AN EVOLUTIONARY APPROACH TO GEOMETRICAL PLACE PROBLEMS

Crina Groşan[1], Mihai Oltean[1], Mihaela Oltean[2]

Abstract. Solving geometrical place problems by applying mathematical methods is a difficult task. In this paper, we propose an evolutionary approach for geometrical place problems. Evolutionary Algorithms deal with a population of solutions. This population (initial randomly generated) is evolved using genetic operators. Population obtained after a specified number of iterations will contain solutions which (very probable) meet the geometrical place problem conditions. Moreover the proposed evolutionary algorithm is very fast and it does not need very much information about the geometrical place problem being solved. The only information that has to be supplied is the distance between a potential solution (an individual) and the geometrical place.

Keywords: Geometrical place problems, Evolutionary algorithms, Genetic Algorithms

1. Introduction

Evolutionary algorithms have been successfully applied in various domains: mathematics, engineering, chemistry, physics, medicine, etc. Their great advantage consists of their ability to obtain multiple solutions in a single run, due to their capacity of dealing with a population of solutions.

In this paper we propose an evolutionary approach for solving geometrical place problems. Geometrical place problems arise in many practical situations (usually in the engineering situations). In the next section an overview of evolutionary algorithms is presented. In the third section an evolutionary algorithm for geometrical place problems is proposed. This algorithm will be effectively applied for solving several geometrical place problems introduced in section 4. A set of conclusions is marked out in section 5 of the paper.

2. An overview of Evolutionary Algorithms

Evolutionary Algorithms (EAs) have been introduced in 1965 by John Holland ([6]). Many surveys on Evolutionary Algorithms and their applications can be founded in ([2], [3], [5]).

EAs use a population of feasible solutions. This population is initially randomly generated over the search space (the definition domain). By applying genetic operators (such as selection, mutation, crossover, etc.) it is hopped that the solutions (also called chromosomes or individuals) will be improved. Each individual from population is evaluated by using a quality (fitness) function. For applying genetic operations individuals are selected according to their fitness. Many selection mechanisms may be used (see [1], [2], [5]). The selected individuals are modified by applying crossover and/or mutation operator. Various forms of these operators can be founded ([4], [7], [8], [9]). In this way new solutions are obtained. It is hopped that some of these new solutions are better than the existing solutions. There are many ways for accepting the new solutions (also called offspring) in the new population. Some algorithms accept the new solution only if this solution is better than its parent (or parents).

Different types of solution encoding can be used depending on the problem. Very usual are binary and the real encoding. Different forms of genetic operators can be considered, depending on these encoding on the individual encoding. A standard EA is described in what follows:

The EA starts with an initial population of individuals each of them being randomly generated over the search space. The population size is fixed. The following steps are repeated for a fixed number of generations. All individuals are evaluated using a fitness function. The selection operator is applied in order to obtain a mating pool. Selected chromosomes are subject to crossover operator and mutation. The evolutionary algorithm is described bellow.

A Standard Evolutionary Algorithm

```
begin
Step 1. Set $t = 0$;
Step 2. Randomly initialize the population $P(t)$;
Step 3. Repeat
           Step 3.1. Evaluate individuals from $P(t)$;
           Step 3.2. Selection on $P(t)$. Let be $P'(t)$ the set of selected individuals.
           Step 3.3. $P''(t)$ = Crossover on $P'(t)$;
           Step 3.4. $P'''(t)$ = Mutation on $P''(t)$;
           Step 3.5. $t = t + 1$;
           Step 3.6. $P(t)$ = Survival $(P'''(t - 1) \cup P(t - 1))$;
     Until $t$ = Number of generations.
end
```

Remark: Different evolutionary algorithms use different survival mechanisms for can be implemented. The survival decides which individuals (from parents and offspring) enter the new population

3. An Evolutionary Algorithm for solving geometrical place problems

In this section, the proposed EA is described. The algorithm is called *Geometrical Place Evolutionary Algorithms* (GPEA) and it will be applied for solving several geometrical place problems.

A real encoding of solutions is considered. That means that a solution of the problem is encoded as a real number (or an array of real numbers) in a given definition domain. Initial population is randomly generated with uniform distribution. The following steps are repeated for a fixed number of generations. Each individual from population is evaluated using an adequate fitness function. A number of individuals equal to the initial population size are selected in order to fill the mating pool. Individuals from the mating pool are randomly paired and recombined using convex crossover operator ([5]). Two parents always produce two offspring. From these four individuals (two parents and two offspring) the best two enters the new population. Over this new population obtained mutation operator (Gausian perturbation with a fixed standard deviation (sigma)) is applied. Each individual is affected by mutation. By mutation an offspring is obtained from each individual. The best between offspring and the parent is accepted in the new population. Individuals obtained after these steps will constitute the population of the next generation. GPEA is presented bellow.

GPEA

```
begin
Step 1. Set t = 0;
Step 2. Randomly initialize the population P(t);
Step 3. Repeat
           Step 3.1. Evaluate population P(t);
           Step 3.2. Selection on P(t). Let be P'(t)
                  the set of selected
                  individuals.
           Step 3.3. P''(t) = Crossover on P'(t);
           Step 3.4. P'''(t) = Mutation on P''(t);
           Step 3.5. t = t + 1;
                    P(t) = P'''(t – 1);
     Until t = Number of generations.
end
```

4. Numerical Experiments

Several numerical experiments are performed in this section in order to assess the performance of applying evolutionary algorithms for solving several geometrical place problems. In all these experiments the GPEA algorithm - described in the previous section - is used.

4.1. Experiment 1

In this experiment geometrical place of the points for which the distance to a given point is equal to a given constant *k* is searched.

In two-dimensional spaces the geometrical place consists of circle of center the given point and radius k. In third dimensional spaces the geometrical place consists on the sphere of center the given point and radius k. The problem can be extended to high dimensional space.

The parameters used by the evolutionary algorithm are presented in Table 1.

Parameter	**Value**
Population size	500
Sigma	1
Number of generations	900

Table 1. The parameters used by GPEA for Experiment 1.

The initial population is depicted in Figure 1.

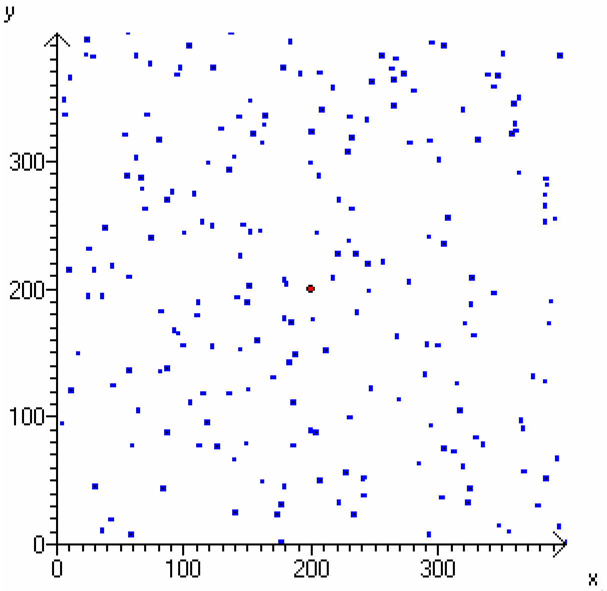

Figure 1. The initial population randomly generated for the Experiment 1.

The result obtained after 200 iterations is depicted in Figure 2.

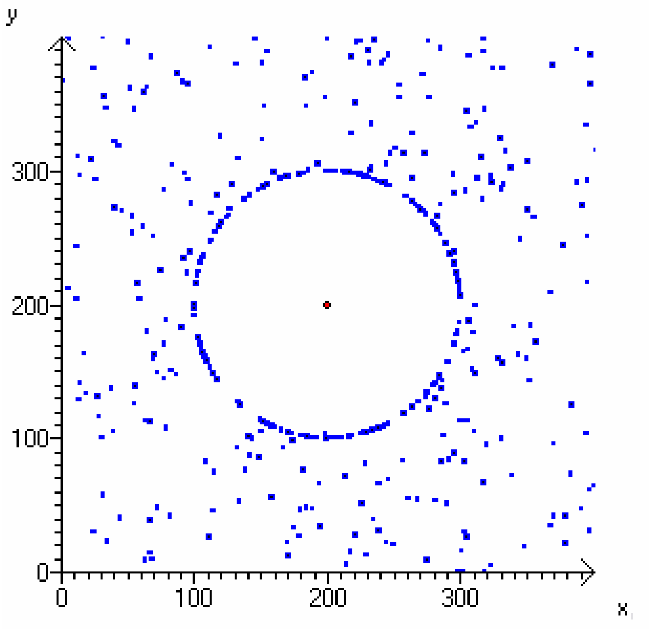

Figure 2. The population after 200 iterations.

The final result obtained after 900 iterations is depicted in Figure 3.

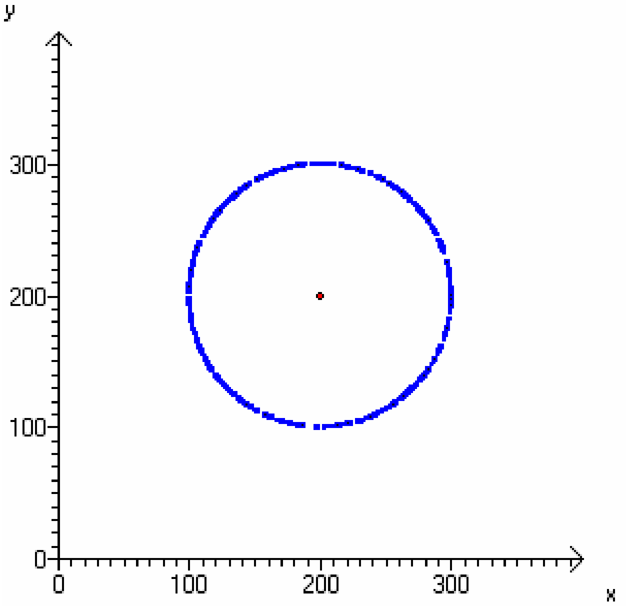

Figure 3. The final population for Experiment 1 obtained after 900 iterations.

4.2. Experiment 2

In this experiment geometrical place of the points for which the sum of the distances to two given points is equal to a given constant *k* is searched.

In two-dimensional spaces the geometrical place consists of ellipse of focuses the two given point. In third dimensional spaces the geometrical place consists on the ellipsoid of focuses the two given points. This problem can also be extended to high dimensional space.

The parameters used by the evolutionary algorithm for this experiment are presented in Table 2.

Parameter	**Value**
Population size	500
Sigma	1
Number of generations	900

Table 2. The parameters used by GPEA for Experiment 2.

The initial population is depicted in Figure 4.

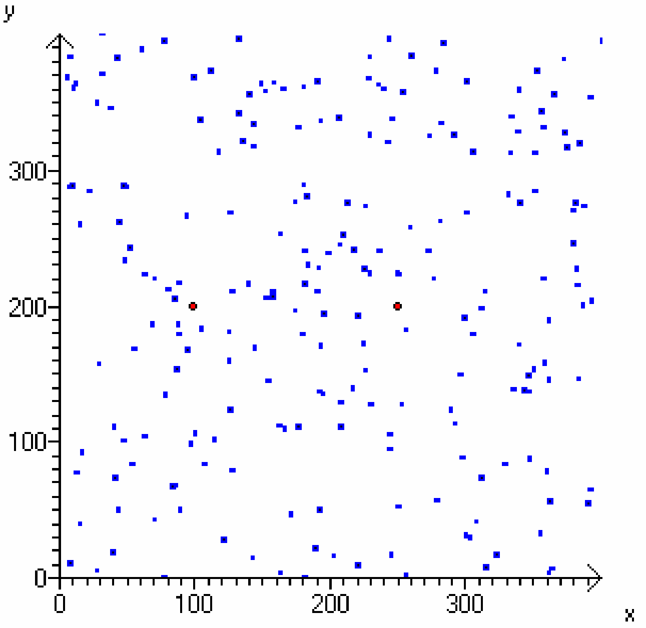

Figure 4. The initial population for the Experiment 2.

The population after 400 generations is depicted in Figure 5.

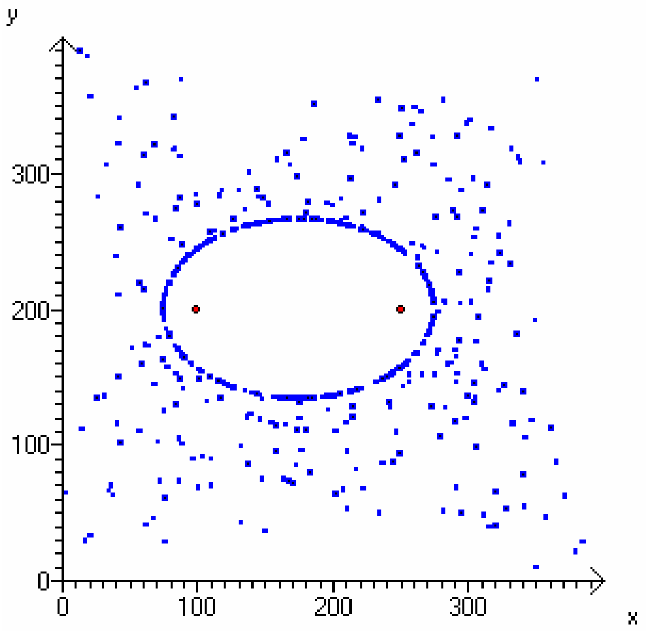

Figure 5. Population after 400 generations for the Experiment 2.

The final population obtained after 900 generations is depicted in Figure 6.

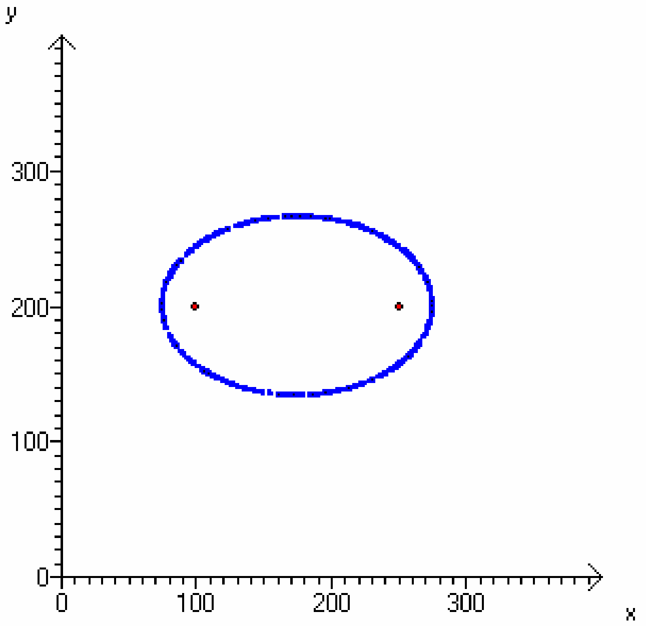

Figure 6. Final population after 9000 generation for Experiment 2.

4.3. Experiment 3

In this experiment geometrical place of the points for which the difference (in absolute value) of the distances to two given points is equal to a given constant *k* is searched.

In two-dimensional spaces the geometrical place consists of hyperbole of focuses the two given point. In third dimensional spaces the geometrical place consists on the hiperboloid of focuses the two given points. This problem can also be extended to high dimensional space.

The parameters used by the evolutionary algorithm for this experiment are presented in Table 3.

Parameter	**Value**
Population size	500
Sigma	1
Number of generations	600

Table 3. Parameters used by GPEA for Experiment 3.

The initial population is depicted in Figure 7.

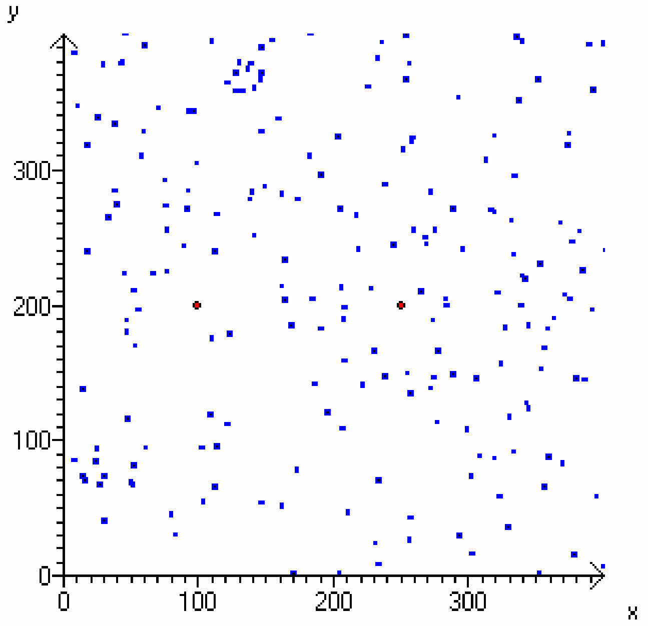

Figure 7. The initial population used in Experiment 3.

The population after 100 generations is depicted in Figure 8.

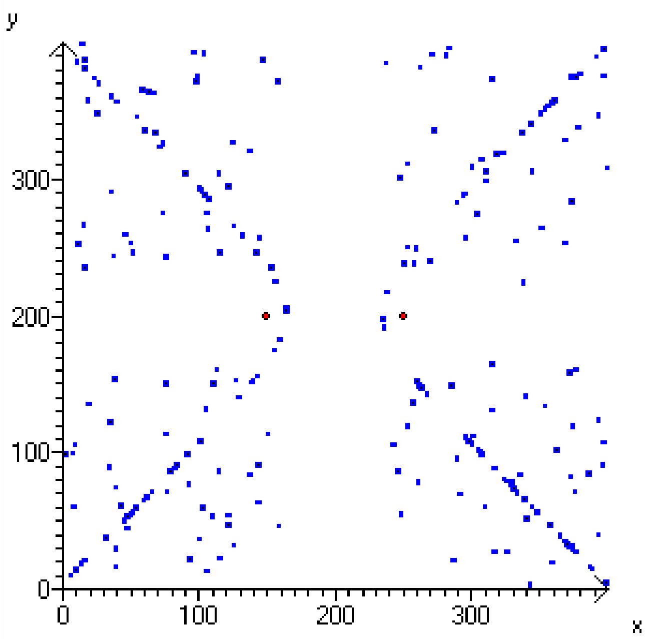

Figure 8. The population after 100 generations for Experiment 3.

The final population obtained after 600 generations is depicted in Figure 9.

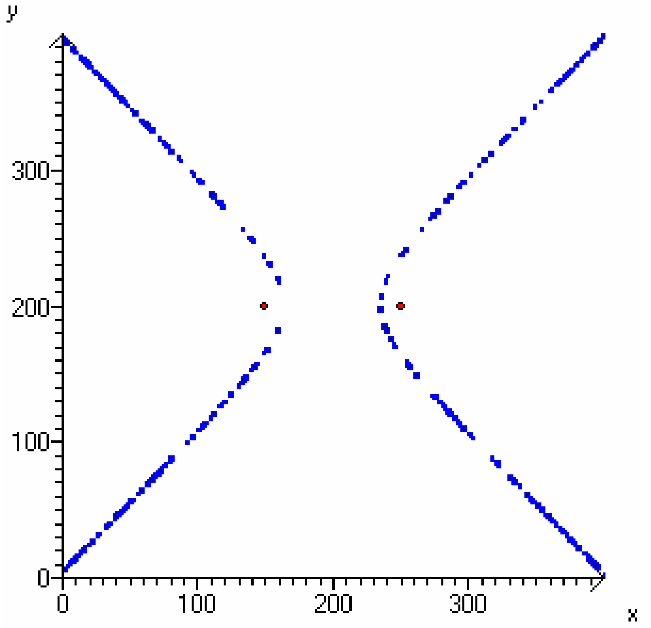

Figure 9. The final population for Experiment 3 obtained after 600 generations.

4.4. Experiment 4

In this experiment the geometrical place of the point for which the distance to two given straight lines is equal is searched. This geometrical place consists of two straight lines: the bisecting line of the given straight lines and the perpendicular line on the bisecting line.

The parameters used by the evolutionary algorithm for this experiment are presented in Table 4.

Parameter	Value
Population size	500
Sigma	1
Number of generations	1000

Table 4. The parameters used by GPEA for Experiment 4.

The initial population (randomly generated) is depicted in Figure 10.

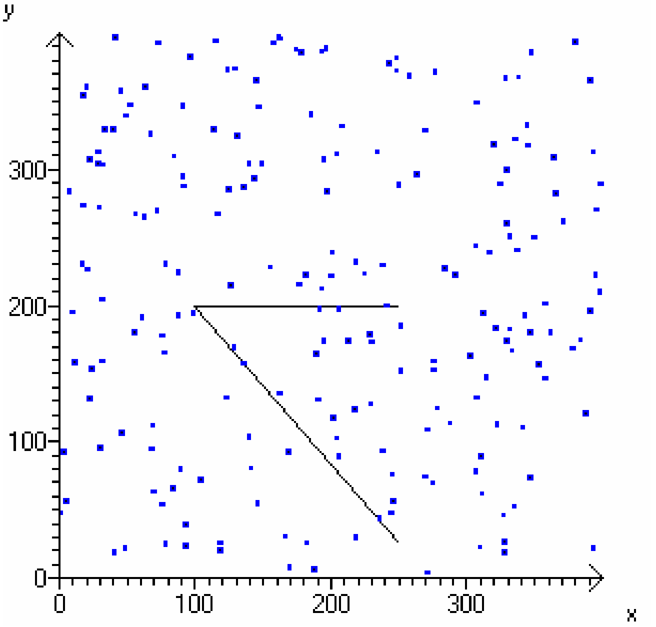

Figure 10. Initial population used in Experiment 4.

The population after 100 generations is depicted in Figure 11.

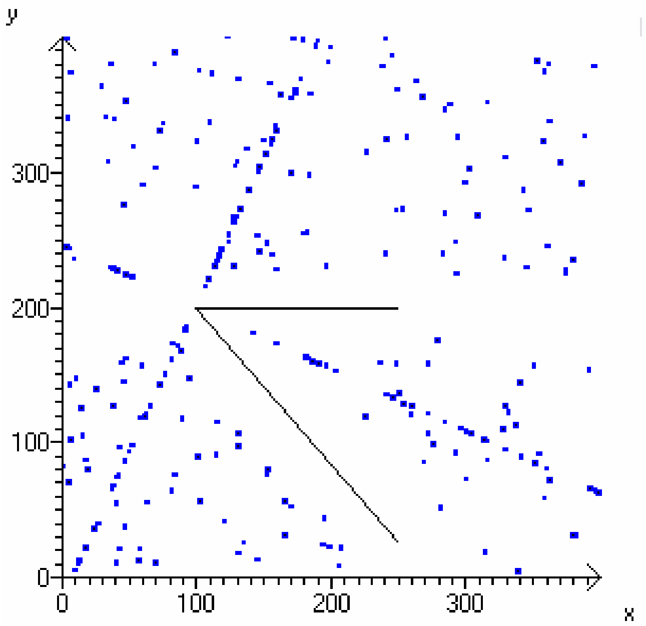

Figure 11. The population after 100 generations for Experiment 4.

The final population obtained after 1000 generations is depicted in Figure 12.

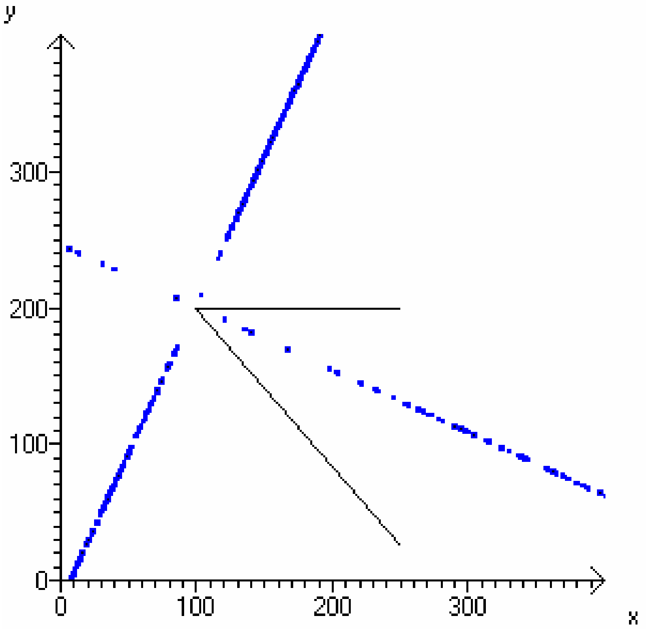

Figure 12. Final population for Experiment 4 obtained after 1000 generations.

5. Conclusions

The experiments presented here are simple examples to illustrate how evolutionary algorithms work for this kind of problems. Certainly, any others geometrical place problems can be considered. The running time for every of the considered experiments is less than one minute. The advantage of applying evolutionary algorithms consists, first of all, in their capacity to determinate in a single run all these solutions. Geometrical place problem are another domain where the applicability of the evolutionary techniques is very useful.

References

1. J. E. Baker. Adaptive selection methods for genetic algorithms. In Proceedings of First International Conference on Genetic Algorithms, 101-111, 1985.
2. G. E. P. Box. Evolutionary operators: a method for increasing industrial productivity. Journal Royal Statistical Society, 6: 81-101, 1993
3. L. Davis. Handbook of Genetic Algorithms. Van Nostrand Reinhold, New York, 1991.
4. L. J. Eshelman, R.A. Caruna, J. D. Schaffer. Biases in the crossover landscape. In Proceeding of the Third International Conference on Genetic Algorithms, J. Schaffer (ed.), Morgan Kaufmann Publisher, Los Altos, CA, 10-19, 1989.
5. D.E. Goldberg. Genetic algorithms in search, optimization and machine learning. Addison Wesley, Reading, MA, 1989.
6. J. Holland. Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, 1975.
7. J. D. Schaffer, A. Morishima. An adaptive crossover distribution mechanism for genetic algorithms. In Proceeding of the Second International Conference on Genetic Algorithms, J. J. Grefenstette (Ed.),Lawrance Erlbaum Associates, Hillsdale, NY, 36-40, 1987.
8. W. M. Spears, K. A. De Jong. On the virtues of uniform crossover. In Proceedings of the Fourth International Conference on Genetic Algorithms, Morgan Kaufmann Publisher, 230-236, 1991.
9. G. Syswerda. Uniform crossover in genetic algorithms. In Proceedings of the third Conference in Genetic Algorithms, J. Schaffer (ed.), Morgan Kaufmann Publisher, Los Altos, CA, 2-9, 1989.

[1] Department of Computer Science,Faculty of Mathematics and Computer Science, Babeş-Bolyai University of Cluj-Napoca, e-mail: {cgrosan, moltean}@cs.ubbcluj.ro

[2] Faculty of Sciences, Lucian Blaga University, Sibiu, e-mail: mihaelaoltean@personal.ro, olteanmihaelaelena@yahoo.com