we've succeeded in the development of a hybrid algorithm for finding a path in the maze, surpassing the analogues used in practice in complex topologies on large maps.

Further research is required by policies of step-by-step map corrections with rebalance of the graph, the use of concurrent and adaptive crossover and mutation operators.

## 7. References

[1] Alhanjouri M., Alfarra B. Ant Colony versus Genetic Algorithm based on Travelling Salesman Problem // International Journal of Computer Technology and Applications. – Vol. 2, 2011. – pp. 570-578.

[2] Cormen T., Leiserson C., Rivest R., Stein C. Introduction to Algorithms. – 3rd edition. – The MIT Press, 2009. – 1292 p.

[3] Delling, D.; Sanders, P.; Schultes, D.; Wagner, D. Engineering Route Planning Algorithms // Algorithmics of Large and Complex Networks: Design, Analysis, and Simulation. – Springer, 2009. – pp. 117-139.

[4] Dorigo M., Maniezzo V. Colorni A.Ant System: Optimization by a Colony of Cooperating Agents // IEEE Transactions on Systems, Man, and Cybernetics-Part B. – pp. 29-41.

[5] Gonzalez R.C., Woods R.E. Digital Image Processing. – 4rd edition. – Prentice-Hall, 2018. – 976 p.

[6] Hangl S., Ugur E., Piater J. Autonomous robots: potential, advances and future direction // Elektrotechnik und Informationstechnik. – 2017, Volume 134, Issue 6. – pp. 293-298.

[7] Koenig S., Likhachev M. D* lite // Eighteenth national conference on Artificial intelligence. – Edmonton, Alberta, Canada, 2002. – pp. 476-483.

[8] Likhachev M.; Gordon G.; Thrun S. ARA*: formal analysis // School of Computer Science, Carnegie Mellon University, 2003.

[9] Maze classification. Lists of Maze generation methods, Maze solving methods, and classes of Mazes in general.

[10] Russell S., Norvig, P. Artificial Intelligence: A Modern Approach, 2rd Edition. – M.: Wiliams, 2006. – pp. 157-162.

[11] Rutkovskaya D., Pilinsky M., Rutkovsky L. Neural networks, Genetic algorithms and Fuzzy systems., 2nd ed. – M: Hotline-Telecom, 2008. – 452 p.

[12] Savostin I. A., Trubakov A. O. Application of genetic programming for the solution of the problem of the optimal robot's motion path construction / / [Proc. of the XVI international scientific-practical conference of students, postgraduates and young scientists "Youth and modern information technologies."] – Tomsk, 2018. – pp. 118-119.