

**Tomasz Praczyk**

**kmdr por. dr inż. Tomasz Praczyk**

Akademia Marynarki Wojennej, Wydział Nawigacji i Uzbrojenia Okrętowego

81-103 Gdynia, ul Śmidowicza 69

t.praczyk@amw.gdynia.pl

## **ZASTOSOWANIE ALGORYTMÓW GENETYCZNYCH DO KONSTRUKCJI ZADAŃ TESTOWYCH DLA SYSTEMU ANTYKOLIZYJNEGO BEZZAŁOGOWEGO POJAZDU PODWODNEGO**

*Konstrukcja niezawodnego, automatycznego systemu antykolizyjnego dla Bezzałogowego Pojazdu Nawodnego wymaga intensywnego testowania. System musi być sprawdzony w wielu różnych sytuacjach tak, aby można było stwierdzić, że jest bezpieczny i nie doprowadzi do kolizji. Tradycyjna metoda tworzenia zadań testowych polega na wykorzystaniu do tego celu człowieka. Projektant testów, bazując na swoim doświadczeniu, konstruuje kolejne testy starając się przy tym, aby utworzony przez niego zbiór zadań testowych reprezentował wszystkie możliwe sytuacje, z którymi pojazd może mieć do czynienia na morzu. Problem jednak polega na tym, że człowiek nie jest w stanie przewidzieć wszystkich możliwych sytuacji, co może skutkować nieodpowiednim przygotowaniem systemu antykolizyjnego do pracy a w konsekwencji kolizją.*

*W artykule zaproponowano inny sposób konstrukcji zadań testowych. Funkcję tą ma pełnić algorytm genetyczny, którego zadaniem jest poszukiwanie sytuacji stanowiących trudność dla systemu.*

***Słowa kluczowe:*** autonomiczny pojazd nawodny, algorytmy genetyczne, symulacje.

## WPROWADZENIE

Bezzałogowy Pojazd Nawodny zbudowany został w ramach projektu nr. O R00 0106 12 - „Bezzałogowa wielowariantowa platforma pływająca dla zabezpieczenia działań morskich służb państwowych” i jest pierwszym w kraju pojazdem typu USV (ang. Unmanned Surface Vehicle). Pojazd jest zdalnie sterowaną bezzałogową platformą nawodną, z możliwością automatycznego przejścia na sterowanie ręczne. Wyposażony został w urządzenia i sensory, pozwalające na elastyczną realizację różnego rodzaju zadań. Podstawowa wersja BPN może być rozbudowywana i dostosowywana do potrzeb odbiorcy poprzez montaż kolejnych modułów, takich np. jak zdalnie sterowany karabin maszynowy, wyrzutnia granatów głębinowych, bezzałogowy pojazd podwodny, reflektor poszukiwawczy itp. oraz specjalnych sensorów służących po pomiaru skażeń radioaktywnych i skażeń środowiska morskiego.

Obecnie, większość zadań realizowanych przez BPN odbywa się w trybie zdalnego sterowania, tzn. poprzez przekazywanie mu drogą radiową ciągu poleceń operatora, umiejscowionego w Mobilnym Stanowisku Dowodzenia, z dala od pojazdu. Takie rozwiązanie możliwe jest dzięki zastosowaniu zespołu sensorów dostarczających operatorowi informacji o stanie BPN i jego najbliższego otoczenia oraz systemowi komunikacji pomiędzy BPN a MSD, o dużej przepustowości i wysokich parametrach niezawodnościowych (system podstawowy + systemy zapasowe). Podczas konstrukcji BPN wzięto również pod uwagę możliwość awarii wspomnianego powyżej systemu komunikacji i tym samym utratę zdolności do sterowania pojazdem. Zdecydowano, że w takim przypadku BPN musi być w stanie samodzielnie przedostać się do wcześniej wyznaczonego punktu akwenu, którym może być ostatnia znana pozycja okrętu bazy wraz MSD lub inny punkt będący w zasięgu pojazdu. Aby sprostać tak postawionemu zadaniu, pojazd musi posiadać umiejętność wyznaczania optymalnej (bezpiecznej i możliwie krótkiej) trajektorii ruchu do punktu docelowego a także zdolność do unikania różnego rodzaju niebezpieczeństw (statki, jachty, zerwane pławy, itp.), które może napotkać w trakcie drogi. Jedną z koncepcji opracowanych w trakcie realizacji projektu zakłada, że uzyskanie powyższych umiejętności będzie zadaniem systemu, który na potrzeby niniejszego artykułu został nazwany Systemem Bezpiecznego Powrotu do Bazy (SBPB). Założono, że system ten składać się powinien z odbiornika GPS, systemów AIS [3] oraz ARPA [6], uproszczonej mapy elektronicznej (czarno-biała mapa rastrowa z metodami jej wykorzystania) a także komponentów decyzyjnych.

Zadaniem GPS będzie dostarczanie systemowi informacji o stanie BPN (aktualny kurs, prędkość oraz pozycja) natomiast zadaniem AIS (ang. Automatic Identification System - dokładne informacje o jednostkach występujących na akwenu, np. kurs, prędkość, pozycja, rozmiary - nie wszystkie jednostki są wyposażone w AIS co powoduje, że mogą być one niewidoczne dla BPN), ARPA (ang. Automatic Radar Plotting Aid - przybliżone informacje o obiektach widocznych przy pomocy radaru, np. kurs, prędkość, odległość, namiar) i mapy elektronicznej będzie udostępnianie danych o świecie zewnętrznym wokół pojazdu (ruch jednostek, pławy, ład, itp.). W przypadku GPS, AIS oraz ARPA wszystkie dane transmitowane będą do komputera pokładowego systemu, poprzez złącze szeregowo RS 232, tam będą rozkodowywane (dane zakodowane są w postaci zgodnej ze standardem komunikacyjnym NMEA 0183 - National Marine Electronics Association [7]) i przekazywane do części decyzyjnej systemu, za pośrednictwem pamięci wspólnej.

## **INTRODUCTION**

An Unmanned Surface Vehicle has been constructed within Project No. O R00 0106 12 entitled "A multi-task unmanned surface platform for securing operations of national marine services" and is the first such vehicle in Poland. The vehicle is a remotely controlled unmanned surface platform with the possibility of being automatically switched on to manual steering. It is equipped with devices and sensors allowing flexible realization of tasks of various kinds. The basic version of the USV may be extended and adjusted to the recipient's needs by supplying it with additional modules. For example, the USV could be modified to carry a remotely controlled machine-gun, a depth charge launcher, an unmanned underwater vehicle, a searchlight, etc., as well as special sensors for measuring radioactive and marine environment contamination.

Currently, the majority of tasks realized by the USV are executed in a remote control mode, i.e. by communicating a series of commands from an operator located at a Mobile Command Station, far from the vehicle. Such a solution is enabled by the combination of a set of sensors providing the operator with the information on the USV's condition and its surroundings, and a communication system, which utilises high reliability parameters (the basic system + supporting systems), which is capable of a handling a high rate of data flow between the USV and MCS. However, in the construction of the USV, the possibility of failure of the said communication system was considered with the consequent loss of control over the vehicle. It was decided that in such a case the USV must be able to move to a previously determined destination point on its own, for instance, to the last known position of the base ship with the MCS or another target within the vehicle's range. In order to fulfil this task the vehicle must be capable of designating the optimal (safest and shortest) movement trajectory up to the target destination, as well as avoiding various kinds of hazards (ships, yachts, floating buoys, etc.) encountered on its way.

One of the concepts worked out during the project realization assumed that obtaining the above capabilities should be the task of the system, called in this article the Safe Return to Base System (SRBS). It was assumed that the system would be composed of a GPS receiver, AIS [3] and ARPA [6] systems, a simplified electronic chart (a black-and-white raster map with the application methods), as well as decision-making components. The task of the GPS is to deliver to the system information on the USV's condition (current course, speed and position), whereas the task of the AIS (Automatic Identification System) is to provide accurate information on the other vessels present on the body of water, e.g. their course, speed, position and size. As not all vessels are equipped with the AIS, and as such are invisible to the USV, an ARPA (Automatic Radar Plotting Aid) provides approximate information (e.g. course, speed, distance, coordinates) on the objects seen with the use of the radar, and the electronic map – providing data on the vehicle's surroundings (vessel movement, floating buoys, land, etc.). In the case of the GPS, AIS and ARPA all the data are transmitted to the deck computer through a serial connection RS 232, and decoded there (data are encoded according to communication standard NMEA 0183 - National Marine Electronics Association [7]) and transmitted to the decision-making system component through shared memory. Moreover, it is assumed that the electronic chart will be included in the system as a separate programme component (language class C++) with proper auxiliary services for planning the vehicle's route and its movement around the body of water.

Zakłada się ponadto, że mapa elektroniczna zostanie włączona do systemu jako oddzielny komponent programowy (klasa języka C++) z odpowiednimi usługami pomocnymi podczas planowania trasy pojazdu a także podczas przemieszczania się po akwenuie.

Informacje dostarczane przez GPS, AIS, ARPA oraz mapę elektroniczną będą wykorzystywane przez część decyzyjną systemu do sterowania jednostką. Zakłada się, że funkcję tą będą spełniały wspólnie trzy komponenty programowe, tj. System Wyznaczania Trasy (SWT), System Przemieszczania Pojazdu po Trasie (SPPT) oraz System Antykolizyjny (SA). Zadaniem SWT będzie wyznaczenie optymalnej trasy dla BPN na podstawie informacji pozyskiwanych z mapy elektronicznej (przeszkody o charakterze stałym takie jak ląd) oraz GPS (punkt startowy trasy - pozycja BPN, w której utracono kontakt z MSD). Wyznaczona trasa będzie następnie przekazywana do SPPT, który będzie nadzorował przemieszczanie pojazdu w kierunku kolejnych punktów trasy. Punkty te będą kolejno dostarczane do SA, którego funkcją będzie sterowanie BPN w taki sposób, aby jak najszybciej dotarł on do każdego z nich jednocześnie unikając wszelkich możliwych niebezpieczeństw.

Kluczowym elementem systemu dla bezpieczeństwa pojazdu jest SA. Musi on zostać skonstruowany w taki sposób, aby możliwość wystąpienia kolizji została ograniczona do minimum. Jednym z rozwiązań jest system działający zgodnie z pewnym algorytmem określonym przez człowieka. Algorytm ten uwzględnia dane dostarczane przez sensory BPN i na ich podstawie wyznacza kurs oraz prędkość pojazdu. Innym rozwiązaniem, które może zostać zastosowane w SA jest system regułowy np. rozmyty system ekspertowy. W tym przypadku mamy do czynienia ze zbiorem reguł decyzyjnych, które określają sposób postępowania BPN. Każda z reguł w zależności od danych wejściowych charakteryzuje się pewnym stopniem zapłonu (dopasowaniem do danych wejściowych). Stopień ten określa wpływ danej reguły na ostateczny wynik działania całego systemu. Jeszcze inną możliwością w zakresie antykolizji jest sztuczna sieć neuronowa, która na podstawie sygnałów wejściowych zawierających informacje o otoczeniu pojazdu określa kurs i prędkość pojazdu.

Ponieważ SA jest elementem całego systemu, który w największym stopniu decyduje o bezpieczeństwie pojazdu, musi on zostać przygotowany z dużą starannością. Podczas, gdy w przypadku innych elementów systemu błąd popełniony na etapie ich konstrukcji może co najwyżej wpłynąć na niezdolność pojazdu do wykonywania części z jego zadań, tak w przypadku SA błąd konstrukcyjny może prowadzić do kolizji i w konsekwencji do utraty pojazdu. Aby uniknąć takiej sytuacji, każdy potencjalny system antykolizyjny przed instalacją na pojeździe powinien zostać poddany intensywnemu testowaniu w warunkach symulacyjnych. Podczas symulacji powinno być sprawdzone zachowanie się systemu na sytuacje, z którymi może się on spotkać podczas pracy w warunkach rzeczywistych. Bardzo ważne jest, aby zbiór zadań testowych wykorzystywany podczas symulacji, oprócz sytuacji typowych, uwzględniał również sytuacje, z którymi pojazd może mieć do czynienia bardzo rzadko. Jest to niezmiernie ważne, gdyż jeśli sytuacje nietypowe zostaną pominięte w fazie symulacji, to ewentualne błędy konstrukcyjne systemu i w konsekwencji nieprzygotowanie go do działania we wszystkich możliwych sytuacjach, zostanie ujawnione dopiero w momencie kolizji!. Jest tak dlatego, iż kolejny etap konstrukcji systemu czyli testy w warunkach rzeczywistych ze względów ekonomicznych (mała liczba testów przeprowadzanych zazwyczaj w sytuacjach typowych) nie są w stanie wychwycić wszystkich błędów popełnionych podczas jego budowy.

Data provided by the GPS, AIS, ARPA and the electronic chart will be used by the decision-making part of the system in controlling the vehicle. It is assumed that this function will be fulfilled collectively by three programme components, i.e. the Route Designation System (RDS), Vehicle Translocation System on the Route (VTSR) and Anti-collision System (AS). The task of the RDS is to designate an optimal route for the USV based on the information obtained from the electronic chart (depicting solid obstacles, such as the land) and GPS (route starting point – the USV's position in which the contact with MCS was lost). The designated route will then be transmitted to the VTSR which supervises the vehicle's movement towards consecutive waypoints. The waypoints will be reported one by one to the AS responsible for steering the USV in such a way as to ensure that it reaches each of them in the quickest possible manner, avoiding at the same time any possible danger.

The AS is a key system element with regard to the vehicle's safety. It must be constructed in such a way as to reduce the risk of collision to the minimum. One of the solutions designed to reduce this risk works in accordance with a certain algorithm which has been predetermined by the system's designers. The algorithm takes into consideration data supplied by the USV sensors and on their basis determines the vehicle's course and speed. Another solution that may be applied for AS is a rule-based system, e.g. a fuzzy expert system. In this case we are faced with a set of decision-making rules that determine the USV's mode of proceeding. Each of the rules, depending on input data, is characterised by a certain degree of adjustment to such data). Such a degree determines the impact of a given rule on the final performance of the entire system. Yet another possibility within anti-collision is an artificial neuron network which determines the vehicle's course and speed based on input signals containing information on its surroundings.

Since the AS is the part of the system that takes the highest responsibility for the vehicle's safety it must be prepared with great diligence. Whereas in the case of other system elements a mistake made during their construction phase may at worst have an influence on the vehicle's inability to carry out parts of its tasks, in the case of the AS a construction error may lead to a collision, and, consequently, to the vehicle's loss. In order to avoid such an event, each potential anti-collision system should undergo intense testing in simulation conditions before its installation in the vehicle. Simulations should test the system's response to the situations the vehicle may encounter in real conditions. It is crucial that the set of testing tasks used in simulations, besides typical situations, encompasses also those that the vehicle may have to deal with very rarely. This is important because, if untypical situations are omitted in the simulation phase, the possible system construction errors and, consequently, the lack of vehicle preparation to operate in all possible situations will be revealed as late as at the moment of a collision! Such rigorous testing is essential as, by the time the vehicle enters the next phase of its development, i.e. tests taking place in real conditions, economic reasons usually result in minimal testing being carried out. Compounding the issue, those tests which are carried out are usually of the more common events a vessel may find itself in and as such are not able to detect all the mistakes made during the construction.

Apart from having an effect on system efficacy, adequate preparation of the testing tasks also determines the pace of the construction work. If a significant part of the testing tasks includes more or less similar equivalents, certain system activities will be verified repeatedly, which will cause an unnecessary prolongation of the time taken to construct the final system. In order to avert this situation, the algorithm for generating testing tasks must be able to recognise tasks requiring the same skills from the system.

Odpowiednie przygotowanie zadań testowych, oprócz wpływu na skuteczność systemu, decyduje również o szybkości prac mających na celu jego wytworzenie. Jeśli znaczna część zadań testowych posiadała będzie w zbiorze zadań mniej lub więcej podobne do siebie odpowiedniki, oznaczało to będzie, że pewne zachowania systemu weryfikowane będą wielokrotnie wydłużając w ten sposób niepotrzebnie proces konstrukcji gotowego systemu. Aby tego uniknąć, algorytm generowania zadań testowych musi mieć zdolność do rozpoznawania zadań wymagających od systemu tych samych umiejętności. Niestety, najbardziej intuicyjna metoda wykrywania podobieństw, czyli porównywanie samych zadań i wyznaczanie pomiędzy nimi pewnej miary podobieństwa nie będzie w tym przypadku skuteczne. Po prostu zadania mogą się znacznie od siebie różnić parametrami jednak wywoływać taką samą reakcję systemu.

Najprostszą metodą generowania zadań testowych jest użycie do tego celu człowieka-eksperta. Wiedząc, jakie sytuacje zdarzają się na morzu często a jakie rzadko, co może stanowić dla pojazdu dużą trudność a co będzie zadaniem nie wymagającym wyrafinowanych umiejętności unikania kolizji, ekspert może wygenerować bazę zadań testowych przydatną w początkowej fazie projektowania systemu. Ponieważ nawet człowiek z dużym doświadczeniem i wiedzą nawigacyjną, podczas projektowania testów może pominąć pewne pożądane zachowania się pojazdu w sytuacjach kolizyjnych, testy przygotowane przez eksperta nie powinny stanowić ostatecznej bazy weryfikacyjnej zdolności systemu. Aby uzyskać system, który da nam gwarancję bezpieczeństwa nawet w najbardziej nieprzewidzianych sytuacjach, tworzenie testów musi zostać zautomatyzowane. Jedną z możliwości w tym zakresie jest zastosowanie algorytmów genetycznych. Algorytmy te należą do populacyjnych technik optymalizacji globalnej, które można również przystosować do roli generatora zadań testowych. Sposób wykorzystania algorytmów genetycznych w tej roli został opisany w niniejszym artykule. Oprócz części opisowej artykuł zawiera również część eksperymentalną, w której zaprezentowano wyniki zastosowania algorytmów genetycznych do poszukiwania zadań testowych niewykonalnych dla prostego algorytmicznego systemu antykolizyjnego, dla którego człowiek nie był w stanie skonstruować takich zadań.

## 1. ALGORYTMY GENETYCZNE

Algorytmy genetyczne razem z programowaniem genetycznym i strategiami ewolucyjnymi, tworzą obecnie jedną z najbardziej popularnych i szybko rozwijających się gałęzi sztucznej inteligencji. Algorytmy ewolucyjne - taka jest zbiorcza nazwa wszystkich technik wspomnianych powyżej - służą generalnie do rozwiązywania złożonych zadań optymalizacyjnych. W tym celu stosują one mechanizmy zaobserwowane w świecie naturalnym a więc selekcje i dobór naturalny, większe szanse na potomstwo i przetrwanie u osobników lepiej przystosowanych, tworzenie osobników potomnych z osobników rodzicielskich, poprzez wykonanie różnych operacji genetycznych na ich genotypie. Zazwyczaj działanie każdego algorytmu ewolucyjnego rozpoczyna się od wygenerowania losowej populacji rozwiązań (lub rozwiązań zakodowanych w postaci genotypu) pewnego problemu. W kolejnym kroku algorytmu następuje proces oceny każdego z rozwiązań. Do tego celu wykorzystuje się tzw. funkcję przystosowania, której konstrukcja jest dopasowana do zadania postawionego algorytmowi. Wartość tej funkcji dla ocenianego osobnika powinna odzwierciedlać jakość tego osobnika jako rozwiązania zadania.

Unfortunately, the most intuitive methods for similarity detection, i.e. comparing the tasks and determining the level of similarity will in this case not be effective. The tasks may simply be significantly different when considering their parameters, and yet evoke the same system responses.

The easiest method for testing task generation employ's a human. Having a good knowledge of the kinds of situations that are common or rare at sea, what may pose a significant difficulty to the vehicle and what will not require sophisticated skills for collision avoidance, this individual can generate a testing task base useful in the initial system design phase. However, since even a person with rich experience and navigational knowledge may omit certain required vehicle responses in collision situations when designing the tests, such tests should not constitute the final verification base of system performance. In order to obtain a system that would guarantee safety even in the most unpredictable circumstances, test construction must be automatized. One of the possibilities here is the application of genetic algorithms. Such algorithms belong to population-based global optimization techniques which may also be adjusted to the role of a testing task generator. This article presents the manner of using genetic algorithms in the said role, touching on both the descriptive and experimental aspects of the technique. The later part presents the results of the application of genetic algorithms to identify testing tasks that would be unfeasible for a simple algorithm anti-collision system, and which humans were incapable of constructing.

## **1. GENETIC ALGORITHMS**

Genetic algorithms, together with genetic programming and evolutionary strategies, currently constitute one of the most popular and fast developing branches of artificial intelligence. Evolutionary algorithms, as this is the collective name of all the above techniques, are generally used in solving complex optimization tasks. For this purpose they make use of mechanisms observed in the natural world, i.e. natural selection - greater chances for offspring and survival of those better adjusted, creating filial specimens from parental specimens by performing various genetic operations on their genotype. Usually, the activity of each evolutionary algorithm begins with the generation of a random population of solutions (or solutions encoded in a form of a genotype) to a given problem.

The next step of the algorithm consists of the implementation of an evaluation process for each solution. The process makes use of the so-called fitness function, whose construction is adjusted to the task presented to the algorithm. Function value, for the individual under evaluation, should reflect the individual's quality as the task solution. After evaluating all the individuals within the population there is a process of selection of parental individuals, i.e. those whose genetic material (partial solution or a code of this solution) survives in subsequent algorithm iterations. Selection of parental individuals is random; however, better individuals, characterised by a higher evaluation, are more likely to be selected than worse-evaluated ones. The selected individuals are then subject to various genetic operations as a result of which a new population of filial individuals is created. Such individuals usually replace their 'parents' in the next step of the algorithm and the entire cycle of evaluation, selection and reproduction is repeated.

Po ocenie wszystkich osobników w populacji następuje proces selekcji osobników rodzicielskich, czyli tych osobników, których materiał genetyczny (częściowe rozwiązanie lub kod tego rozwiązania) przetrwa w kolejnych iteracjach algorytmu. Wybór osobników rodzicielskich jest losowy jednak osobniki lepsze charakteryzujące się wyższą oceną posiadają większe prawdopodobieństwo wylosowania niż osobniki gorsze. Wybrane osobniki są następnie poddawane różnym operacjom genetycznym, w wyniku działania których powstaje nowa populacja osobników potomnych. Osobniki te, w kolejnym kroku algorytmu najczęściej zastępują swoich "rodziców" i cały cykl oceny, selekcji oraz reprodukcji wykonywany jest od nowa. Proces ten jest powtarzany wielokrotnie aż do uzyskania rozwiązania satysfakcjonującego lub spełnienia pewnego założonego wcześniej kryterium stopu.

```
CanonicalGeneticAlgorithm()
{
  generate initial population of random binary strings;
  evaluate population;
  do{
    select parental individuals for reproduction
      based on their fitness;
    generate offspring: apply crossover to parents;
    apply mutation to offspring;
    evaluate offspring;
    replace parents with offspring;
  }while(termination criterion is satisfied)
}
```

Rys. 1. Pseudokod kanonicznego algorytmu genetycznego [4].

Cechą charakterystyczną algorytmów genetycznych, proponowanych jako narzędzie konstrukcji zadań testowych systemu antykolizyjnego BPN, odróżniającą je od innych technik ewolucyjnych, jest postać genotypu a także operatory genetyczne, stosowane podczas tworzenia osobników potomnych. Jeśli chodzi o genotyp, to w przypadku algorytmów genetycznych mamy do czynienia z rozwiązaniami (lub częściowymi rozwiązaniami) zakodowanymi w postaci liniowych ciągów binarnych. Ciągi te w fazie reprodukcji poddawane są najczęściej dwóm podstawowym operatorom genetycznym, czyli krzyżowaniu i mutacji. Zadaniem pierwszego z nich wykonywanego bardzo często, jest wymiana kodu genetycznego pomiędzy dwoma wybranymi osobnikami rodzicielskimi. Ponieważ taka wymiana tworzy zazwyczaj osobniki bardzo odmienne od rodziców, można stwierdzić, że funkcją krzyżowania jest eksploracja nowych, niepoznanych dotąd obszarów przestrzeni rozwiązań. Z kolei, jeśli chodzi o mutację, to jest to operator wykonywany bardzo rzadko, a jego działanie polega na zamianie pojedynczego genu (bitu) w genotypie na wartość przeciwną. W odróżnieniu do krzyżowania, mutacja ma mały zasięg, operując lokalnie w pobliżu wcześniej wygenerowanych rozwiązań.



```
CanonicalGeneticAlgorithm()  
{  
  generate initial population of random binary strings;  
  evaluate population;  
  do{  
    select parental individuals for reproduction  
      based on their fitness;  
    generate offspring: apply crossover to parents;  
    apply mutation to offspring;  
    evaluate offspring;  
    replace parents with offspring;  
  }while(termination criterion is satisfied)  
}
```

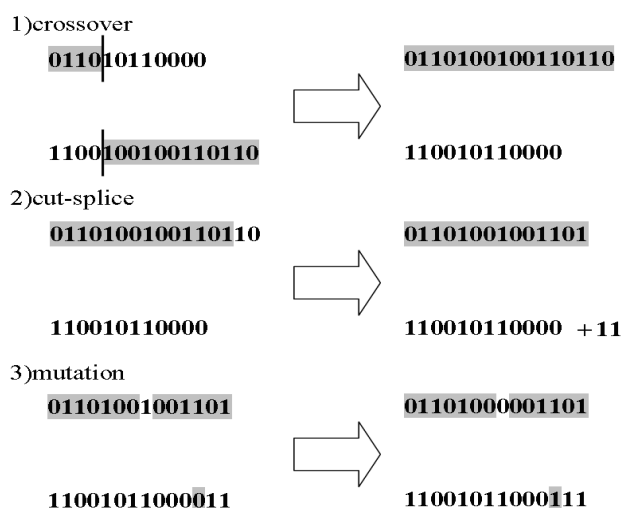
Fig. 1. Pseudo-code of a canonical genetic algorithm [4].

The process is repeated numerous until a satisfactory solution is reached or a certain, previously assumed, stop criterion is met.

A characteristic feature of the genetic algorithms proposed as a tool for the construction of the testing tasks (for the anti-collision system of the USV) is the genotype form and the genetic operators used during the creation of filial individuals. These factors distinguish these specific algorithms from the other evolutionary techniques. As regards the genotype, in the case of genetic algorithms we deal with solutions (or partial solutions) encoded in the form of linear binary strings. During the reproduction phase, the strings are usually subject to two basic genetic operations, i.e. crossover and mutation. The task of the first of them is often an exchange of the genetic code between two selected parental individuals. Because such an exchange usually produces individuals that are significantly different from the parents, we may assume that the function of a crossover consists of the exploration of new, so far unknown, solution areas. Mutations, on the other hand, are operators that are employed very rarely. Their function consists in replacing a single gene (bit) in the genotype with the opposite value. As opposed to crossover, mutation has a small range and operates only locally in the vicinity of previously generated solutions.

Besides the above basic genetic operators there are also other operators serving additional functions in genetic algorithms, such as, for instance, changing genotype length. Such an operator adds or removes from the genotype a short fragment, randomly determined, and is used in situations when a change of the quantity of solutions encoded in the genotypes is allowable.

Oprócz podstawowych operatorów genetycznych opisanych powyżej, istnieją również inne operatory spełniające w algorytmach genetycznych dodatkowe funkcje, takie jak na przykład zmiana długości genotypu. Operator ten dodaje lub usuwa z genotypu fragment o niewielkiej losowo określonej długości i wykorzystywany jest w sytuacjach, kiedy dopuszczalna jest zmiana wielkości rozwiązań zakodowanych w genotypach.



Rys. 2. Trzy podstawowe operatory genetyczne stosowane w algorytmach genetycznych [4].

## 2. ZASTOSOWANIE ALGORYTMÓW GENETYCZNYCH DO KONSTRUKCJI ZADAŃ TESTOWYCH

W celu zastosowania algorytmów genetycznych do dowolnego problemu optymalizacyjnego, konieczne są trzy rzeczy. Pierwszą z nich jest wybór konkretnego algorytmu, który chcemy zastosować. Jest wiele odmian algorytmów genetycznych różniących się takimi elementami jak na przykład operatory genetyczne (np. krzyżowanie jednopunktowe, dwupunktowe), sposób selekcji osobników rodzicielskich (np. selekcja turniejowa, ruletkowa), sposób tworzenia populacji potomnej (populacja rodzicielska w całości zastępowana jest populacją potomną, w jednej iteracji algorytmu tworzony jest tylko jeden osobnik potomny, który zastępuje jeden z osobników z aktualnie przetwarzanej populacji) czy też schemat ewolucji (np. ewolucja w jednej populacji lub w wielu populacjach, czyli tzw. ko-ewolucja). Każdy z algorytmów ma dobrze znane wady i zalety, które pozwalają na wybór jednego z nich do rozwiązywanego zadania.

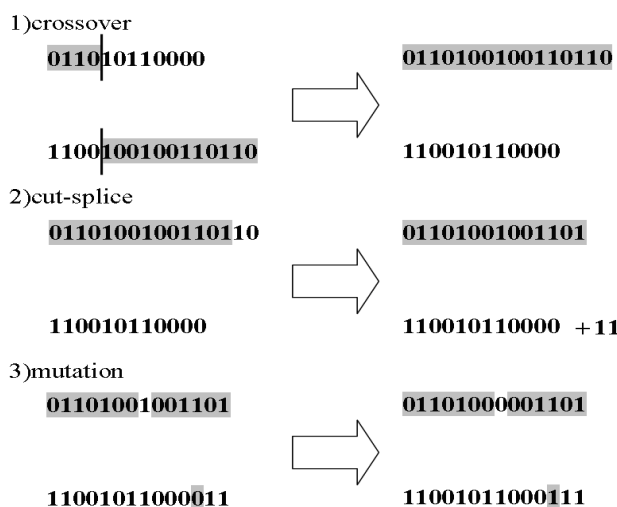


Fig. 2. Three basic genetic operators used in genetic algorithms [4].

## 2. USING GENETIC ALGORITHMS TO CONSTRUCT TESTING TASK

In principle, in order to apply genetic algorithms in relation to a randomly selected optimization problem, we need to undertake three actions. The first of them is to select a concrete algorithm that we wish to apply. There are many varieties of genetic algorithms distinguishable by such elements as genetic operators (e.g. one-point or two-point crossover), the manner of selecting parental individuals (e.g. tournament, roulette selection), the way of creating filial population (parental population is replaced completely with filial population, one algorithm iteration creates one filial individual to replace one individual from the currently transformed population) or an evolutionary scheme (e.g. evolution within one or many population, the so-called co-evolution). Each algorithm has well-known advantages and disadvantages that allow for selecting one of them for solving a given task.

The next thing that needs to be determined before a genetic algorithm is activated are the algorithm parameters. The number and type of these parameters depend on a particular evolutionary technique, however usually they include: the number and length of individuals (genotypes); crossover and mutation probability and maximum iteration number. Parameter values depend mainly on the task we are trying to solve, and the most common way of their determination relies on the trial and error method.

The last thing that is to be defined before we apply a genetic algorithm is the fitness function. As mentioned before, this function reflects the target at which the genetic algorithm should aim. It should be construed in such a way as to lead the algorithm towards better and better solutions with each iteration, indicating the search areas with high fitness values.

Drugą z rzeczy, którą należy określić zanim uruchomimy algorytm genetyczny są parametry algorytmu. Liczba i typ parametrów zależą od konkretnej techniki ewolucyjnej, jednak najczęściej są to takie parametry jak: liczba i długość osobników (genotypów), prawdopodobieństwa krzyżowania i mutacji, maksymalna liczba iteracji. Wartości parametrów zależą głównie od zadania, które rozwiązujemy a najczęstszym sposobem ich ustalania jest metoda prób i błędów.

Ostatnią z rzeczy, którą należy zdefiniować zanim zastosujemy algorytm genetyczny jest funkcja przystosowania. Tak jak zostało to już wcześniej wspomniane funkcja ta odzwierciedla cel, do którego powinien dążyć algorytm genetyczny. Powinna być ona tak skonstruowana, aby w kolejnych iteracjach prowadzić algorytm w kierunku coraz lepszych rozwiązań wskazując mu obszary przestrzeni poszukiwań o wysokich wartościach przystosowania.

W przypadku generowania zadań testowych dla systemu antykolizyjnego, algorytm genetyczny może zostać ukierunkowany w dwojaki sposób. Pierwszy z nich polega na poszukiwaniu zadań testowych, które stanowią trudność dla istniejącego systemu antykolizyjnego. Podczas kolejnych iteracji ewolucyjnych algorytm konstruuje binarnie zakodowane zadania testowe np. w postaci wektora parametrów ruchu jednostek występujących na akwenie razem z BPN, które są następnie oceniane pod względem trudności, którą sprawiają systemowi antykolizyjnemu. Algorytm uzyskując informację o tym, które z testów są trudniejsze dla systemu a które nie sprawiają mu żadnych trudności stara się tworzyć coraz trudniejsze testy. Miarą trudności testu może być informacja czy system doprowadził do kolizji czy nie. W przypadku braku kolizji, pomiar trudności testu może odbywać się w oparciu o czas, który był potrzebny pojazdowi na dotarcie do celu. Przykładowa funkcja przystosowania może w tym przypadku wyglądać następująco:

$$F(t) = \begin{cases} 100 + 1/CK & \text{(a)} \\ 10 & \text{(b)} \\ CD/MC & \text{(c)} \end{cases} \quad (1)$$

3.0

gdzie:

$t$  – test;

$CK$  – liczba jednostek czasu, po których nastąpiła kolizja;

$MC$  – maksymalna liczba jednostek czasu, którą dysponuje pojazd w trakcie testów na dotarcie do celu;

$CD$  – liczba jednostek czasu potrzebna na dotarcie do celu;

(a) – sytuacja, kiedy w trakcie testu nastąpiła kolizja;

(b) – sytuacja, kiedy w trakcie testu nie było kolizji ale pojazd nie dotarł do celu w wymaganym czasie;

(c) – sytuacja, kiedy pojazd dotarł do celu.

In the case of generating testing tasks for an anti-collision system, the genetic algorithm may be directed in two ways. The first option is based on searching testing tasks which pose a certain difficulty for the existing anti-collision system. During subsequent evolutionary iterations, the algorithm constructs binary encoded testing tasks, for instance in the form of vector of movement parameters of the vessels present on the body of water together with the USV, which are then evaluated with regard to the level of difficulty that the anti-collision system is faced with. Thus provided with the information on which of the tests are more difficult for the system, and which do not cause any difficulty, the algorithm attempts to create more difficult tests.

The measurement of test difficulty may be determined in a variety of ways. For example, in the instance of there being no collision, test difficulty measurement may be based on the time needed by the vehicle in order to reach the destination point. An exemplary fitness function may in such a case be presented as follows:

$$F(t) = \begin{cases} 100 + 1/CK & \text{(a)} \\ 10 & \text{(b)} \\ CD/MC & \text{(c)} \end{cases} \quad (1)$$

3.0

where:

$t$  – test

$CK$  – number of time units after which the collision took place

$MC$  – maximum number of time units that the vehicle has during tests in order to reach the destination point

$CD$  – number of time units needed to reach the destination point

(a) – a situation during the test in which the collision took place

(b) – a situation during the test in which there was no collision, however the vehicle failed to reach the destination point in the required time

(c) – a situation in which the vehicle reached the destination point.

The mode of test generation directed by the function (1), however, has one flaw. Namely, it causes redundant tasks to appear in the testing task set. If it turns out that in a certain test area (binary vectors, including movement parameters, of the vessels present on the body of water near the USV) there are tests that the anti-collision system is not able to pass without a collision, in subsequent evolutionary cycles this area is bound to be highly represented in the test population created by the algorithm. It may even happen that the algorithm will focus on one area with a high fitness value (difficulty level) omitting other areas, thus leading to a situation where the testing task set is limited to tests of a similar profile.

Sposób generowania testów ukierunkowany przez funkcję (1) posiada jedną wadę. Mianowicie powoduje on, że w zbiorze zadań testowych będą pojawiały się zadania redundantne. Jeśli okaże się, że w pewnym obszarze przestrzeni testów (wektorów binarnych zawierających parametry ruchu jednostek występujących na akwenu w pobliżu BPN) występują testy, których system antykolizyjny nie jest w stanie zaliczyć bez kolizji, to w kolejnych cyklach ewolucyjnych obszar ten będzie bardzo licznie reprezentowany w populacji testów, tworzonej przez algorytm. Może się nawet okazać, że algorytm skupi się w jednym obszarze o wysokim przystosowaniu (stopniu trudności), pomijając inne obszary i doprowadzając do sytuacji, w której zbiór zadań testowych zostanie ograniczony do testów o podobnej charakterystyce.

W celu uniknięcia takiej sytuacji, można zastosować dwa proste zabiegi. Pierwszym z nich jest wielokrotne uruchamianie ewolucji i budowa zbioru zadań testowych, w oparciu o końcowe populacje testów z wielu uruchomień. Ponieważ taki zabieg może spowodować nadmierny rozrost zbioru zadań testowych, dodatkowo konieczna jest modyfikacja funkcji przystosowania, poprzez zastosowanie tzw. techniki "crowding-u" czyli redukcji przystosowania testów, w przypadku, kiedy w populacji występują inne, podobne do nich testy. Może to formalnie wyglądać w następujący sposób:

$$F_1(t) = \frac{F(t)}{\sum_i^n P(i,t)} \quad (2)$$

$$P(i,t) = \begin{cases} 0 & \textit{i niepodobny do t} \\ 1 & \textit{i podobny do t} \end{cases} \quad (3)$$

gdzie:

$F_1$  – zmodyfikowana funkcja przystosowania;  
 $n$  – liczba testów w populacji.

Problemem w tym przypadku jest określenie podobieństwa pomiędzy testami. Zwyczajna odległość Euklidesowa pomiędzy wektorem parametrów definiującym test, jest w tym przypadku nieskuteczna. Na przykład wektor 0|0.5|0|5|180|2|180|20 gdzie na odpowiednich pozycjach są parametry dwóch obiektów czyli NR1|D1|KR1|V1|NR2|D2|KR2|V2 (NR - zamiar na obiekt, D - odległość od obiektu, KR - kurs obiektu, V - prędkość obiektu) i wektor 180|2|180|20|0|0.5|0|5 charakteryzują się znaczną odległością Euklidesową, jednak reprezentują taki sam test. Każdy z nich zawiera definicję tych samych dwóch jednostek, jednak w innej kolejności.

Innym sposobem określania podobieństwa pomiędzy testami może być porównanie trajektorii przemieszczania się lub kolejnych decyzji pojazdu uzyskanych dla danego testu. W tym przypadku otrzymamy wektory kolejnych pozycji na akwenu, w których znajdował się pojazd lub wektory kursów i prędkości wyznaczonych w kolejnych chwilach czasowych symulacji przez system antykolizyjny.

In order to avoid such a situation we may apply two simple procedures. The first of them rests in multiple evolution activation and creating a testing task set based on final test populations from numerous activations. Since this may cause an excessive development of the testing task set, it is necessary to additionally modify the fitness function by applying the so-called 'crowding' technique, i.e. test fitness reduction in cases where the population contains other similar tests. Formally speaking, this may be presented as follows:

$$F_1(t) = \frac{F(t)}{\sum_i^n P(i,t)} \quad (2)$$

$$P(i,t) = \begin{cases} 0 & i \text{ not similar to } t \\ 1 & i \text{ similar to } t \end{cases} \quad (3)$$

where:

$F_1$  – a modified fitness function  
 $n$  – number of tests within population.

The problem here consists in determining the similarity level between tests. In this case, the standard Euclidean distance between the parameter vector defining the test is ineffective. For example, vector 0|0.5|0|5|180|2|180|20 with parameters of two objects indicated in proper positions, i.e. NR1|D1|KR1|V1|NR2|D2|KR2|V2 (NR – object's position, D – distance from the object, KR – object's course, V – object's speed) and vector 180|2|180|20|0|0.5|0|5 are characterised by high Euclidean distance, however they represented the same test. Each of them contains definitions of two vessels shown in a different order.

Another way of determining similarity between tests may rest in comparing the movement trajectory or consecutive decisions of the vehicle obtained for a given test. In this case we will receive vectors of subsequent positions of the vehicle on the body of water or course and speed vectors set by the anti-collision system for particular simulation moments. The tests of a low Euclidean distance between the said vectors will be considered as similar.

The above method of application of genetic algorithms in creating test tasks for anti-collision systems has one defect. As it is mainly focused on test difficulty it will tend to omit simple tests. In the situation when the creation of the anti-collision system is carried out in an automatic way, without the help of an expert, i.e. the system will be a random learning system, e.g. a neuron network, the omission of 'simple' tasks in the test task set will lead to insufficient training of such a system and, consequently, to possible errors in the situations which, at first glance, do not require great skills in vehicle manoeuvring; this results from the fact that each learning system requires in its preparation phase a full review of situations it may encounter during the operation. Lack of certain tests in the testing task set that may seem needless to an expert as, for instance, they concern situations in which the anti-collision system is not necessary and should not be activated to modify the vehicle's movement parameters, may result in the "learning" anti-collision system not 'knowing' how to proceed. In order to function properly it must be able to independently obtain the knowledge on how to act in various situations, also those that seem banal to the expert.

Testy o małej odległości Euklidesowej pomiędzy opisanymi powyżej wektorami, będą uznawane za podobne.

Metoda użycia algorytmów genetycznych do konstrukcji zadań testowych dla systemu antykolizyjnego opisana powyżej ma jedną wadę. Kierując się przede wszystkim trudnością testów, będzie ona pomijała testy proste. W sytuacji, kiedy tworzenie systemu antykolizyjnego odbywało się będzie nie z udziałem eksperta a w sposób automatyczny, tzn. systemem tym będzie dowolny system uczący, np. sieć neuronowa, pominięcie w zbiorze zadań testowych zadań "prostych" prowadziło będzie do niedostatecznego wytrenowania takiego systemu i w konsekwencji do możliwych błędów w sytuacjach, które na pierwszy rzut oka nie wymagały będą dużych zdolności w manewrowaniu pojazdem. Wynika to z tego, iż każdy system uczący, podczas procesu przygotowania go do pracy wymaga pełnego przeglądu sytuacji, z którymi może się zetknąć w trakcie działania. Brak pewnych testów w zbiorze zadań testowych (testów, które dla eksperta mogą się wydawać zbędne ze względu na przykład na fakt, iż dotyczą one sytuacji, w których system antykolizyjny jest niepotrzebny i nie powinien w jakikolwiek sposób zmieniać parametrów ruchu pojazdu) może spowodować, że antykolizyjny system uczący nie będzie "wiedział" jak w takich sytuacjach postępować. Aby mógł poprawnie funkcjonować, musi samodzielnie pozyskać wiedzę o tym, jak działać w najróżniejszych sytuacjach, również tych, które dla eksperta mogą się wydawać banalnie proste.

Aby ukierunkować algorytm genetyczny na tworzenie zróżnicowanych testów, konieczne jest zastosowanie innej funkcji przystosowania niż funkcje (1) i (2). W tym przypadku, ważne musi być nie poszukiwanie testów trudnych, ale testów innych od tych wygenerowanych dotychczas. W tym celu konieczne jest zastosowanie specjalnego repozytorium, które będzie przechowywało testy wzbogacające wiedzę przyszłego systemu antykolizyjnego o tym, z czym może mieć on do czynienia na morzu. Funkcja przystosowania wymuszająca na algorytmie genetycznym poszukiwanie nowych, jeszcze nieznanymi sytuacji testowych, dla systemu antykolizyjnego może wyglądać w następujący sposób:

$$F_2(t) = \frac{mk}{\sum_i^m d(i,t)} \quad (4)$$

gdzie;

$m$  - liczba testów w repozytorium;

$k$  - długość wektorów charakteryzujących test;

$d$  - odległość Euklidesowa pomiędzy wektorami reprezentującymi testy  $i$ -ty (z repozytorium) oraz  $t$ -ty (wszystkie składowe wektorów powinny być znormalizowane do przedziału  $\langle 0,1 \rangle$ ).



In order to direct the genetic algorithm towards generating diversified tests, it is necessary to apply a different fitness function from those of the functions (1) and (2). What is important here is not searching for difficult tests but tests different from the ones generated so far. For this purpose it is substantial to use a special repository to store the tests enriching the knowledge of the future anti-collision system with what it may encounter at sea. The fitness function forcing the generic algorithm to seek new, still unknown, test situations for the anti-collision system may be represented in the following way:

$$F_2(t) = \frac{mk}{\sum_i^m d(i,t)} \quad (4)$$

where:

- m* – the number of tests in the repository
- k* – length of vectors characterising the test
- d* – Euclidean distance between vectors representing *i*- (from the repository) and *t*-tests (all vector components should be standardized to the range <0,1>).

### **3. EXPERIMENTS**

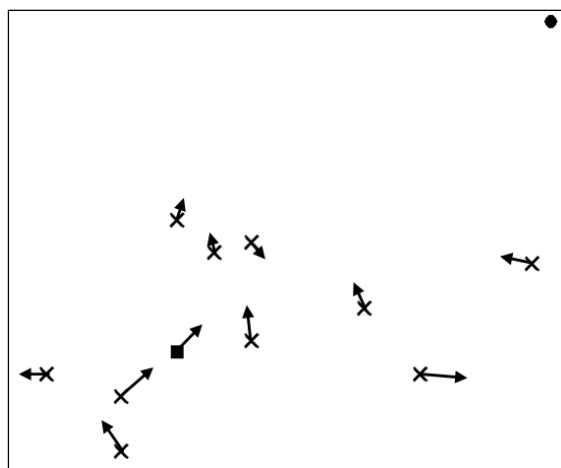
The works realized within the project No. O R00 0106 12 allowed preparation and implementation of a number of variants of an USV anti-collision system. One of them was a simple algorithm system whose operation depends on adjusting the USV's course (with constant vehicle speed equal to 10 knots) to the anticipated behaviour of other vessels present on the body of water, estimated on the basis of information obtained from ARPA and AIS [5]. The efficiency of this system was verified based on 110 various testing tasks prepared by a human. The tests showed that the said system was able to correctly manoeuvre the vehicle, leading it to the destination point in all 110 tasks. In order to ensure system capabilities, an attempt was made to work out additional tasks that would reveal any possible shortcomings and construction errors of the system. When designing additional tests the focus was placed on making tasks as difficult as possible for the system. However, the efforts of finding the system's weak points through new tasks proved to be of no effect. During additional tests the system once again proved to be correct in the operation. All additional tasks prepared by a human were fulfilled safely, without leading to a collision. In order to be absolutely certain of the system's capabilities and obtain final confirmation of its effectiveness it was decided to check it on the basis of the tests prepared by a genetic algorithm. This was done, first of all, to check whether the algorithm would find any tasks that would be unfeasible for the system. A simple canonical algorithm [1,2] of the following parameters was used for this purpose:

- number of individuals in the population = 100
- mutation probability = 0.001
- crossover probability = 0.7
- maximum iteration number = 60000
- chromosome length = 320 bits
- fitness function = function (1).

### 3. EKSPERYMENTY

W ramach prac realizowanych w ramach projektu O R00 0106 12 opracowano i zaimplementowano szereg wariantów systemu antykolizyjnego BPN. Jednym z nich był prosty system algorytmiczny, którego działanie polega na dopasowaniu kursu BPN (przy stałej prędkości pojazdu równej 10 węzłów) do przewidywanego zachowania innych jednostek występujących na akwenu estymowanego na podstawie informacji uzyskiwanych z ARPA i AIS [5]. Efektywność tego systemu została zweryfikowana w oparciu o 110 różnorodnych zadań testowych, opracowanych przez człowieka. Podczas testów okazało się, że wspomniany system bezbłędnie manewruje jednostką, doprowadzając ją bezpiecznie do celu we wszystkich 110 zadaniach. Aby upewnić się, co do zdolności systemu podjęto próbę opracowania dodatkowych zadań, które wykryłyby jego ewentualne niedociągnięcia i błędy konstrukcyjne. Podczas projektowania dodatkowych testów dążono do tego, aby maksymalnie utrudnić zadanie systemowi. Próby wykrycia słabych punktów systemu poprzez dodatkowe zadania okazały się jednak bezowocne. Podczas dodatkowych testów system ponownie okazał się bezbłędny w swoim działaniu. Wszystkie dodatkowe zadania opracowane przez człowieka zostały przez niego wykonane bezkolizyjnie. Aby uzyskać całkowitą pewność, co do umiejętności systemu i ostatecznie potwierdzić jego skuteczność postanowiono sprawdzić go w oparciu o testy przygotowane przez algorytm genetyczny. Decydując się na zastosowanie algorytmu genetycznego chodziło przede wszystkim o to, czy uda się mu znaleźć jakiegokolwiek zadanie niewykonalne dla systemu. Do tego celu wykorzystano prosty algorytm kanoniczny [1,2] o następujących parametrach:

- liczba osobników w populacji = 100
- prawdopodobieństwo mutacji = 0.001
- prawdopodobieństwo krzyżowania = 0.7
- max liczba iteracji = 60000
- długość chromosomów = 320 bitów
- funkcja przystosowania = funkcja (1).



Rys. 3 Przykładowy test wygenerowany przez algorytm genetyczny, w którym system antykolizyjny doprowadził do kolizji (kwadrat – początkowa pozycja BPN, koło – pozycja docelowa BPN, krzyżyki – pozycje początkowe innych obiektów, strzałki – kursy i prędkości).

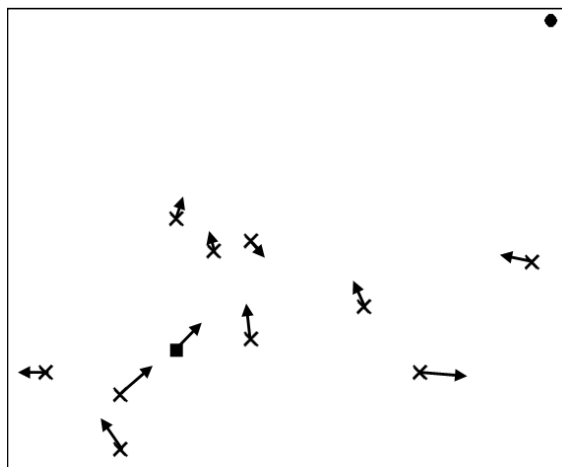


Fig. 3 Sample test generated by genetic algorithm in which the anti-collision system led to a collision (square – USV's initial position, circle – USV's destination point, crosses – initial positions of other objects, arrows – courses and speeds).

Defining a single test (the test was located in a single chromosome) required determining the initial positions (initial coordinates and distance from the USV) as well as movement parameters (course, speed) of 10 objects present on the body of water together with the USV. An additional assumption for every test was also a limited speed of each object, i.e. maximum 15 knots (the maximum speed of the USV during tests reached 20 knots). The USV's initial position and its destination point were more or less 2 nautical miles apart and were predetermined.

As the experiment showed, the genetic algorithm was able to find tasks that the system could not successfully complete in each of its activations. A sample task generated by the genetic algorithm in which the system led to a collision is presented in Fig. 3.

In the end, the application of the genetic algorithm resulted in 60 new test tasks that the algorithmic anti-collision system was not able to complete. The analysis of tasks and mistakes made by the system facilitated the working out of a new improved version of the anti-collision system which managed to successfully complete all the tests designed both by humans and the genetic algorithm.

#### **SUMMARY**

The USV anti-collision system is a key element of the vehicle when it is working in an autonomous mode. In order to ensure that the system will be able to correctly fulfil its tasks, guaranteeing the safe transfer to the destination point, it is necessary to carry out numerous tests in simulative conditions. As it was proved by the experiments presented in this article, tests verifying system efficiency should be designed both by humans and with the use of automatic methods, such as genetic algorithms. As opposed to humans, those methods are far more effective in finding the weak points of the system thus enabling their elimination and, consequently, creating a system that guarantees real safety.

**Scientific work financed from educational funds in the years 2010 – 2012 as a development project No. O R 00 0106 12.**

Definicja pojedynczego testu (test znajdował się w pojedynczym chromosomie) wymagała określenia początkowej pozycji (początkowy namiar i odległość od BPN) oraz parametrów ruchu (kurs, prędkość) dla 10 obiektów występujących razem z BPN na akwenu. Dodatkowym założeniem dla każdego z testów była również ograniczona prędkość każdego z obiektów, która wynosiła maksymalnie 15 węzłów (prędkość maksymalna BPN w trakcie testów wynosiła 20 węzłów). Pozycje początkowa BPN oraz punktu docelowego oddalone były od siebie o mniej więcej 2 mile morskie i były wcześniej zadane.

Jak się okazało, algorytm genetyczny w każdym uruchomieniu potrafił znaleźć zadania, których system nie potrafił wykonać bezbłędnie. Przykładowe zadanie wygenerowane przez algorytm genetyczny, w którym system doprowadził do kolizji jest przedstawione na Rys. 3.

Ostatecznie, w wyniku zastosowania algorytmu genetycznego uzyskano 60 nowych zadań testowych, których algorytmiczny system antykolizyjny nie potrafił wykonać. Analiza zadań oraz błędów popełnionych przez system pozwoliła na opracowanie nowej udoskonalonej wersji systemu antykolizyjnego, która przeszła pozytywnie wszystkie testy zaprojektowane zarówno przez człowieka jak i przez algorytm genetyczny.

#### **PODSUMOWANIE**

System antykolizyjny BPN jest kluczowym elementem pojazdu w sytuacji, kiedy pracuje on w trybie autonomicznym. Aby system ten mógł bezbłędnie wypełniać swoje zadanie dając nam gwarancję tego, iż w najróżniejszych sytuacjach w sposób bezpieczny przeprowadzi pojazd do celu, konieczne jest jego intensywne testowanie w warunkach symulacyjnych. Jak wykazały eksperymenty przedstawione w artykule, do projektowania zadań testowych weryfikujących sprawność systemu oprócz człowieka powinny być również wykorzystane metody automatyczne takie jak na przykład algorytmy genetyczne. W przeciwieństwie do człowieka potrafią one ze znacznie większą skutecznością znaleźć słabe punkty systemu pozwalając w ten sposób na ich wyeliminowanie i w konsekwencji na budowę naprawdę bezpiecznego systemu.

**Praca naukowa finansowana ze środków na naukę w latach 2010 - 2012 jako projekt rozwojowy Nr O R 00 0106 12.**

#### LITERATURA/ BIBLIOGRAPHY

1. Goldberg D. E.: "Genetic algorithms in search, optimization and machine learning", Addison Wesley, Reading, Massachusetts, 1989;
2. Holland J. H.: "Adaptation in Natural and Artificial Systems", University of Michigan Press, 1975;
3. Lee A., Zetterberg S.: "Establishing an IALA AIS Binary Message Register: Recommended Process", IALA Conference 17, 108–115, 2010;
4. Praczyk T.: "Using Assembler Encoding to Construct Artificial Neural Networks with a Modular Architecture", Rozprawa habilitacyjna, Wydawnictwo Akademii Marynarki Wojennej, 2011;
5. Praczyk T.: "Classical Anti-collision System of Unmanned Surface Vehicle", Materiały konferencyjne: XVI International School of Computer-aided Designing, Producing and Exploitation, Jurata, 2012;
6. Wawrach R.: „ARPA zasada działania i wykorzystania”, WSM Gdynia, ISBN 83-86703-59-8, 2002;
7. Publications and Standards from the National Marine Electronics Association (NMEA) / NMEA 0183.

### **USE GENETIC ALGORITHMS TO CONSTRUCT TESTING TASKS FOR THE ANTI - COLLISION SYSTEM OF AN UNMANNED SURFACE VEHICLE**

*To build a reliable automatic anti-collision system for an Unmanned Surface Vehicle it is necessary to implement an intensive testing procedure. In order for the system to guarantee safety at sea it has to be verified in many different situations. The traditional method for building such test tasks uses a test designer to create tests based on his or her experience; ideally the complete test set would represent all possible situations that the vehicle may face at sea. However, the problem is that a human cannot predict all possible situations, a flaw which may result in an inappropriate preparation of the anti-collision system and, in consequence, a collision. The following paper proposes another method for constructing testing tasks, a method that utilises a genetic algorithm. The algorithm's aim is to search for situations which may be difficult for the system or situations which are completely different from the ones tested so far.*

**Key words:** *unmanned surface vehicle, genetic algorithms, simulations.*

## **ПРИМЕНЕНИЕ ГЕНЕТИЧЕСКИХ АЛГОРИТМОВ ДЛЯ КОНСТРУКЦИИ ТЕСТОВЫХ ЗАДАНИЙ ДЛЯ СИСТЕМЫ ПРЕДОТВРАЩЕНИЯ СТОЛКНОВЕНИЙ НАДВОДНЫХ ТРАНСПОРТНЫХ СРЕДСТВ**

*Конструкция надежной, автоматической системы предотвращения столкновений для надводного беспилотного транспортного средства требует интенсивного тестирования. Система должна быть проверена в различных ситуациях так, чтобы можно было сказать что она является безопасной и не приведет к столкновениям. Традиционный метод создания тестовых заданий состоит в использовании для этой цели человека. Конструктор тестов, основываясь на своем опыте, разрабатывает дальнейшие испытания, стараясь при этом, чтобы созданный им набор тестовых заданий представлял все возможные ситуации, с которыми транспортное средство может иметь дело на море. Проблема, однако, заключается в том, что человек не в состоянии предусмотреть все возможные ситуации, что может привести к неправильной подготовке системы предотвращения столкновений для работы, а в результате к столкновению. В статье предложен другой способ конструкции тестовых заданий. Эту функцию может выполнять генетический алгоритм, задачей которого является поиск ситуации, которые являются трудными для системы.*

**Ключевые слова:** надводное автономное средство передвижения, генетические алгоритмы, симуляция.