Acta Fac. Paed. Univ. Tyrnaviensis, Ser. C, 2005, no. 9, pp. 13-17

AN EVOLUTIONARY APPROACH TO OPTIMIZATION OF A LAYOUT CHART

Eva Volna

University of Ostrava, 30th dubna st. 22, Ostrava, Czech Republic
e-mail: Eva.Volna@osu.cz

Abstract: This article presents the use of a genetic algorithm to find an optimal layout for the placement of regular patterns of fixed sizes and simple shapes to minimize the waste. Experiments on various pattern designs indicate that genetic algorithms can effectively be used to obtain highly efficient solutions.
Key words: Genetic algorithms, shape layout optimization.

1. Introduction

The placement of pieces to use the minimal amount of surface is important for industry. Each piece presents a challenge for finding a good placement solution. Much effort has been devoted to automate this process by using artificial intelligence and optimization techniques. This paper presents a methodology to generate a suitable shape layout solution using a genetic algorithm [1]. Genetic algorithms seem to be well suited for shape placement, especially since it does not suffer from local minima problems. Another potential approach could be based on symbolic methods – preferably automated reasoning in fuzzy logic [7]. The variations in various tasks on the use of genetic algorithms stems from the representation and the use of differing crossover and mutation methods. In the next sections the problem of representation, algorithm, and experimental results will be presented.

Figure 1 shows an example of using convex shapes in the experiment, where these shapes are modeled as polygonal objects. Each of the shape has a default initial orientation, but may be rotated. A solution or individual is a structure with the following format: $S = [(P_1, O_1), (P_2, O_2), \ldots, (P_n, O_n), L]$ where S is the solution, P represents each piece, O is the orientation of the piece: 0 for 0 degrees, or 1 for 90 degrees, 2 for 180 degrees or 3 for 270 degrees, and L is the length (cost) of the solution.

Figure 1: Type of pieces.

2. Shape layout solution using a genetic algorithm

Particular individual are strips with the fixed width W (e.g. $W = 2 \times l$, where l is a constant). Every individual $I_k = \{(P_1, O_1), (P_2, O_2), \ldots, (P_{10}, O_{10})\}$, $O_i \in \{0, \pi/2, \pi, 3\pi/2\}$ is represented by its chromosome that is a define set of pieces (see Figure 1) with their own orientations. We experimented with ten pieces (e.g. two rectangle pieces, four square pieces, and four triangle pieces). Each of these ten pieces is used just once in the chromosome. This model of chromosome is similar to that in [2]. Each piece is placed starting at the upper-left edge of the strip. If there is no space to place the piece, we move downwards until there is a space or we run to the right on the strip. When placing a new piece we must check that a space is available. We do this by means of a simple 2-D graphics algorithm for checking that none of the vertices of our polygon is inside another, previously placed polygon [3]. To encode an individual into a string, we use a triangle grid (see Figure 2).

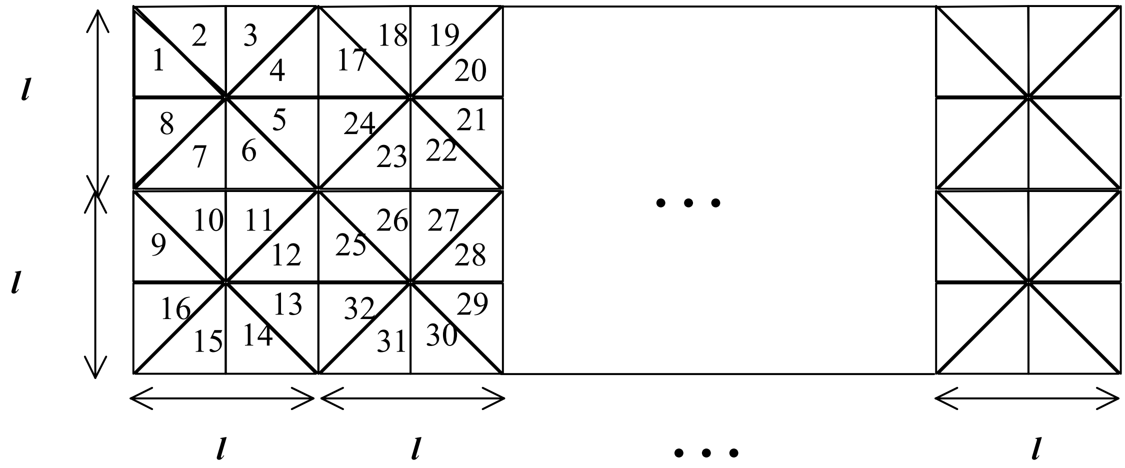

Figure 2: A triangle grid

The purpose of this paper fits in the optimization of apparel shape layout. The genetic algorithm used in this paper is outlined bellow. With its assistance the best optimal population is made from the set of individuals. The initial population is created by randomly generating N individuals. Number of individuals in the population was constant during the whole calculation. The fitness function value of each chromosome is defined as a reciprocal value of its strip's length:

$$F = \frac{l}{L} = \frac{l}{k \cdot l} = \frac{1}{k} \tag{1}$$

Then, for each fitness function the probability of reproduction of its existing individual is calculated by means of standard method (see [4]). All of the calculated fitness function values of the two consecutive generations are sorted descending and individuals attached to the first half creates the new generation. A new individual may be created by either a crossover or a mutation.

The crossover runs in two following steps: we pick a suitable chromosome from our population to crossover at random. After selecting a chromosome to become a part of a new individual, the pieces in that chromosome will be interchanged. We generate a number (a strip's position) that is bounded above by its strip's length. The first (second) new individual includes the first (second) substring of the parent and then we insert all the remaining pieces from this parent in its second (first) substring - that are randomly located, but their orientations are given - to complete our new individual.

If the input condition of mutation is fulfilled (e.g. if a randomly number is generated that is equal to the defined constant), one of the individuals is randomly chosen and its genetic representation is randomly chosen too. The mutation can run in this form: We choose one place of the individual randomly and then we exchange a piece's orientation in this position.

The finding of optimal population is finished when the population achieves the maximal generation or the best solution cannot be further improved in its fitness function value (e.g. in the length of its strip).

3. Experimental results

Once the system has converged, we pick the individual with the best fitness and report its configuration as the solution. We experimented with a population that contained 30 individual and our population size throughout our experiments was constant. Every individual consists of all ten pieces from the defined set of pieces (e.g. two rectangle pieces, four square pieces, and four triangle pieces), see Figure 1. The initial population was created by randomly generating individuals. Each of pieces was placed in the chromosome (e.g. a strips with the fixed width $W = 2\times 1$, where l is a constant) at random and its orientation was generated from the set of the possible orientations. The best and the worst individuals from the initial population are shown in Figure 3.

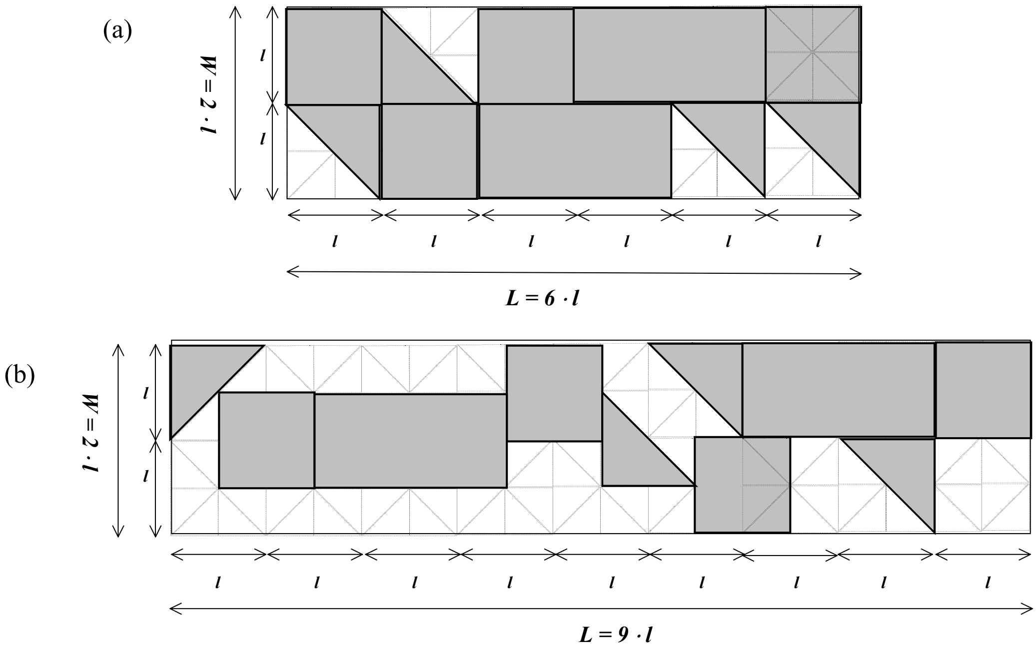

Figure 3: The best (a) and the worst (b) individual from the initial population.

Our experimental results indicate that our genetic algorithm is reasonably good. The calculation is finished in the 569th generation to be characterised by the population of the same individual. The figure 4 illustrates the best individual (e.g. solution) of the final population: its fitness function value is the following $F = \frac{l}{L} = \frac{l}{5 \cdot l} = \frac{1}{5} = 0.2$. If we compare our results to those of human experts, we can observe that it is the only possible solution.

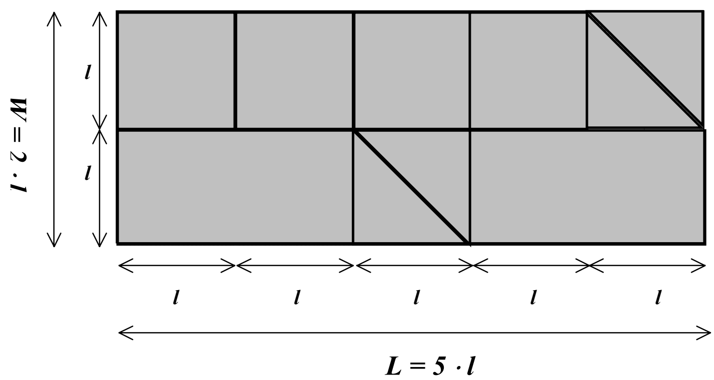

Figure 4: The best individual from the final population.

In Figure 5 the history of fitness function values is shown as: (a) the best individual in the population and (b) the average individual in the population during the whole calculation. Other numerical simulations give similar results. Fitness function is represented here in a relative way so that value one means the upper-most possible fitness function value and value zero means the lowest fitness function value. The three parameters in this method that must be defined for each problem are: the size of the population (e.g. 30 in our experiment), the probability of crossover (e.g. 0.5 in our experiment), and the probability of mutation (e.g. 0.01 in our experiment).

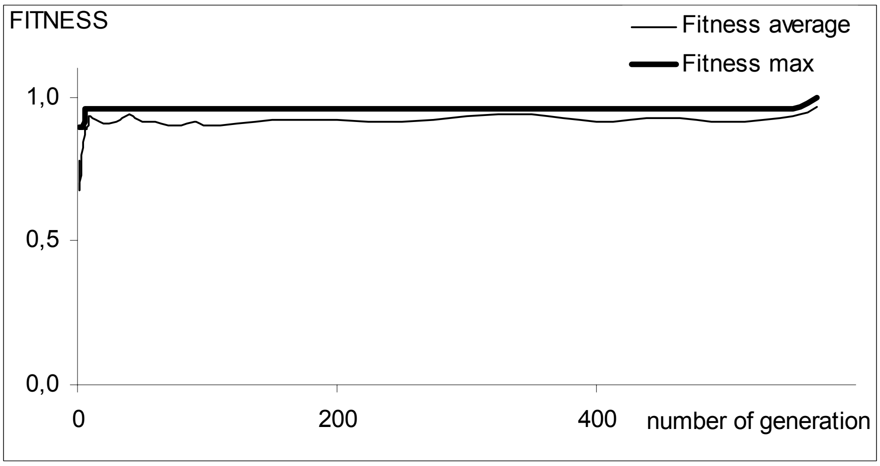

Figure 5: The history of the medial and the best fitness function values during calculation.

A comparison of our results with other researchers [5, 6] is rather difficult, since the overall efficiency depends on the shape of the patterns used. We need to compare our technique with that of other researchers using the same test data. Before we do that a standard test data set has to be established.

4. Conclusion

The results from experiments on various pattern designs indicate that genetic algorithms can effectively be used to obtain highly efficient solutions.

References

[1] GOLDBERG, D. E. '*Genetic Algorithms in Search, Optimization, and Machine Learning'*, Reading MA, Addison-Wesley, pp. 1-23, 1989.

[2] PARGAS, R. P. and JAIN, R. 'A Parallel Stochastic Optimization Algorithm for Solving 2D Bin Packing Problems', *IEEE Proc. of the 9th Int. Conf. on Artificial Intelligence for Applications*, pp. 18-25, 1993.

[3] HARRINGTON, S. '*Computer Graphics*', New York, McGraw-Hill, pp. 3-5, 70-76, 1987.

[4] LAWRENCE, D. '*Handbook of genetic algorithms*', Van Nostrand Reinhold, New York 1991.

[5] BOUNSAYTHIP, C., MAOUCHE, S. and NEUS, M. '*Evolutionary Search Techniques Application to Automated Lay-Planning Optimization Problem'*, IEEE International Conference on Intelligent Systems for the 21st Century, v. 5, pp. 4497-4502, 1995.

[6] MADARASMI, S. and SIRIVAROTHAKUL, P. *’Layout of Garment Patterns for Efficient Fabric Consumption*‘,International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC 2002), Phuket, Thailand, pp. 1176-1179.
[7] HABIBALLA, H. Non-clausal Resolution in Fuzzy Predicate Logic with Evaluated Syntax (background and implementation). In Proceedings of International conf. The Logic of Soft Computing IV, Ostrava, 2005.