Biuletyn WAT
Vol. LVIII, Nr 2, 2009

# An application of the genetic algorithm to optimize location of buoys

TOMASZ PRACZYK

Naval University, 81-103 Gdynia, Śmidowicza 69, Poland

**Abstract.** The paper addresses the problem of building an automatic, spare, radar system to coastal navigation. To fix position, the system uses the information about buoys surrounding the ship. Accuracy of the system depends on many factors. One of them is the way of locating buoys on the given area of the sea. To make the task of the system easier and to make the position fixed by the system more accurate, the buoys should be appropriately arranged. The paper suggests the solution based on genetic algorithms to arrange the buoys. The solution proposed was tested experimentally and results of the tests are presented at the end of the paper.
**Keywords:** optimization, genetic algorithms, radar navigation
**Universal Decimal Classification:** 621.396.96

## 1. Introduction

There are a few concepts of automating radar navigation [6, 7, 8, 9]. One of them is based on a system of self buoys, i.e., buoys recognizable by the positioning system (The problem of differing self buoys from other buoys is presented among other things in [10]). To fix position the system determines a vector of distances (and optionally bearings) to all self buoys surrounding the ship, using radar for that purpose. The vector of distances fixed by the system is passed on to appropriately prepared artificial neural network. The task of the network is to fix position of the ship.

Generally, the task of the network can be viewed as approximation of a position function defined as follows:

$$F : V_D \to P, \tag{1}$$

where: $V_D \subset \Re^n$ — the set of vectors of the size $n$ including distances (and optionally bearings) to buoys;
$P$ — the set of points in geographical coordinate system, i.e., the set of points in the following form $(\phi, \lambda)$, where $\phi$ is the latitude and $\lambda$ is the longitude.

Accuracy of approximating the function above strongly depends on its shape. The more the function is undulated, the greater difficulties with its approximation. To make the task of the network easier and to make the position fixed by the network more accurate, the shape of the position function should be possibly the easiest. Since the shape of the position function depends on mutual arrangement of buoys, it can be easily modified through moving buoys into different places. However, the problem is how to arrange buoys to obtain flat, easy shape of the function. To solve this problem, different methods can be used. Some of them, suggested in the paper, are genetic algorithms (GAs). To test GAs in the problem of arranging the buoys, simple experiments were performed. In the experiments, classical Canonical GA (CGA) [3, 4, 13] and Eugenic Algorithm (EuA) [1, 5, 11] were used. The task of the algorithms was to arrange ten buoys on a virtual area of the sea of the size $10 \times 10$. To evaluate arrangements, they were used to create positioning neural networks which, then, were tested in terms of accuracy of position generated in testing points of the considered research stretch. In the experiments, arrangements generated by GAs were compared to random arrangements.

The paper is organized as follows: section 2 is a short presentation of the positioning system; section 3 is a description of the problem of arrangement of the buoys; section 4 is a description of evolutionary techniques used to solve the problem mentioned, section 5 is a report from the experiments; and section 5 is a summary.

## 2. The concept of the positioning system

There are three key elements of the system: radar, the system of self buoys, and positioning neural network. Radar is a common device on every ship used to measure distance and bearing to different objects. The information about objects and generally the information about world surrounding a ship is presented in a radar screen. Additionally, the information about objects is transmitted outside radar. Most currently used radars generate the message formatted according to NMEA 0183 (the standard established by National Marine Electronics Association) standard. The message includes all the information required by the system, i.e., the information about distances (and optionally bearings) to all objects visible by means of radar. With regard to buoys, they are present on almost every stretch to facilitate navigation. In our case, the buoys have to have three basic features. First, they cannot change position over time (the so-called super-buoys). Second, they have to be visible by means of radar (each buoy has to be equipped with the so-called racon, i.e., a device to reflect radar waves). Third, all the buoys have to be recognizable by the system, i.e., they have to be differentiable from other buoys occurring on the given area of the sea. To differ the self buoys from other buoys, ideas from artificial immune systems can be used [10]. The third element of the system is approximating neural network. At present, to approximate position, two General Regression Neural Networks (GRNNs) are used. The first one is used to approximate the latitude while the second one to approximate the longitude. Both GRNNs implement the following function:

$$f\left(x,\sigma\right)=\frac{\sum_{k=0}^{Z} W_k \phi_k\left(x,\sigma\right)}{\sum_{k=0}^{Z} \phi_k\left(x,\sigma\right)}, \tag{2}$$

$$\phi_k\left(x,\sigma\right)=e^{-\frac{\left(x-x_k\right)^2}{2\sigma^2}}, \tag{3}$$

where: $f\left(x,\sigma\right)$ — the value of approximated function in the point $x$ (in our case $f$ is one of unknown coordinates of the ship, i.e., $\phi$ or $\lambda$, whereas $x$ is the vector of distances to buoys fixed in the position ($\phi$, $\lambda$));
$\sigma$ — the parameter;
$W_k$ — the true value of the function $f$ in the training point $x_k$ (in our case $W_k$ is the value of either $\phi$ or $\lambda$);
$x_k$ — $k^{\text{th}}$ training point (in our case the sample vector of distances to buoys for which we know accurate position of the ship);
$Z$ — the number of training points.

To prepare the networks to work we should have at our disposal the set of training points in the form ($\mathbf{v}_k$, $\phi_k$, $\lambda_k$), $k$ = 1…$Z$, where $\mathbf{v}_k$ is the vector recorded in the point ($\phi_k$, $\lambda_k$) including distances to buoys. All the training points can be produced in a laboratory, on a personal computer. There is no necessity to perform any measurements at sea what is the great advantage of the presented system.

The system works as follows. At the beginning, self buoys are selected from among all buoys visible on a stretch. Next, all self buoys are ordered according to North-South direction. In the following step, the vector of the size $n$ (at each point in time at most $n$ buoys can be visible by means of radar; the field of vision of radar is restricted to the area determined by the range of radar observation $R$) including distances to self buoys is created. At the beginning of the vector distances to North-most buoys are stored. Distances to South-most buoys are at the end of the vector. When fewer than *n* self buoys are visible on a stretch, the end of the vector is filled in with zeros. The last activity of the system is activation of approximating neural networks. The networks fix position of a ship based on the vector of distances prepared beforehand. Illustration of the system is presented in Fig. 1.

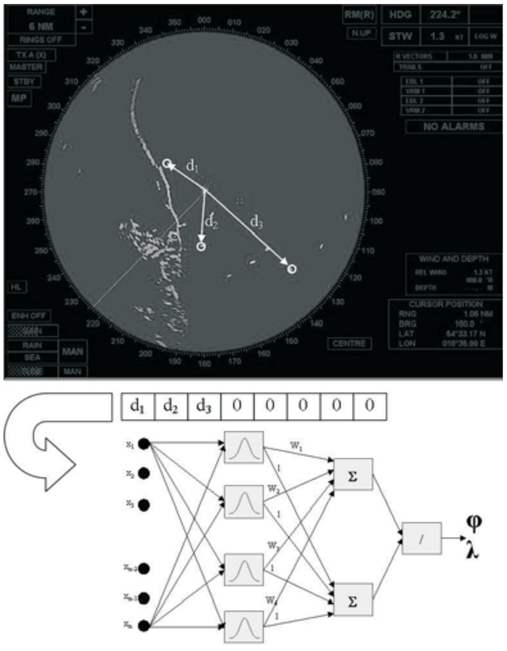

Fig. 1. The concept of the system

The greatest advantages of the system described above are simplicity in creating and autonomy, i.e., independence on outside sources of information. The latter feature is particularly important in the case of military ships.

## 3. Optimization of locating the buoys

The most important evaluating criterion of a positioning system is its accuracy. Accuracy of approximating the position function by means of GRNN depends on the value of $\sigma$ and on training points used to create the network. The first step to adjust an initially created GRNN to a task is to tune the value of $\sigma$. The only method to improve performance of the network when the value of $\sigma$ is fixed and results of the network are still unsatisfactory is to add next training points. The newly introduced points should come from areas where the greatest errors of the position occur. The additional points on the one hand enhance performance of the network but on the other hand extend the calculating time necessary to fix a position (in our case we deal with software, sequential implementation of GRNNs). Accordingly, we cannot simply add next training points to make the system more accurate. Too many training points used to create positioning GRNNs can make them too slow and thereby unpractical. The solution to this problem is to facilitate the position function. The simpler the shape of a function, the fewer training points necessary to accurately approximate it. In the case of the position function to make it flat and easy to approximate similar vectors of distances should correspond to similar positions. Moreover, different vectors should represent different positions. The only method to achieve such effect is to appropriately arrange the buoys (each change in locating the buoys induces a change of the shape of the position function). To accomplish the flat shape of the position function, the buoys should be arranged so as to minimize the following evaluation function:

$$F_e(a) = \frac{2}{m(m-1)} \sum_{i<j}^{m} \left(d_{ij} - d_{ij}^{*}(a)\right)^2, \tag{4}$$

where: $a = \langle \phi_1, \lambda_1, \phi_2, \lambda_2, \ldots, \phi_n, \lambda_n \rangle$ — the vector including positions of $n$ self buoys (the arrangement of buoys);
$m$ — the number of points uniformly distributed on the stretch and used to evaluate arrangements of buoys;
$d_{ij}$ — the normalized distance between the $i^{\text{th}}$ and the $j^{\text{th}}$ point of the stretch;
$d_{ij}^{*}$ — the normalized Euclidean distance between the $i^{\text{th}}$ vector (including distances to buoys) representing the $i^{th}$ point of the stretch and the $j^{\text{th}}$ vector representing the $j^{\text{th}}$ point of the stretch.

In the experiments reported further, to create arrangements of buoys we used two types of GAs, i.e., CGA and EuA. To evaluate created arrangements both algorithms used the following fitness function:

$$F_f(a) = \frac{1}{F_e(a)}. \tag{5}$$

In the following sections, both types of GAs mentioned above are described. The description of GAs is preceded by a brief introduction to GAs itself.

## 4. Genetic algorithms

Genetic algorithms are a class of search, stochastic, population based algorithms drawing inspiration from ideas and principles of natural evolution. All of them operate according to the following simplified procedure.

```
genetic_algorithm()
{
t:=0;
generate initial population at random;
evaluate population;
do{
   select parental individuals from population for reproduction;
   apply operators to selected individuals and generate offspring;
   evaluate offspring;
   replace some parents with some offspring;
   t:=t+1;
   }while(termination criterion is satisfied)
}
```

Fig. 2. The pseudocode of GA

At first, random population of individuals, represented as binary strings, is generated. The whole population is then evaluated, i.e., a measure of goodness, called fitness, is assigned to each individual. In the next step, some individuals from the population are selected. Selection of individuals for reproduction is performed by means of one out of the three main selection schemes: proportional, ranking and tournament selection scheme [2, 3]. All the schemes mentioned present the philosophy of selection according to which better individuals have greater chance to be selected than the worse ones. The chosen individuals become parents of newborn offspring and they undergo different genetic operators. In Gas, the most frequently used genetic operators are crossover and mutation [2, 3]. The crossover operator recombines genetic material from two parental chromosomes creating offspring chromosome. The mutation introduces random changes into chromosome flipping some of its bits (genes) to their opposite state. In GAs, the key operator is crossover [3]. The mutation usually plays supporting role.

Each new individual generated by means of the genetic operators described above is then evaluated, i.e., each of them receives fitness. The next activity of GA is to replace parents with offspring. This way we obtain a new population of individuals which depending on the replacement strategy contains either newly generated individuals or both parents and children. The new population undergoes the same evolutionary procedure as the previous population, i.e., individuals are selected for reproduction, different operators are applied regarding individuals selected, offspring is generated, evaluated and finally the next population arises. This procedure is repeated until some stopping criterion is satisfied.

## 4.1. Canonical Genetic Algorithm

In principle, functioning of CGA was already explained in the previous section. To fully define CGA, it is only enough to present in detail such elements as: the way of selecting individuals for reproduction and functioning of the basic GA operators, i.e., crossover and mutation.

As noted above, there are three different strategies with regard to selection of parental individuals: proportional, ranking and tournament selection. Since, in the experiments, reported further, only the tournament selection was used, we decided to describe only this type of selection. In order to select individuals for reproduction, the tournament selection organizes a cycle of tournaments. Each tournament takes place with participation of a slight number of randomly selected individuals from the entire population. Selecting individuals for the tournament is uniform. Each tournament ends with selecting the winner individual who is better than the remaining individuals taking part in the tournament. The winner individual is, then, as the only individual taking part in the tournament, introduced into newly created population. In order to fill newly created population out it is necessary to repeat the tournament many times.

The next issue that requires to be explained in detail is functioning of the basic GA operators, i.e., crossover and mutation. The task of the former is to exchange genetic material between individuals. The latter introduces random perturbations into chromosomes.

There are three basic forms of crossover, i.e., one-point, two-point and uniform crossover [12, 13]. One-point crossover, used in the experiments, cuts two parental chromosomes into two segments. The left segment from the first chromosome is, then, attached to the right segment of the second chromosome producing the first offspring. The same procedure, as above, is applied in relation to the remaining unattached segments from both chromosomes. In this way, the second offspring arises. Two-point crossover works similarly. However, this time parental chromosomes are cut in two places. To produce offspring two-point crossover swaps central segments of both chromosomes. The third version of crossover is uniform crossover. It exchanges corresponding bits from both chromosomes with the probability 0.5. All types of crossover are depicted in Fig. 3.

The next evolutionary operator used in CGA is mutation. As noted above, mutation is rather supporting operator in relation to crossover. Nevertheless, functioning of CGA without this operator is rather impossible. Mutation helps in the exploration of unknown regions of genotype space. Lack of this operator may cause CGA to be not able to produce new, perhaps useful combinations of bits. Mutation is simple operator since it randomly flips bits from a chromosome to their opposite state. The course of action of mutation is presented in Fig. 4.

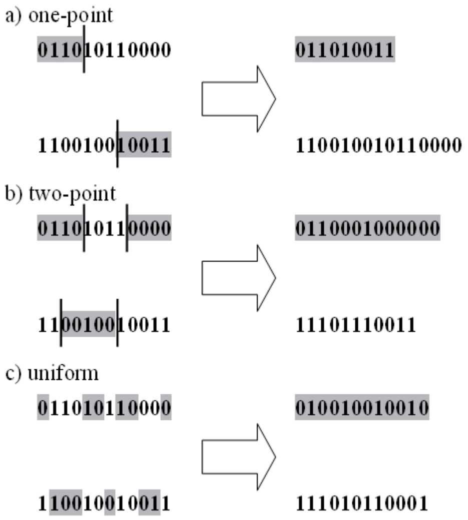

Fig. 3. Three types of crossover: a) one-point crossover; b) two-point crossover; c) uniform crossover

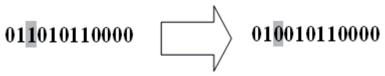

Fig. 4. Mutation

In the experiments reported further, we used CGA with tournament selection, one-point crossover (the form of crossover used in the experiments always produced offspring of the same length as parents), and mutation.

## 4.2. Eugenic Algorithm

The next GA used in the experiments is Eugenic Algorithm (EuA). The main difference between CGA and EuA is a way of constructing offspring. In order to generate offspring CGA always uses two parents, i.e., it applies sexual reproduction. In EuA, a single offspring is created based on the information included in the whole population of individuals. Each individual from the population can affect the form of the newborn offspring. However, better individuals have greater chance to introduce their genetic material into offspring than worse individuals. The pseudocode of EuA is presented in Fig. 5 [11].

In short, EuA proceeds as follows. At first, randomly initialized population of binary individuals is generated. Then, a new individual is created. To create the new individual, EuA uses a temporary population of individuals, which at the beginning is a faithful copy of the original population. Initially, all genes of the new individual are unset. In the subsequent steps, the genes are gradually fixed. First, the most significant gene is fixed, i.e., the gene which seems to have the greatest influence on fitness of individuals. To estimate the significance of a gene, EuA uses absolute difference between mean fitnesses fixed for various alleles (allele is the value of gene; in GAs allele is from the set {0,1}), i.e., [11]:

$$S_g = \left| \bar{f}(g,1) - \bar{f}(g,0) \right|, \tag{6}$$

where $S_g$ is the significance of the gene $g$ and $\bar{f}(g,a)$ denotes the mean fitness of individuals whose the gene $g$ has allele $a$ assigned. The unset gene whose the value of Eq. (6) is the greatest is considered to be the most significant. The most significant gene is the first gene which obtains a value.

$$g^* = \arg\max_{g \in u} S_g. \tag{7}$$

In Eq. (7) $g^*$ denotes the most significant gene while $u$ is the set of unset genes in the newly created individual. Selecting the value of the gene is performed according to the following formula [11]:

$$x_{\text{new}}[g] = \begin{cases} 1 & \text{if} \quad P(g,1) > P(g,0) \\ 0 & \text{otherwise}, \end{cases} \tag{8}$$

$$P(g,a) = \frac{\bar{f}(g,a)}{\sum_{b \in \{'0','1'\}} \bar{f}(g,b)}, \tag{9}$$

where $x_{\text{new}}[g]$ denotes the value of the $g^{\text{th}}$ gene of the newborn individual $x_{\text{new}}$ and $P(g, a)$ is the probability of the allele $a$ to be chosen for the gene $g$. In the next step, the gene chosen is removed from the set of unset genes and the temporary population is restricted. In order to take a decision about the restriction of the population EuA estimates epistasis $E$, i.e., the level of interdependencies between individual genes. The estimate is used by EuA to calculate the probability of the restriction $P_R$ [11].

$$E = 1 - D_{\max}, \tag{10}$$

$$D_{\max} = \max_{g \in u} \left| P(g,1) - P(g,0) \right|, \tag{11}$$

$$P_R = E. \tag{12}$$

```
eugenic_algorithm()
{
t:=0;
//generate initial population of binary strings at random
pop:=getRandomPop();
while(termination criterion is not satisfied)
    {
     u:={1,2,...,l};
     rpop:=pop;
     while(!u->isEmpty())
      {
      //select the most significant gene g* out of all genes from u
      g*:=rpop->getTheMostSignificantGene(u);
      //set the gene g* of newborn individual x_new
      x_new[g*]:= rpop->setAllele(g*);
      u->removeGene(g*);
      //restrict rpop when epistasis of unset genes of x_new is high
      //E denotes epistasis while P_R is probability of restriction
      E:= rpop->getEpistasis(u);
      P_R:=E;
      r:=getRandom(0,1);
      if(r<= P_R)
            rpop:=rpop->restrictPopulation(g*,x_new[g*]);
      }
     //repleace the worst individual with the newborn individual
     w:= pop->getTheWorstIndividual();
     pop->repleaceIndividual(w,x_new);
     t:=t+1;
      }
}
```

Fig. 5. The pseudocode of EuA

If the decision about the restriction is made, each individual whose value of the gene $g^*$ is different from $x[g^*]$ is removed from the temporary population.

The process of creating a new individual described above is performed up to the point when all the genes of the individual are fixed. Once, the new individual is completely formed it replaces the worst individual from the original population and the process of creating a next new individual starts once again. It is continued until some termination criterion is satisfied.

## 5. Experiments

In the experiments, the task of GAs was to arrange ten buoys ($n = 10$) on the virtual stretch of the size $10 \times 10$. To evaluate arrangements generated during the evolutionary process, GAs used the fitness function (5) built based on hundred points ($m = 100$) uniformly distributed on the research stretch. The best arrangements produced by GAs were tested as an element of the positioning system. Each selected arrangement was tested ten times, i.e., ten different positioning systems (a single positioning system consisted of two GRNNs) were produced for each selected arrangement. The positioning systems created to test a single arrangement differed in training sets used to produce them. The training sets used in the experiments included ten, twenty, fifty or hundred training points and representing them vectors of distances ($Z$ = 10, 20, 50, or 100). Training sets generated for different arrangements differed only in vectors of distances to buoys. Points in which the vectors were recorded were the same for all tested arrangements. All the positioning systems created during the experiments were tested in hundred testing points, the same for all arrangements. To ultimately evaluate the arrangements, the following function was used:

$$E(a)=\frac{1}{TP}\sum_{i}^{P}\sum_{j}^{T}\sqrt{\left(\phi_j-\phi_j^i\right)^2+\left(\lambda_j-\lambda_j^i\right)^2}, \tag{13}$$

where: $T$ — the number of testing points ($T = 100$);
$P$ — the number of positioning systems created for a single arrangement ($P = 10$);
$\left(\phi_j,\lambda_j\right)$ — the accurate position in the $j^{th}$ testing point;
$\left(\phi_j^i,\lambda_j^i\right)$ — the position generated by the $i^{th}$ positioning system in the $j^{th}$ testing point.

In the experiments, two GAs, i.e., CGA and EuA, were used to arrange the buoys. Both GAs processed arrangements in the form of 2*$n$ size integer vectors encoded as binary strings. Each integer from the vector was encoded as binary string of size 7 (this means that chromosomes processed by GAs were of size 140 bits). Since, the size of the research stretch was 10 × 10, to create arrangements, integers from the vectors were scaled to the range <0, 10>. The remaining parameters of the evolutionary process are presented below:

— the number of individuals in population including arrangements: 100;
— the number of evolutionary generations: 50 000.

Parameters of CGA:

— crossover probability: 0.7;
— per-bit mutation probability: 0.03;
— the size of tournament: 2.

Parameters of EuA:

— selection noise: 0.01, 0.2;
— creation rate: 0.01, 0.2;
— restriction operator: on.

In the experiments, arrangements prepared by GAs were compared to random arrangements. Generally, there were tested thirty arrangements prepared by means of CGA, thirty arrangements generated by EuA, and thirty random arrangements. All the results presented in the further part of the paper are averaged.

## 5.1. Experimental results

The experiments showed that appropriate arrangement of buoys improves accuracy of the positioning system. However, as to be expected, the improvement is only noticeable for a slight number of training points used to create the system. The more training points the less influence of the arrangement of buoys on the accuracy of position fixed by the system.

TABLE 1

The results of the experiments

| | 10 training points | 20 training points | 50 training points | 100 training points | $10^4 F_e$ |
|---|---|---|---|---|---|
| | $E$ | | | | |
| CGA | 1.27 | 1.14 | 0.98 | 0.011 | 29.8 |
| EuA | 1.61 | 1.45 | 1.27 | 0.013 | 35.9 |
| random | 3.86 | 3.84 | 2.72 | 0.012 | 240.6 |

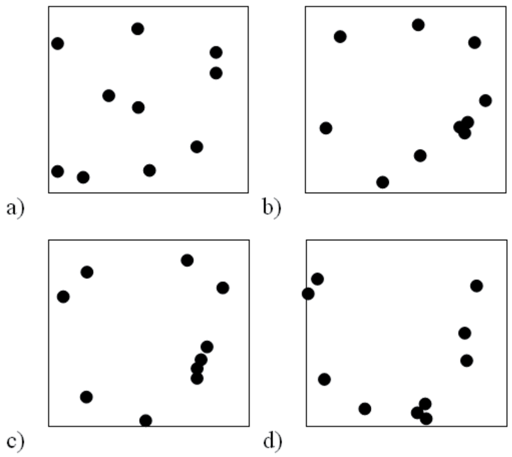

Fig. 6. Example arrangements of buoys: a) random arrangement; b), c) arrangements generated by CGA; d) arrangement generated by EuA

With regard to arrangements, it seems that there exists a noticeable difference between the random arrangements and the arrangements produced by means of GAs. While buoys from the random arrangements are usually spread on the whole research area, buoys from the arrangements produced by GAs form something like a circle. The buoys are located around the research area without any buoys in the center.

## 6. Summary

The paper presents the problem of appropriate arrangement of buoys being an element of the coastal, radar, positioning system. The system fixes position of a ship using for that purpose the information about buoys visible on the given stretch. The position of a ship is fixed by appropriately trained, approximating neural network. To make the task of the network easier, the buoys should be appropriately arranged. In the paper, the experiments are reported in which to arrange the buoys, GAs were used. The experiments showed that in the case of innumerous training sets used to train the positioning network, appropriate arrangement of buoys can improve the accuracy of the position fixed. In the case of training sets including many training samples, the effect of appropriate arrangement of buoys is less and less noticeable. However, the large training sets can also considerably slow down the positioning system making it useless.

*Received August 21 2008, revised September 2008.*

REFERENCES

[1] M. Alden, A. Van Kesteren, R. Miikkulainen, *Eugenic Evolution Utilizing a Domain Model*, in Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2002), San Francisco, CA, Morgan Kaufmann, 2002.

[2] J. Arabas, *Lectures on evolutionary algorithms*, WNT, Warsaw, 2001.

[3] D. E. Goldberg, *Genetic algorithms in search, optimization and machine learning*, Addison Wesley, Reading, Massachusetts, 1989.

[4] J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, Michigan, 1975.

[5] D. Polani, R. Miikkulainen, *Eugenic Neuro-Evolution for Reinforcement Learning*, in Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2000), Las Vegas, NV, 2000.

[6] T. Praczyk, *Application of neural networks and radar navigational aids of shore area to positioning*, Computational Methods in Science and Technology, CMST 12, 2, 2006, 53-59.

[7] T. Praczyk, *Artificial neural networks application in maritime, coastal, spare positioning system*, Theoretical and Applied Informatics, vol. 18, no. 3, 2006, 175-188.

[8] T. Praczyk, *Automatic radar navigational system*, Theoretical and Applied Informatics, vol. 18, 2006, 91-108.

[9] T. Praczyk, *Application of bearing and distance trees to the identification of landmarks of the coast*, International Journal of Applied Mathematics and Computer Science, vol. 17, 1, 2007, 87-98.

[10] T. Praczyk, *Detection of self navigational aids on radar image using ideas from immune systems*, Archives of Control Science, vol. 17, 53, no. 3, 2007, 241-259.

[11] J. W. Prior, *Eugenic Evolution for Combinatorial Optimization, Master's thesis*, The University of Texas at Austin, TR AI98-268, 1998.

[12] G. Syswerda, *Uniform Crossover in Genetic Algorithms*, In Proceedings of the 3rd International Conference on genetic Algorithms, Morgan Kaufmann, 1989.

[13] D. A. Whitley, *Genetic Algorithm Tutorial*, http://citeseer.ist.psu.edu

T. PRACZYK

## Zastosowanie algorytmów genetycznych do optymalizacji rozmieszczenia oznakowania nawigacyjnego

**Streszczenie.** Tematem artykułu jest rozmieszczenie oznakowania nawigacyjnego na akwenach przybrzeżnych. Oznakowanie nawigacyjne jest jednym z elementów automatycznego, przybrzeżnego systemu nawigacji radarowej a jego rozmieszczenie w sposób decydujący wpływa na dokładność pozycji generowanej przez system. W artykule zaproponowano użycie algorytmów genetycznych do określenia położenia poszczególnych pław na akwenie. Proponowane rozwiązanie zostało sprawdzone eksperymentalnie a wyniki testów zamieszczone zostały na końcu artykułu.

**Słowa kluczowe:** optymalizacja, algorytmy genetyczne, nawigacja radarowa
**Symbole UKD:** 621.396.96