Path Planning in Dynamic Environments

Roman Śmierzchalski[1], Zbigniew Michalewicz[2]

[1] Gdynia Maritime University, Faculty of Marine Electrical Engineering, Morska 83,
81-225 Gdynia, Poland

[2] Department of Computer Science, University of North Carolina,
Charlotte, NC 28223, USA and Institute of Computer Science,
Polish Academy of Sciences, Ordona 21, 01-237 Warsaw, Poland

Abstract. The *motion planning* problem for mobile robots is typically formulated as follows: given a robot and a description of an environment, plan a path of the robot between two specified locations, which is collision-free and satisfies certain optimization criteria. Traditionally there are two approaches to the problem: Off-line planning, which assumes perfectly known and stable environment, and on-line planning, which focuses on dealing with uncertainties when the robot traverses the environment. On-line planning is also referred to by many researchers as the *navigation* problem. Additional difficulties in approaching navigation problem is that some environments are dynamic, i.e., the obstacles which are present there, need not be static. In this paper we consider a particular instance of a navigation problem, namely, a problem of computing a near-optimum trajectory of a ship. By taking into account certain boundaries of the maneuvering region, along with navigation obstacles and other moving ships, the problem of avoiding collisions at sea was reduced to a dynamic optimization task with static and dynamic constrains. The paper presents a modified version of the Evolutionary Planner/Navigator algorithm, ϑEP/N++, to address the problem. The introduction of a time parameter, the variable speed of the ship, and time-varying constraints representing movable ships, are the main features of the new system. Sample results, having the form of ship trajectories obtained using the program for navigation situations are also presented.

1 Introduction

The *motion planning* problem for mobile robots is typically formulated as follows [19]: given a robot and a description of an environment, plan a path of the robot between two specified locations, which is collision-free and satisfies certain optimization criteria. Traditionally there are two approaches to the problem: Off-line planning, which assumes perfectly known and stable environment, and on-line planning, which focuses on dealing with uncertainties when the robot traverses the environment. On-line planning is also referred to by many researchers as the *navigation* problem.

A great deal of research has been done in motion planning and navigation (see [19] and [6] for surveys). However, different existing methods encounter one or many of the following difficulties:

- high computation expenses,
- inflexibility in responding to changes in the environment,
- inflexibility in responding to different optimization goals,
- inflexibility in responding to uncertainties,
- inability to combine advantages of global planning and reactive planning.

In order to address these difficulties, we initiated the study of an Evolutionary Planner/Navigator (EP/N) system [7, 18, 17]; the inspiration to use evolutionary techniques was triggered by the following ideas/observations:

- randomized search can be the most effective in dealing with NP-hard problems and in escaping local minima,
- parallel search actions not only provide great speed but also provide ground for *interactions* among search actions to achieve even greater efficiency in optimization,
- intelligent behavior is the result of a collection of simple reactions to a complex world,
- a planner can be greatly simplified, much more efficient and flexible, and increase the quality of search, if search is not confined to be within a specific map structure,
- it is more meaningful to equip a planner with the flexibility of changing the optimization goals than the ability of finding the absolutely optimum solution for a single, particular goal.

The EP/N embodied the above ideas by incorporating some problem specific knowledge into evolutionary algorithm. With such an approach, the EP/N is pursuing all the advantages as described above. Less obvious though, is that with the unique design of chromosome structure and genetic operators, the EP/N does not need a discretized map for search, which is usually required by other planners. Instead, the EP/N "searches" the original and continuous environment by generating paths by various evolutionary operators. The objects in the environment can simply be indicated as a collection of straight-line "walls". This representation accommodates both known objects as well as partial information of unknown objects obtained from sensing. Thus, there is little difference between off-line planning and on-line navigation for the EP/N. In fact, the EP/N unifies off-line planning and on-line navigation with the same evolutionary algorithm and chromosome structure.

In this paper we discuss a generalization of EP/N: its version, called ϑEP/N++, to address additional issues present in dynamic environments. The introduction of a time parameter, the variable speed of the ship, and time-varying constraints representing movable ships, are the main features of this version. The system was tested for particular environments: navigation of ships in collision situations.

When determining a safe trajectory for so-called *own ship*, we look for a trajectory that balances the cost of necessary deviation from a given route, or from the optimum route leading to a destination point, and the safety of passing all static and dynamic obstacles, called here *strange ships (or targets)*. In this paper the following terminology is used: the term *own ship* means the ship, for which the trajectory must be generated, and *strange ship* or *target* mean other ships in the environment, which must be avoided. All trajectories, which meet safety conditions (thus the risk of collision is reduced to a satisfactory degree) constitute a set of permissible trajectories. The safety conditions are, as a rule, defined by the operator, based on the speed ratio between the ships involved in the passing maneuver, as well as the actual visibility, weather conditions, navigation area, maneuverability of the ship, etc. The simplest way of determining the safe trajectory seems to use additional device — a decision supporting system, which would make an extension of the conventional Automatic Radar Plotting Aids (ARPA) system.

1.1 Previous work

Dove et al. [2] presented a guidance concept for a ship entering a harbor, in which two autonomous systems (VTS and the system on board of the ship) were applied for evaluating the trajectory along given seaways, with simultaneous evaluation of the time correlation of subsequent positions of the ship, taking into account dynamic characteristics of the ship. The guidance principles were defined using an adaptation multi-dimensional optimum controller basing on the square quality factor, and taking into account both the minimization of the actual position and course deviation for a given time instant, and the minimization of the overall economic costs of guidance, represented by rudder positions and engine activity. In that paper, a non-linear discrete model of the ship was assumed. Process state variables were estimated using a linear Kalman filter. Dove et al. concept was developed further by Burns [1], who extended the guidance problem to a set of ships moving along a given voyage route. The proposed reactions in the collision situation were found with the theory of fuzzy sets. An autonomous ship guidance system was presented by Iijima and Hagiwara [5]. In order to evaluate the collision situation, make a decision, and give maneuvering orders, the authors developed the computer expert system. The system was tested on a training maneuvering ship in Tokyo Bay. During 1990–1994, a complex application prototype was worked out based on expert system technology applied for oceanic and coastal navigation. The integrated intelligent system consisted in a number of sub-systems which executed particular functions (e.g., optimum navigation, course planning, and automatic anchorage). The sub-systems were connected, via local area network, to the “Captain Expert” — an expert system based on the knowledge and experience of navigators with long practice. The system was developed for guiding ships in oceanic an coastal navigational conditions in a fully automated manner, without crew interference, only being in touch with land-based services. A collision avoiding system restricted waters was developed by Hayashi et al. [4]. That system made use of an electronic map and the radar operation for evaluating the actual position and giving an assessment of the overall navigational situation. Sudhendar and Grabowski [16] discussed possible directions of further development, formulated requirements for an intelligent pilot system, and presented the actual state of work intended to meet particular requirements of coastal services in the United States and Canada. The inner structure of the expert system was discussed on the basis of a piloting system for the St. Lawrence Seaway in Canada. A detailed analysis of models and the synthesis of algorithms for safe, optimum steering were described in [13]. In their works the problem of determining a safe trajectory as a non-linear programming task was formulated, where a kinematics model of the own ship was applied. Another possible approach to this problem is the reduction of the solution space to a finite-dimensional one by creating so-called digitized matrix of permissible maneuvers for a given collision situation and a certain time instant [13]. In [9, 10, 11] the problem of avoiding collisions was formulated as the multi-criteria optimization task. Three separate criteria were used. The attempt to estimate the safe trajectory using classifier systems was presented in [3]. The collision situation was modeled as a fuzzy process with many inputs; for selecting the steering rules the authors made use of a fuzzy classifier system.

In the overwhelming majority of the reviewed publications on automatic ship guidance, the navigational process in the areas of intensive traffic was supported by expert systems. In this work, we report on experiments with evolutionary system, which takes into consideration the motion of other ships; see also [12, 14, 15].

1.2 Organization of the paper

The paper is organized as follows. In Section 2, the definition of the ship navigation environment is provided. Some aspects of evolutionary algorithm used in our implementation are discussed in Section 3, while Section 4 presents an example of a modified *on-line* version of the evolutionary safe path search algorithm. Section 5 concludes the paper.

2 Environment

The ship sails in an environment with some natural constraints (e.g., lands, canals, shallow waters) as well as other constraints resulting from formal regulations (e.g., traffic restricted zones, fairways, etc). These constraints are assumed stationary and are defined by polygons — in a similar manner to that used in creating electronic maps. When sailing in a stationary environment, the own ship meets other sailing strange ships/targets (some of which constitute a collision threat).

The degree of the threat of collision with dangerous targets is not constant and depends on the approach parameters: D_{CPA} (Distance at Closest Point of Approach) and T_{CPA} (Time of Closest Point of Approach), as well as on the speed ratio of both ships, and the distance and bearing of the target.

It is assumed that the dangerous target is each target that has appeared in the area of observation and can cross the estimated course of the own ship at a dangerous distance. Actual values if this distance depend on the assumed time horizon. The ranges of 5–8 N. miles in front of the bow, and 2–4 N. miles behind the stern of the ship are assumed. In the evolutionary task, the targets threatening with a collision are interpreted as moving dangerous areas having shapes and speeds corresponding to the targets determined by the ARPA system. The moving constraints represent approaching ships, and the shape of each constraint depends on the safety conditions: on an assumed value of the safe approach distance (D_{safe}), assumed safe distance , speed ratio, and bearing of the moving target. A safe distance is selected by the operator depending on the weather conditions, sailing area, and speed of the ship. When planning the safe trajectory, the evolutionary algorithm should take into account both the fixed constraints, and the areas of danger representing the moving targets, which dynamically change their locations [12, 14]. Figures 1 and 2 display models of the environment where:

- fixed navigation constraints are modeled using convex and concave polygons,
- moving targets are modeled as moving hexagons,
- the dimensions of the own-ship are neglected due to small length of the own-ship with respect to the maximum length of the areas representing the moving targets.

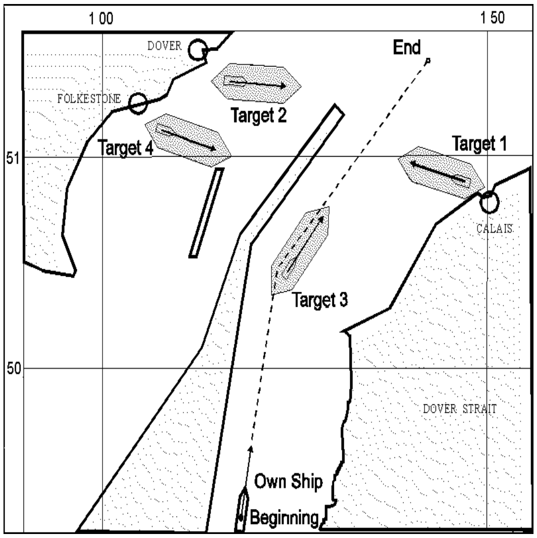

Fig. 1. Navigation situation in Dover Straits. There is an own-ship, four strange ships, and several navigational constraints

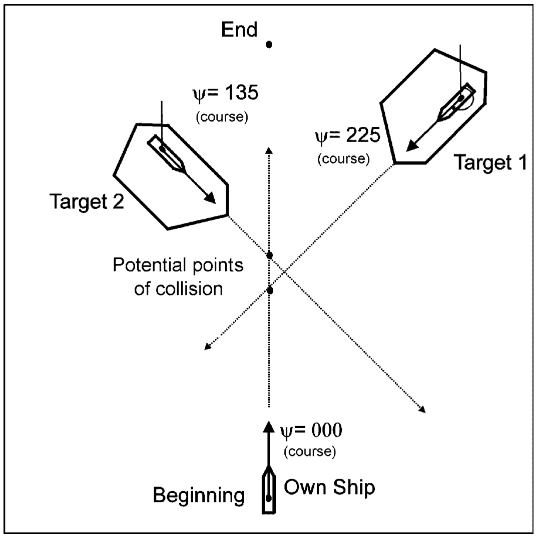

Fig. 2. Approaching two moving targets — hexagon constraint shapes

2.1 Planning the trajectory in a collision situation

According to transport plans, the own ship should cover a given route R_0 in some assumed time. On the other hand, it has to move safely down a given trajectory, i.e., it must avoid navigation obstacles and cannot come too close to other moving targets. Estimation of a ship's trajectory in a collision situation represents a difficult trade-off between a necessary deviation from a given course and the safety of sailing. Hence it is a multicriterion planning problem which takes into account the safety and economy of the ship motion.

The estimation of the own ship trajectory in the collision situation consists of determining a path, S, as the part of the given route R_0, from the present location (starting point) $(x_0, y_0) \in R_0$ to the actual end point $(x_e, y_e) \in R_0$. This path has the form of a sequence of elementary line segments s_i $(i = 1, ..., n)$, linked with each other in turning points (x_i, y_i). The choice of the actual starting and end point depends on an assumed sensible horizon and is made by the operator. The boundaries of the environment are defined as

$$E = \{(x, y) \in R^2 \ : \ a \leq x \leq b, \ c \leq y \leq d\}; \tag{1}$$

O_stat_j $(j = 1, ..., k)$ and $O\_dyn_j(t)$ $(j = k + 1, ..., l)$ represent the sets of static and dynamic constraints, respectively. Note that each dynamic constraint, $O\_dyn_j(t)$, is time-dependent; i.e., it defines different subareas of E for different values of t. Clearly, static constraints represent time-independent constraints (e.g., lands, canals, restricted zones, etc), whereas dynamic constraints represent strange-ships.

The space $SF(t)$ of safe (anti-collision) paths is defined as

$$SF(t) = E - \bigcup_{j=1}^{k} O\_stat_j - \bigcup_{j=k+1}^{l} O\_dyn_j(t). \tag{2}$$

In other words, a path S is safe (i.e., it belongs to the set of safe paths $SF(t)$) if any segment s_i $(i = 1, \ldots, n)$ of S stays within the limits of environment E, does not cross static constraints O_stat_j, and at the time instances t determined by the current locations of the own ship, does not come in contact with moving areas $O\_dyn_j(t)$ representing targets. Paths which cross the restricted areas generated by static and dynamic constrains are called unsafe, or dangerous paths.

The task of estimating the own-ship trajectory in a collision situation (so-called the *steering goal*) is performed as an evolutionary search for safe paths in the permissible space E, with subsequent selection of a near-optimum path S^* from the set SF with respect to the fitness function (defined by the path cost).

3 Evolutionary algorithm ϑEP/N++

A crucial step in the development of an evolutionary trajectory planning system was made by the introduction of dynamic parameters: the time, and the moving constraints. In the evolutionary algorithm for trajectory planning eight genetic operators were used, which were: soft mutation, mutation, adding a gene, swapping gene locations, crossing, smoothing, deleting a gene, and individual repair [14]. The level of adaptation of the trajectory to the environment determines the total cost of trajectory. The trajectory costs include both the safety cost $Safe\_Cost(S)$ and that connected with the economy $Econ\_Cost(S)$ of the ship motion along the trajectory of concern. The total cost of the trajectory is defined as:

$$Total\_Cost(S) = Safe\_Cond(S) + Econ\_Cond(S) \tag{3}$$

The safety conditions are met when the trajectory does not cross fixed navigational constraints, nor moving areas of danger. The actual value of the safety cost function $Safe\_Cost(S)$ is evaluated as the maximum value defining the quality of the turning points s_i with respect to their distance from the constraints:

$$Safe\_Cond(S) = w_c \cdot clear(S), \tag{4}$$

where: $clear(S)$=c_i, w_c is weight coefficient, c_i is the difference in length between the distance to the constraint-closest turning point s_i and the safe distance d. The trajectory cost connected with economic conditions $Econ\_Cost(S)$ includes: total length of trajectory S consisting of n line sections s_i, function of maximum turning angle between particular trajectory sections at turning points s_i time needed for covering the trajectory S. The total cost of the trajectory adaptation to the environment, resulting from the economic conditions, is equal to:

$$Econ\_Cond(S) = w_d \cdot dist(S) + w_s \cdot smooth(S) + w_t \cdot time(S), \tag{5}$$

where: w_d, w_s, w_t are weight coefficients.

In many cases, the most effective maneuver for the own ship seems to be where the course changes *and* the speed is reduced. The speed reduction of the own ship can make it possible to pass a target without significant changes in its course. The analysis of examples where the speed of the own ship was initially assumed constant clearly shows the need for making this parameter variable along particular trajectory sections. In practice the speed is modified using an additional genetic operator: the speed mutation [13, 15]. A set of permissible speed values was defined as ϑ=3.6, 8.6, and 13.6 knots; the mutation operator can select from this set an appropriate speed for any trajectory section under consideration. Those speeds correspond to the following telegraph settings: slow ahead, half ahead, and full ahead. Additionally, the total time of trajectory passing, $time(S)$, was added to the function of the trajectory fitness, which took into consideration changes in the own ship's speed.

The next aspect of the evolutionary algorithm ϑEP/N++ is the *on-line* work, in which changes in parameters of motion of particular targets are taken into account. An interesting experiment was to check how the algorithm would determine the passing trajectory when one of the targets reduced its speed or its course.

The operation of the evolutionary trajectory planning algorithm system has been examined for a number of collision situations. Tests of the modification version of algorithm which changes the own ship's speed along the trajectory sections were discussed in previous works [13, 14, 15]. This article is focused on the discussion of experiments of *on-line* algorithm version. In all experiments reported here, the population size is 40, i.e., the evolutionary system processes 40 paths.

4 Simulation studies

In our initial experiments the own-ship moves with a speed ϑ (along the a safe path S) from the starting point (x_0, y_0) to the end point (x_e, y_e), and at the initial instant t_0, the motion of the strange-ships is defined as uniform. For each target, its motion is represented by the following parameters: bearing, distance, speed, and course, estimated by the ARPA system. The path of the own ship has the form of a sequence of elementary line segments s_i $(i = 1, ..., n)$, linked with each other in turning points (x_i, y_i).

These initial experiments considered the speed of the own ship constant; however, it is possible to gain additional efficiency while varying the speed. We return to this issue later in the paper.

It is relatively easy to initialize the population of paths: each path (individual) can be generated randomly. Next, each path is evaluated. To determine whether a path is safe, the path is examined with respect to the set of static and dynamic constraints. The instantaneous locations of the dynamic areas with respect to the evaluated path depend on time t_c, determined by the first crossing point (x_c, y_c) between the own ship's path S and the trajectory of the target. For example, in the Figure 3, these crossing points are the points of the biggest collision threat for paths 1, 2, and 3. Having known the length of the line segment from the starting point (x_0, y_0) to the crossing point (x_c, y_c) and assuming that the own ship will keep moving with the uniform speed ϑ, it is possible to determine time t_c which the own ship needs in order to cover this distance.

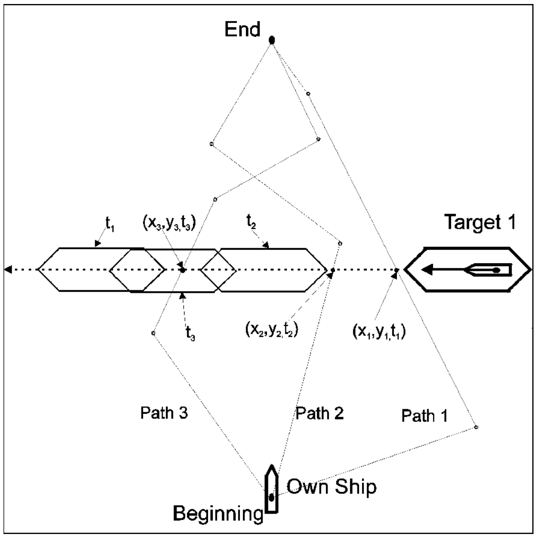

Fig. 3. Crossing paths and the corresponding dangerous areas

After time t_c, the instantaneous location of the target with respect to the own ship is modeled as a dangerous area of hexagonal shape. Referring again to Figure 3, three locations of the target (at times t_1, t_2, and t_3) are given for three paths; note that the path segment of path 1 between the own ship and the intersection with the trajectory of the target is the longest one (i.e., longer than similar path segments of paths 2 and 3); consequently, t_1 is larger than t_2 and t_3, and the hexagonal shape of the target for t_1 is the leftmost one. Of course, as explained earlier, the detailed shape and dimensions of the hexagon depend on the safety conditions given by the operator.

After the paths are evaluated, selected paths are modified by specialized set of operators (for details, see [18]).

The values assumed in the paper are the following:

- the distance in front of the bow which guarantees avoidance of the collision is equal to $3 \cdot D_{safe}$ (in practice, safe distance D_{safe} is taken from the range between 0.5 and 3.0 nautical miles)
- the distance behind the stern is equal to D_{safe}
- the width of the dangerous area on each side of the own ship is chosen with the preference of the ship's passage behind the stern of the target, which depends on the course and bearing of the target.

4.1 Simple example

Before we present the results of the evolutionary system on several test cases, we provide two simple examples, where the set of static constraints is empty and the dynamic constraints are defined by one or two strange-ships, respectively.

The first example (Figure 4) shows the situation when the own ship approaches a single target on its right side. As usual, time horizons for collision avoidance are around 30 minutes, we assumed $x = y = 8$ nautical miles. The population consisted of 10 individuals (paths), and the system converged after 300 generations . It is clear that the own ship, steering along the developed trajectory, will pass the target safely, passing it behind the stern. It is interesting to note that initially (see $Generation = 50$) the own ship tried to move "left" (somewhat along the target), but clearly, much better maneuver is to go slightly right (as it is the case for $Generation = 300$).

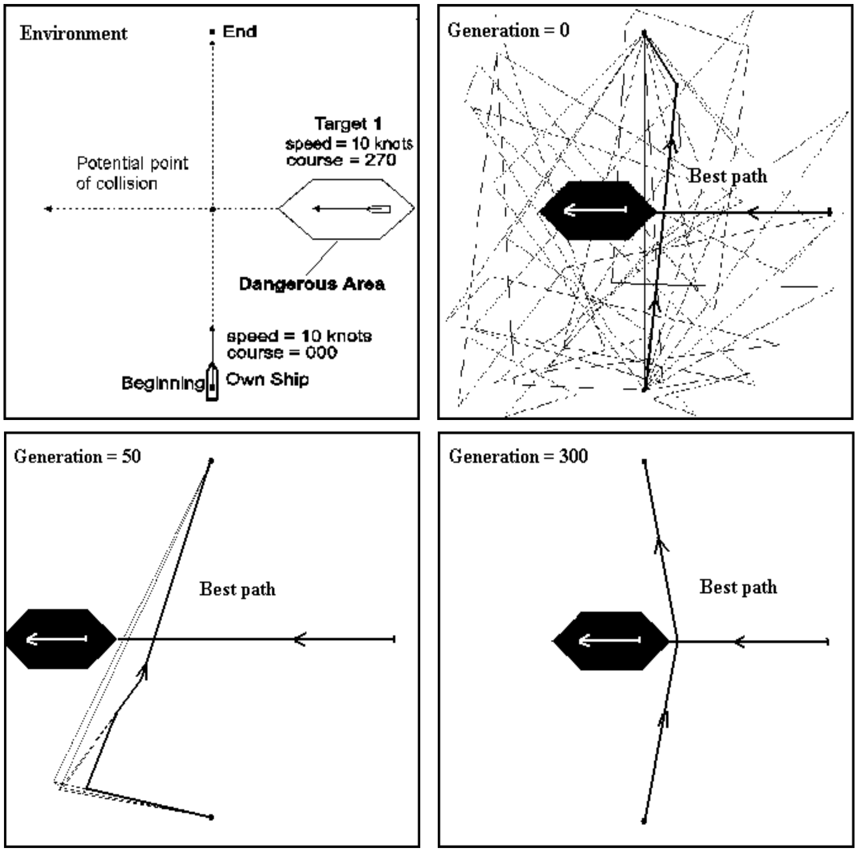

Fig. 4. Evolution of paths for the case of approaching one moving target

Note that in all three motion diagrams of Figures 4 (as well as in all further figures) the locations of the dynamic areas are shown (black hexagons) with respect to the best path, these locations depend on time determined by the first crossing point between the own ship's path and the trajectory of the target).

4.2 *On-line* path planing

In order to test the operational correctness of the *on-line* version of algorithm ϑEP/N++, certain trajectories were calculated using the *off-line* and *on-line* versions and then compared with each other. The test was divided into three phases. During the first phase, paths obtained in the *off-line* mode were tested. Then, during the next phase, the real motion of the own ship was studied; the ship was traveling along the trajectory assumed in the *off-line* mode. For the *on-line* version (phase number three), a quality assessment for the calculated trajectory was made on the basis of *on-line* calculations performed after changing the parameters of motion of one or more dynamic constraints — targets. The comparison was made for two sample environments with a relatively high level of complexity.

The first case presents the navigational situation in which the own ship passes around three islands and four moving targets from different directions and at different speeds. Input parameters for the simulation are shown in Figure 5. The speed of the own ship was defined as equal to 3.6, 8.6, or 13.6 knots. The progress of trajectory adaptations, made in the *off-line* mode for the case of the own ship meeting four moving objects in a collision situation, is shown in Figure 5, after 200, 500, and 1,000 generations, respectively.

During the computational process, after 1,000 generations (computing time 12 seconds on a standard PC) no trajectory changes in the population were observed. Moving along the determined trajectory with changing speed, the own ship can pass the targets in front of their bows or behind their sterns, sailing between the islands. The ship speed is changed along subsequent trajectory sections. Initially, the ship reduces the speed to pass first two targets and then, after sailing between the islands it sails faster as it is not restricted by excessive approach to Target 3 and Target 4, at the same time making it possible to reach the final destination point in the shortest time. The execution of the proposed trajectory gives the optimum cost of trajectory passing with respect to the safety and economic criteria.

The next phase examines the real motion of the own ship traveling along the assumed (in *off-line* mode) trajectory and passing targets (Figure 6). During this phase it is possible to assess more accurately the correctness of positions of the passed targets with respect to the own ship. Figure 6 presents the navigational situation in the real motion at 10, 30, 40, and 50 minutes. In contrast to the relative presentation (Figure 5), the positions of the targets here (shaded ares representing dynamic constraints) and the position of the own ship on the trajectory are determined at times 10, 30, 40, and 50 minutes, which elapsed after the simulation has started. Additionally, the system simulates possible changes

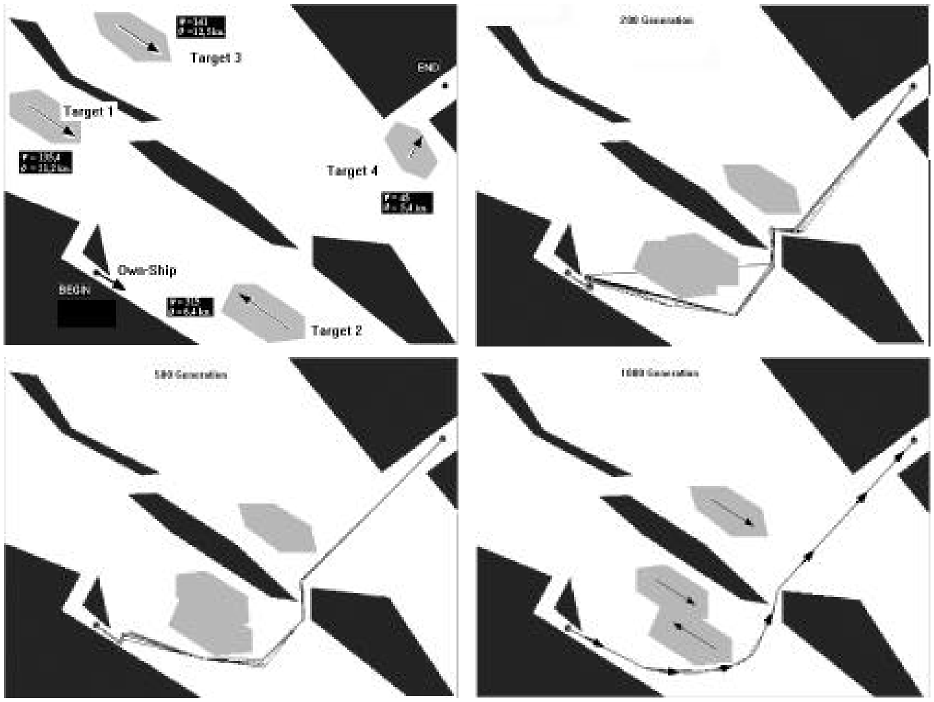

Fig. 5. Trajectory evolution in the *off-line* mode after 200, 500 and 1000 generations, respectively, for the case of approaching four moving targets in the presence of static navigation constraints (population 40 paths). The speed of own ship varies.

of parameters of motion (speed or/and course) of the targets. Changing parameters of the targets creates a new navigational situation — a new environment which triggers the adaptation of the own ship trajectory calculated in the *on-line* mode. Switching to the *on-line* mode, the system vEP/N++ adapts trajectory to the new environment.

For the test environment defined in Figure 5, the algorithm switched to the *on-line* mode at time of 20 minutes. Then selected environment parameters were changed, namely the course of the target seen in the right-hand part of the screen was changed from 45 degrees to 210 degrees, and the speed of motion of the target seen in the upper left part of the screen was changed from 5.2 to 17.6 knots. Figure 7 shows a newly calculated safe trajectory (after 1,000 generation) which has taken into account the above changes. After comparing Figures 5 7, it is clear that the path calculated in the *on-line* mode is similar to the trajectory obtained in the off-line mode, with certain differences resulting from trajectory corrections made due to environment changes introduced. This demonstrates the operational correctness of the *on-line* version.

The second case (Figure 8, left) presents a situation when the own ship leaves the channel and meets three targets moving from different directions and at

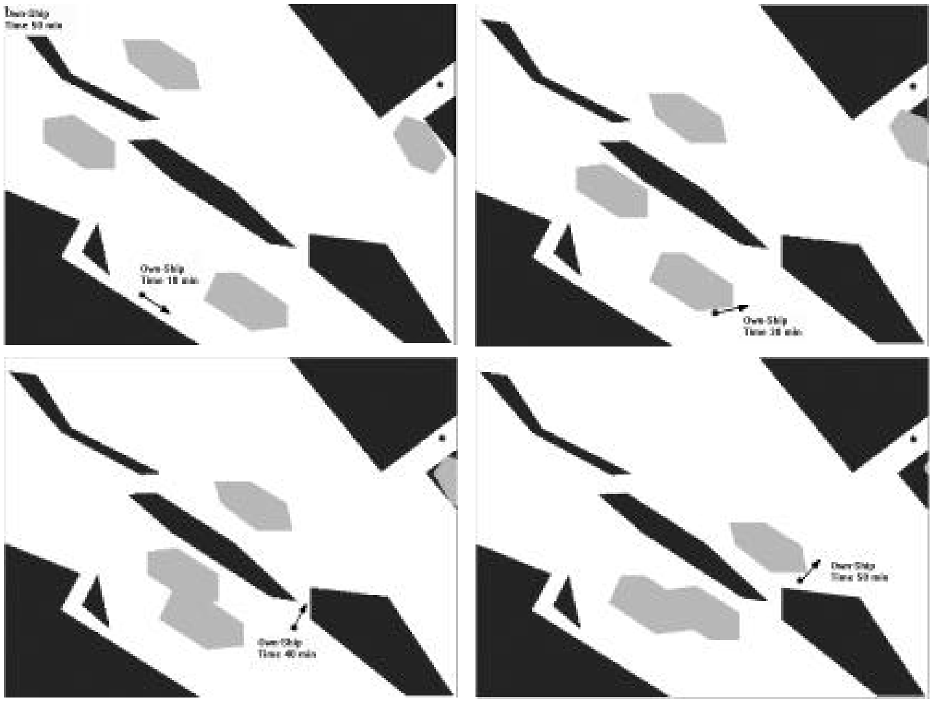

Fig. 6. Navigational situation in real motion at 10, 30, 40, and 50 minutes after start

different speeds. Input parameters for the simulation made for this environment are shown in Figure 8 (left). The speed of the own ship is equal to $\vartheta = 18.7$ knots. The adaptation of the own ship trajectory calculated in the *off-line* mode (first phase) for collision meeting of three moving objects is shown in Figure 8 (right), after 1,000 generations.

Figure 9 refers to the second phase of the algorithm operation. In this phase the real motion of the own ship traveling along the assumed (in *off-line* mode) trajectory and passing strange targets is presented. For the tested environment and navigational situation, the positions of own ship and targets are displayed at times of 10 and 20 minutes after the start.

In order to examine the operation of the *on-line* mode in the test environment, at 20 minutes, selected parameters were changed in the motion of the target seen in the lower left-hand part of the screen. The speed was changed from 12.7 knots to 25 knots, and its course from 45 degrees to 103 degrees. The time of switching the system to the *on-line* mode is shown in Figure 10 (left). The newly calculated safe trajectory taking into account changes introduced to the motion of the target is shown in Figure 10 (right).

For the test environment, the comparison of the solutions shown in Figure 8 (right) and Figure 10 (right) leads to the conclusion that the path calculated in the *on-line* mode is safe and close to optimum. This demonstrates the correctness

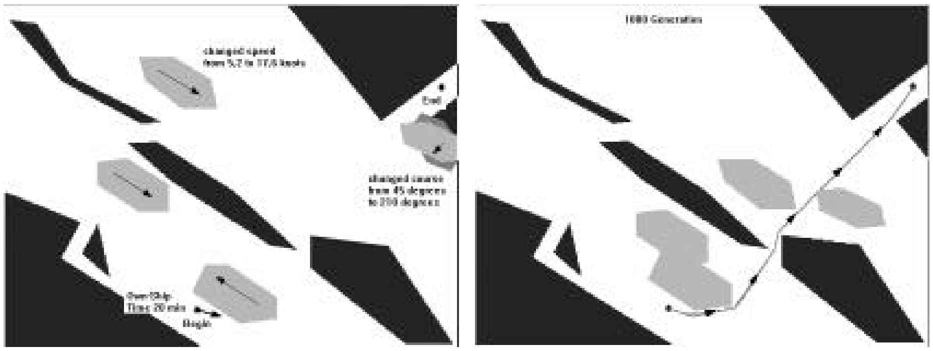

Fig. 7. Changing parameters of motion of two targets at time of 20 minutes (left) and newly calculated safe trajectory after 1,000 generations (right)

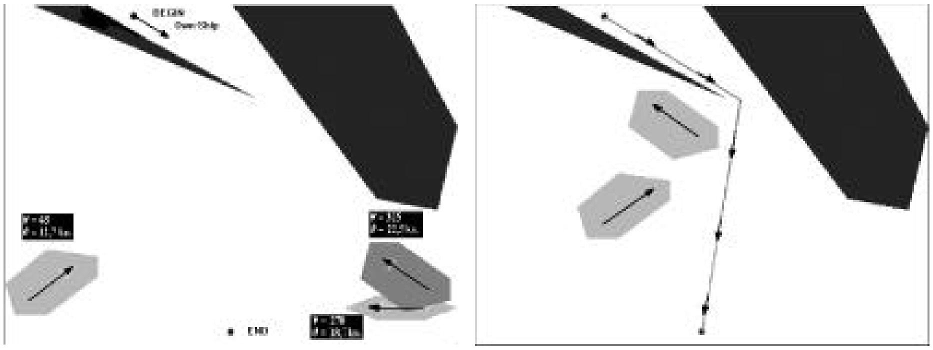

Fig. 8. Situation of meeting with the own ship and three targets (left). Trajectory evolution in the *off-line* mode after 1,000 generations (right)

of operation of the *on-line* program version. It should be stressed here that two most complex environments were selected for the present tests, out of all environments earlier studied by the authors (see [13, 14, 15]).

5 Conclusions

The evolutionary method of estimating the safe and optimum passing path being the own ship's trajectory in the environment with static and dynamic constraints is a new approach to the problem of avoiding collisions at sea. A number of preliminary tests presented in the paper make it possible to formulate the following conclusions:

- evolutionary algorithms can be effectively used for solving problem of avoiding collisions at sea when the environment is modeled as a set of polygons representing navigation constraints and moving targets,

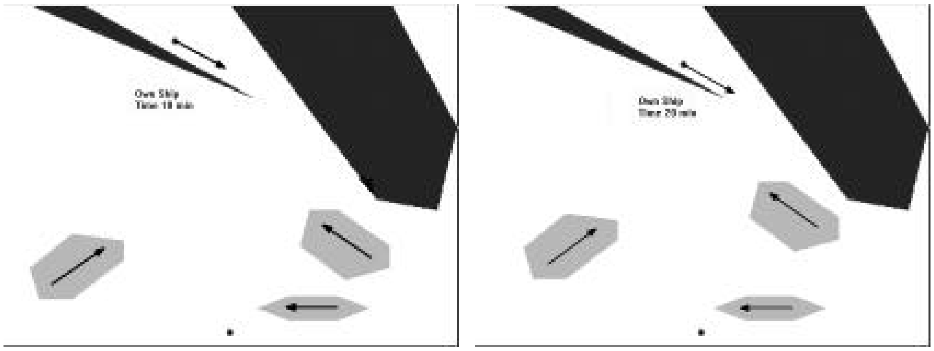

Fig. 9. Navigational situation in the real motion at times of 20 and 30 minutes after start

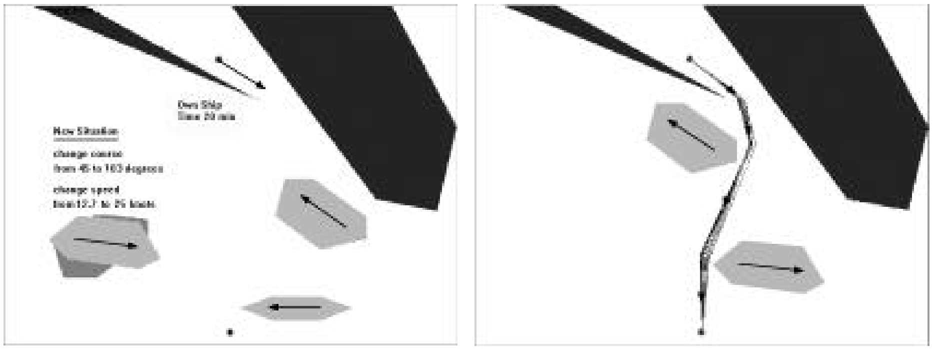

Fig. 10. Changing parameters of motion of one target at time of 20 minutes (left) and newly calculated safe trajectory after 1,000 generations (right)

- the task of evolutionary estimation of the own ship trajectory in a collision situation is reduced to an adaptive search for a set of safe paths S in a permissible space X , with subsequent selection of the optimum trajectory with respect to the fitness function,
- the strange ship - target is modeled in the evolutionary environment as a dynamic constraint, a moving area of danger having a hexagonal shape. The detailed shape and dimensions of the hexagon depend on safety conditions and parameters of motion entered by the operator.

The introduction of additional elements (*on-line mode*) to the present program in order to include other environment changes does not impose any significant problems for the evolutionary path planning, and undoubtedly make the process more similar to real navigation situations. Each newly occurred situation can be in a natural way added to the operational diagram of the evolutionary algorithm.

Acknowledgements

The research reported in this paper was supported by the grant 3T11A 003 26 from the Polish State Committee for Scientific Research.

References

1. Burns R. S., An Intelligent Integrated Ship Guidance System. 2nd IFAC Workshop *Control Applications in Marine Systems*, pp. 199-208, Genova, Italy 1992.
2. Dove M. J., Burns R. S., Stockel C.T., An Automatic Collision Avoidance and Guidance System for Marine Vehicles in Confined Waters. *Journal of Navigation*, Vol. 39, p. 180,1986.
3. Furuhashi T., Nakaoka K., and Uchikawa Y., A Study on Classifier System for Finding Control Knowledge of Multi-Input Systems, (F. Herrera and J.L. Verdegay, Editors), *Genetic Algorithms and Soft Computing*, Physica-Verlang, 1996.
4. Hayashi, S., Kuwajima, S. Sotooka, K., Yamakazi H. and Murase H . A stranding avoidance system using radar image matching: development and experiment. *Journal of Navigation*, Vol. 44, p. 205, 1991.
5. Iijima Y. and Hayashi S. Study towards a twenty-first century intelligent ship.*Journal of Navigation*, Vol. 44, p. 184, 1991.
6. Latombe, J.C., *Robot Motion Planning*. Kluwer Academic Publishers, 1991.
7. Lin, H.-S., Xiao, J., and Michalewicz, Z., Evolutionary Algorithm for Path Planning in Mobile Robot Robot Environment, In *Proc. First IEEE Conference on Evolutionary Computation*, Orlando, Florida, June 1994, pp. 211-216.
8. Michalewicz Z., *Genetic Algorithms + Data Structures = Evolution Programs*. Spriger-Verlang, 3rd edition, 1996.
9. Smierzchalski R., The Application of the Dynamic Interactive Decision Analysis System to the Problem of Avoiding Collisions at the Sea, *in Polish* 1st Conference Awioniki, Jawor, Poland 1995.
10. Smierzchalski R., The Decision Support System to Design the Safe Maneuver Avoiding Collision at Sea. ISAS'96, Orlando, USA, 1996.
11. Smierzchalski R., Multi-Criterion Modeling the Collision Situation at Sea for Application in Decision Support, In *Proceedings of MMAR'96*, Międzyzdroje, Poland, 1996.
12. Smierzchalski R., Trajectory planning for ship in collision situations at sea by evolutionary computation, In *Proceedings of the IFAC MCMC'97*, Brijuni, Croatia, 1997.
13. Smierzchalski R., Analysis and Synthesis of Navigator Decision Support Algorithms in Collision Situation at Sea, (in Polish) Academic Press, Gdynia Maritime University, Gdynia 1999.
14. Smierzchalski R. and Michalewicz, Z., Adaptive Modeling of a Ship Trajectory in Collision Situations at Sea, In *Proccedings of the 2nd IEEE World Congress on Computational Intelligence*, ICEC'98, Alaska, USA 1998, pp. 364–369.
15. Smierzchalski R. and Michalewicz, Z., Modeling of a Ship Trajectory in Collision Situations at Sea by Evolutionary Algorithm, *IEEE Transaction on Evolutionary Computation* Vol.4, No.3, pp.227–241, 2000.
16. Sudhendar H., Grabowski M., Evolution of Intelligent Shipboard Piloting Systems: A Distributed System for the St Lawrence Seaway. *Journal of Navigation*, Vol. 49, No.3, 1996.

17. Xiao, J. and Michalewicz, Z., *An Evolutionary Computation Approach to Planning and Navigation*, chapter in *Soft-Computing and Mechatronics*, K. Hirota and T. Fukuda (Editors), *Physica-Verlag*, 1999.
18. Xiao, J., Michalewicz, Z., Zhang, L., and Trojanowski, K., Adaptive Evolutionary Planner/Navigator for Mobile Robots, IEEE Transactions on Evolutionary Computation, Vol.1, No.1, 1997, pp.18–28.
19. Yap, C.-K., Algorithmic Motion Planning, In *Advances in Robotics, Vol.1: Algorithmic and Geometric Aspects of Robotics*, J.T. Schwartz and C.-K. Yap Ed., pp.95–143, Lawrence Erlbaum Associates, 1987.

This article was processed using the LaTeX macro package with LLNCS style