TRANSNAV International Journal on Marine Navigation and Safety of Sea Transportation
Volume 5 Number 4 December 2011

Experimental Research on Evolutionary Path Planning Algorithm with Fitness Function Scaling for Collision Scenarios

P. Kolendo, R. Smierzchalski & B. Jaworski
Gdansk University of Technology, Gdansk, Poland

ABSTRACT: This article presents typical ship collision scenarios, simulated using the evolutionary path planning system and analyses the impact of the fitness function scaling on the quality of the solution. The function scaling decreases the selective pressure, which facilitates leaving the local optimum in the calculation process and further exploration of the solution space. The performed investigations have proved that the use of scaling in the evolutionary path planning method makes it possible to preserve the diversity of solutions by a larger number of generations in the exploration phase, what could result in finding better solution at the end. The problem of avoiding collisions well fitted the algorithm in question, as it easily incorporates dynamic objects (moving ships) into its simulations, however the use scaling with this particular problem has proven to be redundant.

1 INTRODUCTION

The problem of path planning occurs in numerous technical applications, such as, motion planning for mobile robots [12], ship weather routing in ocean sailing, or safety path planning for a ship in a collision situation at sea [5]. The problem is defined in the following way: having given a moving object and the description of the environment, plan the path for object motion between the beginning and end location which avoids all constraints and satisfies certain optimization criteria. The problem can be divided into two basic tasks: the off-line task, in which we look for the path of the object in the unchanging environment, and the on-line task, in which the object moves in the environment that meets the variability and uncertainty restrictions. The on-line mode of the path planning relates to the control of the moving object in the non-stationary environment, in which parts of some obstacles reveal certain dynamics.

The main goal of the present paper is to present collision scenarios simulation results acquired using evolutionary path planner and to analyse how the fitness function scaling impacts the solution [1,2,14]. Particular instance of the path planning problem as the navigation problem of avoiding collision at sea [5, 6] is considered. By taking into account certain boundaries of the manoeuvring region, along with the navigation obstacles and other moving ships, we reduce the problem to the dynamic optimization task with static and dynamic constrains. We consider this an adaptive evolutionary task of estimating the ship path in the unsteady environment. The research was performed using Evolutionary Planner/Navigator (υEP/N++) system [7, 8, 9] which takes into account specific nature of the process of avoiding collisions, by using different types of static and moving constraints to model the real environment of moving targets and their dynamic characteristics.

2 EVOLUTIONARY ALGORITHMS

Evolutionary Algorithms (EA) are optimization methods that try to mimic evolutionary path in order to find the best solution for a specific problem. Each member of a generation - a set of potential solutions - is being rated against a fitness function to determine member's individual adaptation rate - the quality of the solution. Best fits are being then selected to prepare a new generation. Also, additionally, there is a small chance of offspring's mutation that helps to keep the population differentiated. This process is repeated until an optimal solution is found or the maximum, presumed number of generations is reached. To find out more about EA please refer to position [13] of the bibliography.

3 PLANNER INTRODUCTION

When determining the safe trajectory for the so-called *own ship*, we look for a trajectory that compromises 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, here referred to as *strange ships* (or *targets*). In this paper the following terminology is used: the term *own ship* means the ship, for which the trajectory is to be generated, and *strange ship* or *target* mean other ships in the environment, i.e. the objects which are to be avoided. All trajectories which meet the safety conditions reducing the risk of collision to a satisfactory level constitute a set of feasible trajectories. The safety conditions are, as a rule, defined by the operator based on the speed ratio between the ships involved in the passing manoeuvre, the actual visibility, weather conditions, navigation area, manoeuvrability of the ship, etc. The most straightforward way of determining the safe trajectory seems to be the use of an additional automatic device – a decision supporting system being an extension of the conventional Automatic Radar Plotting Aids (ARPA) system.

Other constraints resulting from formal regulations (e.g. traffic restricted zones, fairways, etc) are assumed stationary and are defined by polygons – in a similar manner to that used in creating the electronic maps. When sailing in the stationary environment, the own ship meets other sailing strange ships/targets (some of which constitute a collision threat).

It is assumed that the dangerous target [6] 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. The actual values of this distance depend on the assumed time horizon. Usually, the distances of 5-8 nautical miles in front of the bow, and 2-4 nautical miles behind the stern of the ship are assumed as the limits for safe passing. In the evolutionary task, the targets threatening with a collision are interpreted as the moving dangerous areas having shapes and speeds corresponding to the targets determined by the ARPA system.

The path S is safe (i.e., it belongs to the set of safe paths) if any line segment of S stays within the limits of the environment E, does not cross any static constraint and at the times t determined by the current locations of the own ship does not come in contact with the moving representing the targets. The paths which cross the restricted areas generated by the static and dynamic constrains are considered unsafe, or dangerous paths.

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

The υEP/N++ is a system based on evolutionary algorithm [1] incorporating part of the problem maritime path planning specific knowledge into its structures. The evolutionary approach provides many benefit such as real or close to real time operations, complex search and high level of adjustment possibilities. Due to the unique design of the chromosome structure and genetic operators the υEP/N++ does not need a discretised map for search, which is usually required by other planners. Instead, the υEP/N++ "searches" the original and continuous environment by generating paths with the aid of various evolutionary operators. The objects in the environment can be defined as collections of straight-line "walls". This representation refers both to the known objects as well as to partial information of the unknown objects obtained from sensing. As a result, there is little difference for the υEP/N++ between the off-line planning and the on-line navigation. In fact, the υEP/N++ realises the off-line planning and the on-line navigation using the same evolutionary algorithm and chromosome structure.

A crucial step in the development of the evolutionary trajectory planning systems was made by introducing the dynamic parameters: time and moving constraints. In the evolutionary algorithm used 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 [8, 9]. The level of adaptation of the trajectory to the environment determines the total cost of the trajectory, which includes both the safety cost and that connected with the economy of the ship motion along the trajectory of concern.

The current version of planner is also updated with the possibility of using fitness function scaling, which can essentially improve the quality of the results. Scaling is used to increase or suppress the diversity of the population, by controlling the selection pressure. It helps to maintain diversity of individuals at the initial phase of the computation, or to find a final solution at the end of it. The process of computation can be improved in the two abovementioned ways by using different scaling functions at proper times and changing the scaling parameters.

The scaling schemes which can be applied in the EA are: linear scaling, power law scaling, sigma truncation scaling, transform ranking scaling, ranked and exponential scaling [1, 2, 3, 10, 11]

4 SIMULATION ENVIROMENT

The operations of the evolutionary path planning algorithm υEP/N++ [8] was examined for multiple situations, one of them depicted as an example in Fig. 1. The simulation is usually performed in the environment populated with static and dynamic constraints, however tests presented in this paper consider only dynamic objects. The static constraints are represented by black polygons, while the dynamic objects (characterised by their own course and speed) were marked by grey hexagons. The figures below show only the dynamic objects (targets), that reveal a potential point of collision [10] with the own ship. The positions of the dynamic objects are displayed for the best route, which is bolded in the figures. In all here reported experiments the population size was 30, i.e. the evolutionary system processed 30 paths.

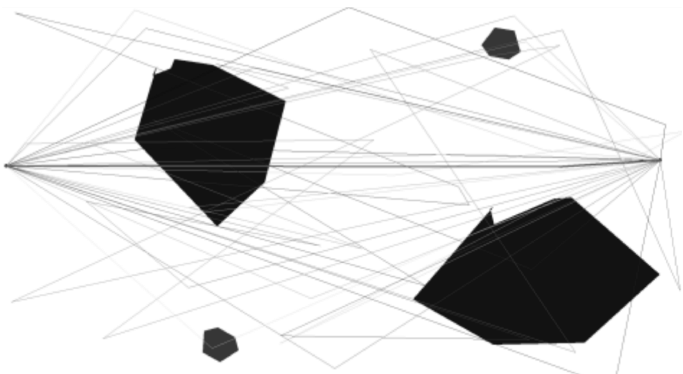

Figure 1. Example Simulation with the initial population.

As previous research show [16], scaling of the fitness function should allow later convergence of the results, thus presenting better solutions. Our tests were to check if this is the case also with the most common collision scenarios. Experiments were performed multiple times and the results represent the mean of what the simulations have shown. Due to the nature of EA, each program run is unique and it happened several times that a run without scaling produced results similar to those of the mean of the scaled ones (and the other way round), although those were marginal and depended mostly on the disruption of paths in the initial population. However it is worth noting that the planner is always able to find a good, feasible solution.

The runs performed for scaled cases were using power scaling with a power factor of 2. The program was set to the maximum of 400 generations, however during the tests generations were observed step by step (each generation was observed separately) in order to notice if scaling is working as expected and if the moment of final convergence is picked up precisely. The moment of achieving the final solution is one of the most crucial differences between a scaled and non-scaled EA. To show this clearly, each simulation was performed both for a scaled and non-scaled runs which allowed to form proper remarks.

5 SIMULATIONS

The simulations were performed for three most common collision situations and for two more complex. In each of them the own ship is represented by its own trajectory (as its size is negligible compared to the environment and target's safety zone). Both own ship and target have their starting positions equally far from the Point of the Potential Collision (*PPC*) and move at the same speed of 10 knots. The three scenarios differ from each other by the targets' trajectories. It is important to underline that our planner plots a path only for the own ship and the strange ship course is fixed and cannot be changed. υEP/N++ task is thereby to plot a new path that would avoid collision and reduce the costs of the course change to minimum while keeping the whole procedure save. The course of own ship is always 0^0, while the targets' course is 240^0 in situation 1 (Figure 2a), 300^0 in situation 2 (Figure 2b) and 120^0 in the last scenario (Figure 2c). Also two additional, advanced simulations were performed, which were constructed based on material in [15]. The example results, representing the mean of the observed runs for both scaled and non-scaled attempts are shown on Figures 4 to 8. Shortcut Gen. refers to number of the generation of the run shown in a segment. On those figures we can see the process of path plotting from the earliest generations (which quality is far from the ultimate one) through the final ones, observing how the υEP/N++ tries to calculate an optimal route. The figures show that once an good feasible path is found, the algorithm utilises genetic operators in order to shorten and smoothen the plotted course. The desired course is also often tangent to the target's safe area, as this correlates with the goals listed.

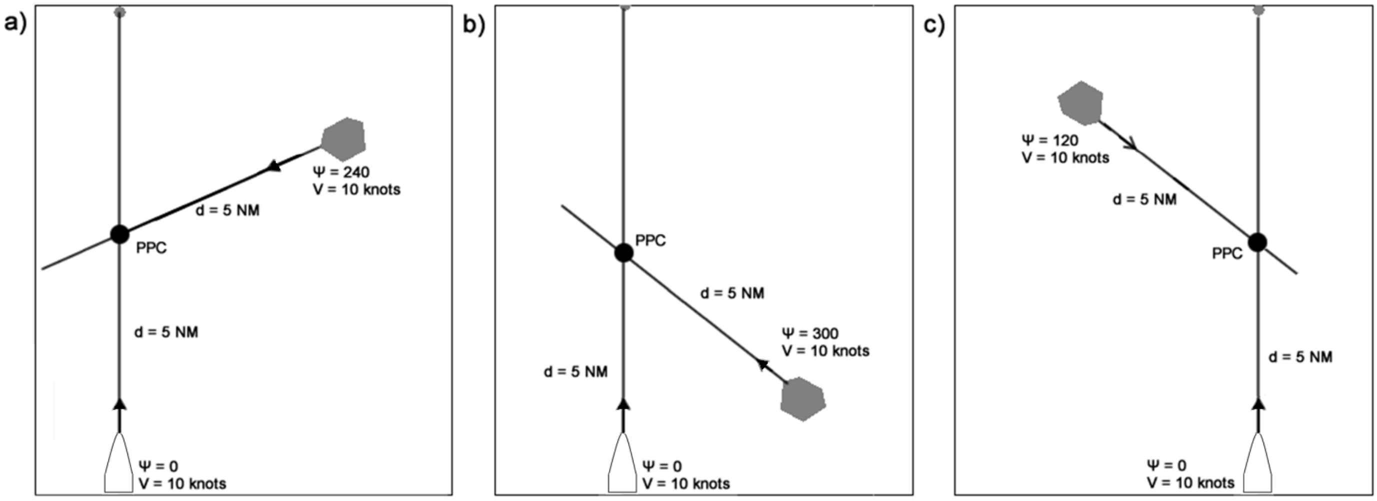

Figure 2. Basic Simulation Situations – target ship on a) 240^0 b) 300^0 c) 120^0 course

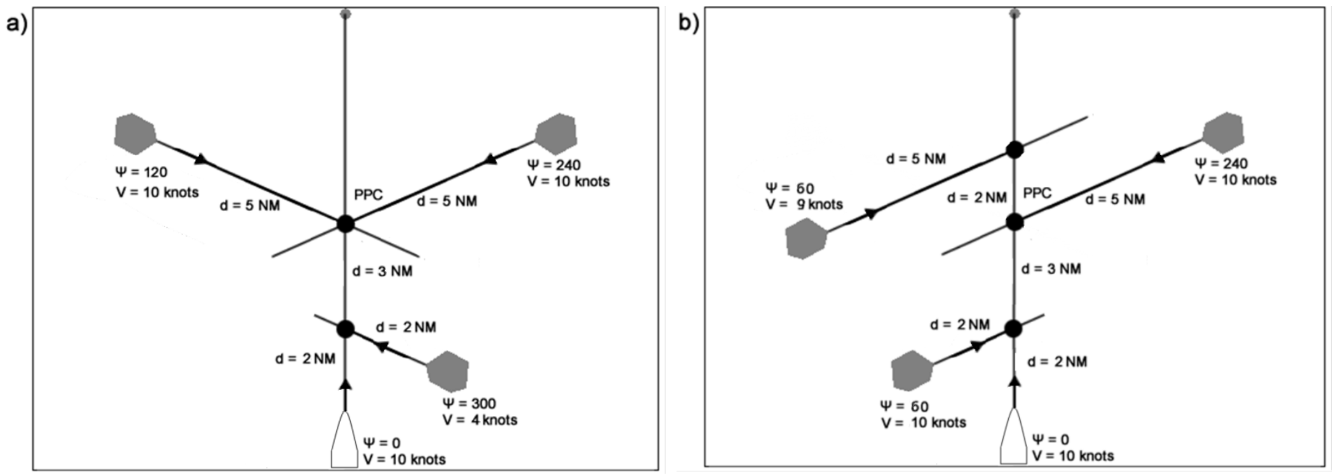

Figure 3. Advanced Simulation Situations.

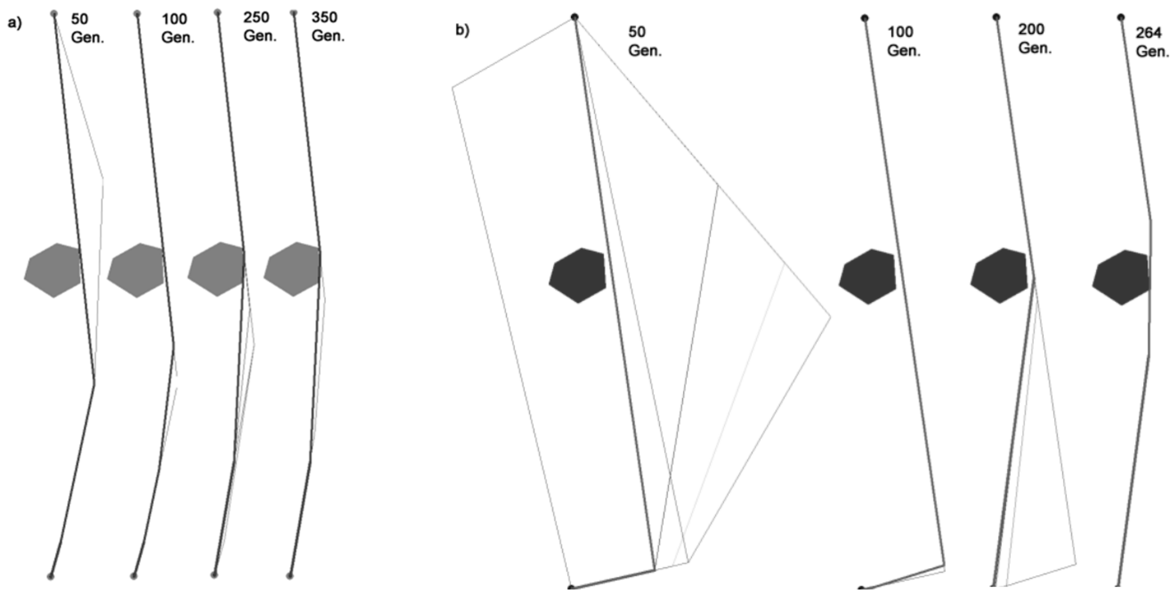

Figure 4. Simulation for Situation 1 a) with scaling b) without scaling

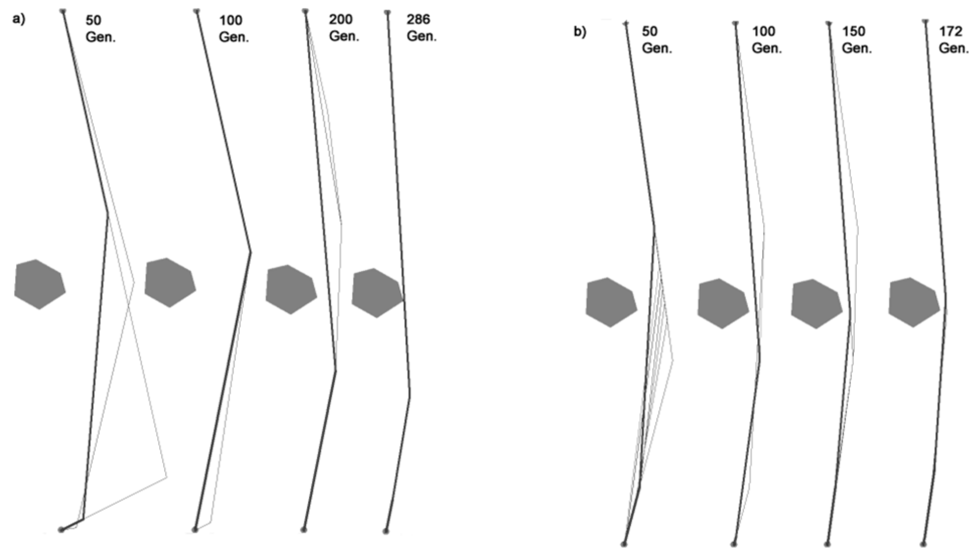

Figure 5. Simulation for Situation 2 a) with scaling b) without scaling

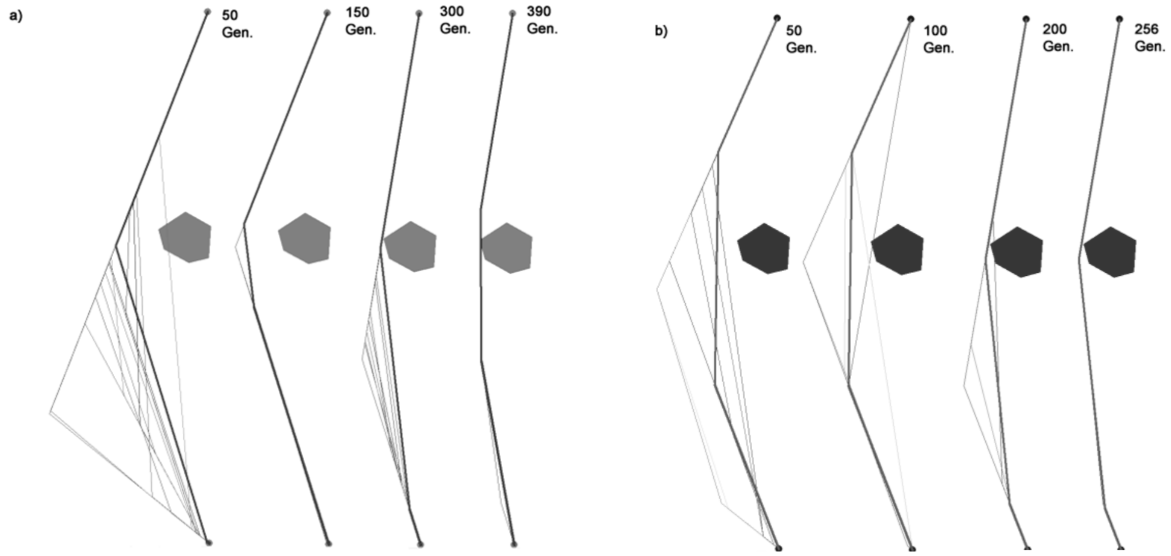

Figure 6. Simulation for Situation 3 a) with scaling b) without scaling

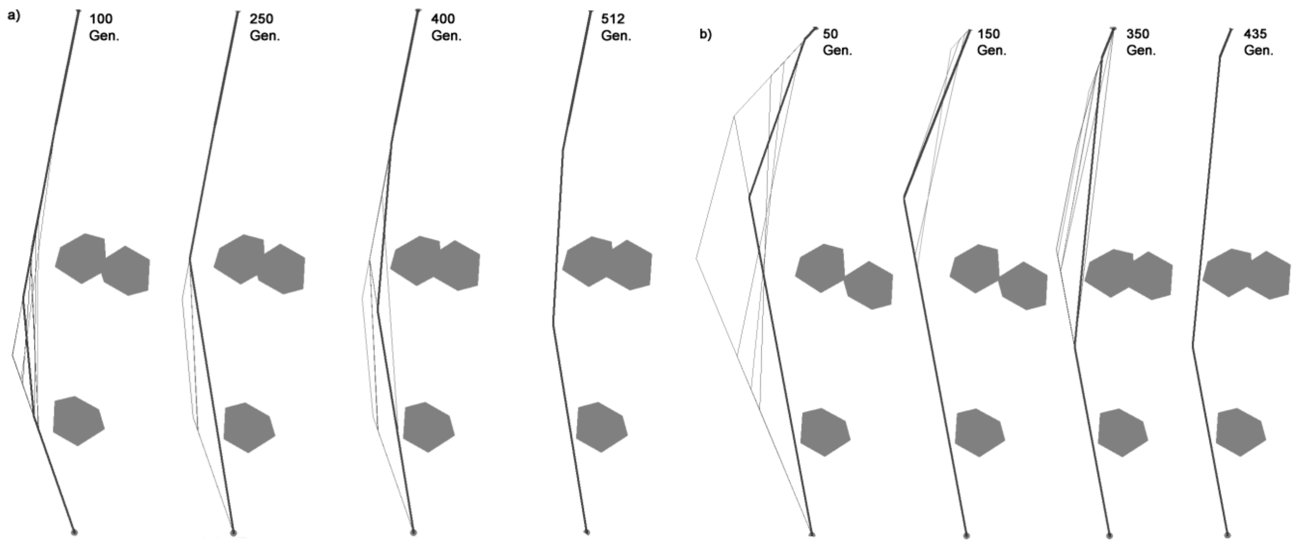

Figure 7. Simulation for Advanced Situation from Figure 3a a) with scaling b) without scaling

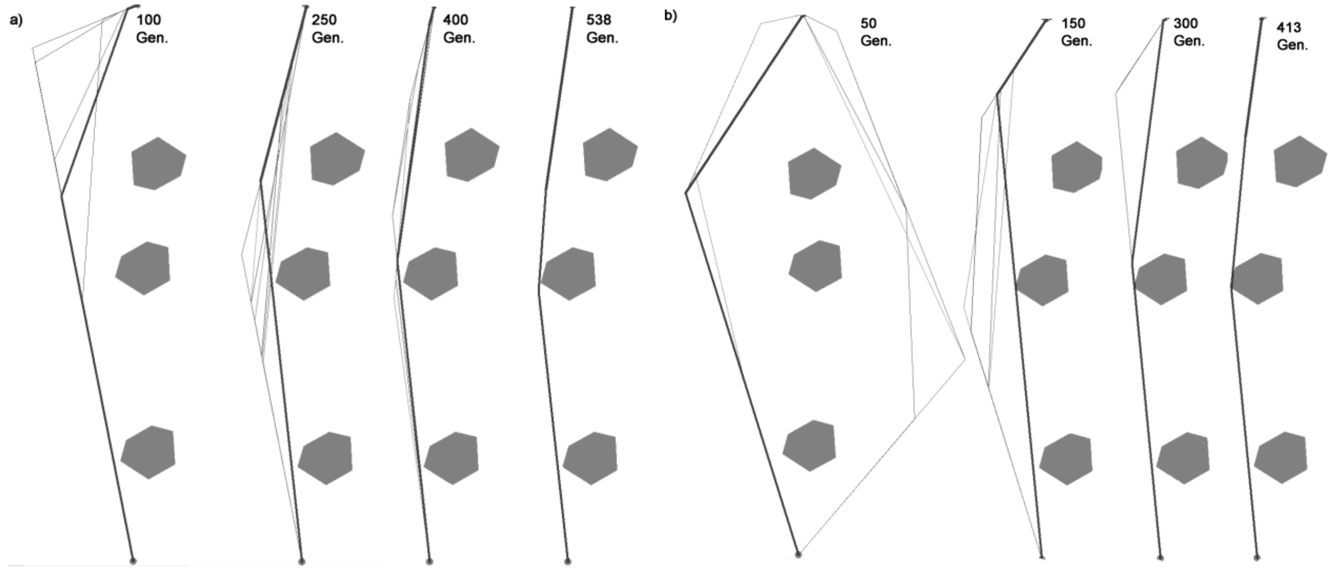

Figure 8. Simulation for Advanced Situation from Figure 3b a) with scaling b) without scaling

6 RESULTS AND CONCLUSIONS

The simulation results above proven that υEP/N++ is able to efficiently find a feasible and acceptable path both in case the scaling is applied and when it is not present, just as shown in detail in [14]. As one can clearly see from figures 2,3 and 4 applying scaling extended the time of the final solution convergence and allowed new paths to form during the generation run. However, the end result was similar, and both versions (with and without scaling) of the algorithm provided comparably good results. However it is also worth to underline that when scaling was applied, the paths calculated by the algorithm became smoother (required course corrections of smaller value), thus more economical as best seen on Fig. 5. As the presented examples are rather non-complex, it is only logical to deduce that applying scaling for this particular problem proven non-necessary, although the computation time extension was barely noticeable. As scaling is a great tool to widen the area of search, it has little effect when faced with a noncomplex challenge, at least for tasks typical for υEP/N++. However, as scaling application did smooth the paths, one can't ignore the improvement noticed, even for a problem of such a small magnitude. Scaling will perform much better in tasks where a path through a hugely populated area has to be plotted, where it's effect will be much better noticeable.

This example research only worked with the power scaling, but as further work is done, where different scaling methods are tested and compared with one another, it could be worth to extend the scaling rating subroutine to be able to run different scaling options.

This paper shows how important it is to set proper scaling strategy that would cope well with the task ahead. Although the experiments presented here were using only two different strategies, their different affect was apparent. One can easily notice that as it is important to select efficient genetic operators, it is as important to match the right scaling scheme. υEP/N++ is equipped with procedures that grade genetic operators and as the run goes, it chooses and utilizes the best one of them. As further research goes, a similar routine can be devices for scaling, so that the algorithm can turn it on, when it seems it can better the results and abandon it, when it becomes redundant. One of the features of Genetic Algorithms is its great scalability, however the parameters have to always be well adjusted to the problem, even to a particular case that is being research. The algorithm has to be able to dynamically adjust to the problem's requirements if it's to be used in the industrial scale.

ACKNOWLEDGEMENTS

The authors thank the Polish Ministry of Science and Higher Education for funding this research under grant no. N514 472039.

BIBLIOGRAPHY

1. Goldberg D.E., *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 1989
2. Farzad A. Sadjali, *Comparison of fitness scaling functions in genetic algorithms with applications to optical processing*, , Proc. of SPIE Vol. 5557, pp. 358
3. Hopgood A., Mierzejewska A., *Transform Ranking: a New Method of Fitness Scaling in Genetic Algorithms*, Research and development in intelligent systems XXV: proceedings of AI-2008, pp.349-355.
4. Michalewicz Z., *Genetic Algorithms + Data Structures = Evolution Programs*. Spriger-Verlang, 3rd edition, 1996.

5. Śmierzchalski R., *Trajectory planning for ship in collision situations at sea by evolutionary computation*, In Proceedings of the IFAC MCMC'97, Brijuni, Croatia, 1997.
6. Śmierzchalski R., *Ships' domains as collision risk at sea in the evolutionary method of trajectory planning*, Computer Information and Applications Vol II, 2004, pp. 117 – 125.
7. Śmierzchalski R. and Michalewicz, Z., *Adaptive Modeling of a Ship Trajectory in Collision Situations at Sea*, In Procceedings of the 2nd IEEE World Congress on Computational Intelligence, ICEC'98, Alaska, USA 1998, pp. 364 - 369.
8. Śmierzchalski 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, 2000, pp.227-241.
9. Śmierzchalski R., Michalewicz, Z., *Path Planning in Dynamic Environments*, chapter in "Innovations in Machine Intelligence and Robot Perception", Springer-Verlag, 2005. pp.135-154.
10. Tidor B., Michael de la Maza, *Boltzmann Weighted Selection Improves Performance of Genetic Algorithms*, MIT, Artificial Intelligence Laboratory, December 1991, pp. 1-18.
11. Wall Mathew, GAlib: A C++ Library of Genetic Algorithm Components, MIT 1996
12. 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., , Lawrence Erlbaum Associates, 1987, pp.95 - 143.
13. Eiben E.A., Smith J.E.: *Introduction to evolutionary computing*, Springer 2003
14. P. Kolendo, R. Śmierzchalski, B. Jaworski: *Scaling Fitness Function in Evolutionary Path Planning Method*, 20th IEEE International Symposium on Industrial Electronics IEEE-ISIE 2011
15. Young-Il Lee, Yong-Gi Kim: *A Fuzzy Collision Avoidance Technique for Intelligent Ship*.