TRANSNAV International Journal on Marine Navigation and Safety of Sea Transportation

Volume 4
Number 2
June 2010

Solving Multi-Ship Encounter Situations by Evolutionary Sets of Cooperating Trajectories

R. Szlapczynski
Gdansk University of Technology, Gdansk, Poland

ABSTRACT: The paper introduces a new approach to solving multi-ship encounter situations by combining some of the assumptions of game theory with evolutionary programming techniques. A multi-ship encounter is here modelled as a game played by "thinking players" – the ships of different and possibly changing strategies. The solution – an optimal set of cooperating (non-colliding) trajectories is then found by means of evolutionary algorithms. The paper contains the description of the problem formulation as well as the details of the evolutionary program. The method can be used for both open waters and restricted water regions.

1 INTRODUCTION

The two main approaches to the problem of determining optimal ship trajectories in encounter situations are the methods based on either differential games or evolutionary method. The methods based on differential games were introduced by Lisowski (Lisowski, 2005). They assume that the process of steering a ship in multi-ship encounter situations can be modelled as a differential game, played by all ships involved, each having their strategies. The game is differential, since it describes the dynamics and kinematics of all ships. The method's main limitations include the high computational complexity and difficulties in handling the stationary obstacles and ship domains other than a circle (its radius being the safe distance).

The second approach – the evolutionary method of finding the trajectory of the own ship has been developed by Śmierzchalski (Śmierzchalski, 1998). It has been among the first methods that utilized the concept of a ship domain instead of safe distance between ships. The method assumes the kinematical model of the own ship and aims to find an optimal balanced trajectory (the balance being between the costs of deviation from a given trajectory and the safety of avoiding static and dynamic obstacles). For a given set of pre-determined trajectories the method finds a safe trajectory, which is optimal according to the fitness function – the optimal safe trajectory. The method's main limitation is that it assumes the target motion parameters not to change and if they do change, the own trajectory has to be recomputed.

The approach proposed here combines some of the advantages of both methods: the low computational time, supporting all domain models and handling stationary obstacles (all typical for evolutionary method), with taking into account the changes of motion parameters (changing strategies of the players involved in a game). Therefore, instead of finding the optimal own trajectory for the unchanged courses and speeds of the targets, a set of optimal cooperating trajectories of all ships is searched for.

The next section presents a formulation of an optimisation problem. Then the structure of an evolutionary population member and its evaluation method are described including a discussion of the constraints and fitness function. Some details on the mechanisms of evolution (including specialised functions and operators) are further provided. Finally the paper summary is presented.

2 OPTIMISATION PROBLEM

It is assumed that we are given the following data:

- stationary constraints (obstacles and other constraints modelled as polygons),
- positions, courses and speeds of all ships involved,
- ship domains,
- times necessary for accepting and executing the proposed manoeuvres.

Obstacles, ship positions and ship motion parameters are provided by ARPA (Automatic Radar Plotting Aid) systems. Ship domain may be determined being given a particular ship motion parameters and length. By default, Coldwell (Coldwell, 1982) domain (an off-centred ellipse) is applied. The necessary time is computed on the basis of navigational decision time and the ship's manoeuvring abilities. By default a 6-minute value is used here.

Knowing these parameters, the goal is to find the set of trajectories, which minimizes the average way loss spent on manoeuvring, while fulfilling the following conditions:

- none of the stationary constraints are violated,
- none of the ship domains are violated,
- the minimal acceptable course alteration is not lesser than 15 degrees,
- the maximal acceptable course alteration is not be larger than 60 degrees,
- speed alteration are not be applied unless necessary (collision cannot be avoided by course alteration up to 60 degrees),
- a ship only manoeuvres, when it is obliged to,
- manoeuvres to starboard are favoured over manoeuvres to port board.

The first two conditions are obvious: all obstacles have to be avoided and the ship domain is an area that should not be violated by definition. All the other conditions are either imposed by COLREGS (Cockroft, 1993) (International Regulations for Preventing Collisions at Sea) or by the economics. In particular, the course alterations lesser than 15 degrees are not always detected by the ARPA systems (and therefore may lead to collisions) and the course alterations larger than 60 degrees are highly inefficient. Ships should only manoeuvre when necessary, since each manoeuvre of a ship makes it harder to track its motion parameters for the other ships ARPA systems.

2.1 *Ship domains and obstacle domains*

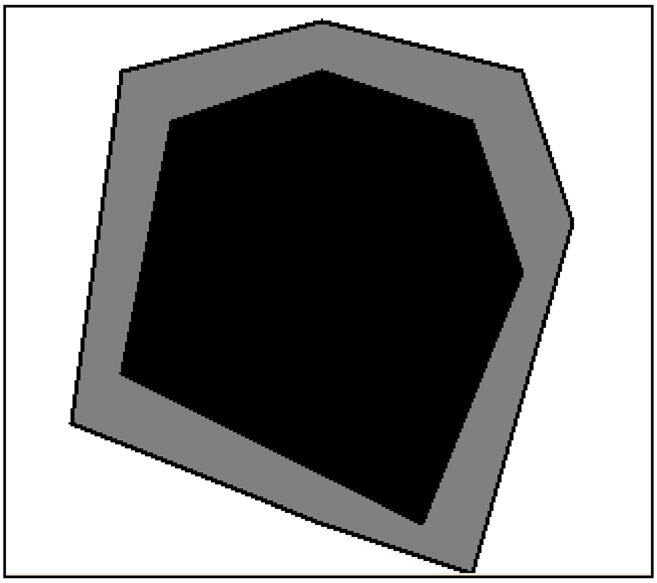


Figure 9 An obstacle (black colour) surrounded by its automatically generated domain (grey colour).

Each stationary constraint is defined as a polygon given as a sequence of the coordinates of its vertices. Each such polygon is then surrounded by additional domain, whose dimensions are computed by the method. A domain size is specified by the user; by default a 0.25 nautical mile distance is used. An example of an obstacle and its domain is shown in Figure 1.

As for the ship domains, the method supports the following ship domain models:

- a circle-shaped domain (traditional domain shape),
- an off-centred circle (domain shape according to Davis)
- an ellipse (domain shape according to Fuji)
- an off-centred ellipse (domain shape according to Coldwell)
- a hexagon (domain shape according to Śmierzchalski)
- a user defined domain (a polygon of user-defined vertices)

The dimensions of those domains are set by the user; the default dimension values are given in Tables 1-3.

Table 2. The dimensions of a circle-shaped domain and Davis domain

	Domain radius [n.m.]	**Domain center moved from the ship's position** Towards starboard [n.m.]	Towards bow [n.m.]
A circle	0.5	0	0
Davis domain	0.5	0.1	0.2

Table 3. The dimensions of a Fuji domain and Coldwell domain

	Ellipse's semi axes [n.m.]	**Domain center moved from the ship's position** Towards starboard [n.m.]	Towards bow [n.m.]
Fuji Domain	0.77, 0.33	0	0
Coldwell domain	0.77, 0.33	0.1	0.2

Table 4 The dimensions of a hexagonal domain

Distance towards bow [n.m.]	**Distance towards stern** [n.m.]	**Distance towards starboard** [n.m.]	**Distance towards port board** [n.m.]
1	0.25	0.6	0.25

3 POPULATION MEMBERS AND THEIR EVALUATION

3.1 *The structure of an individual*

Each individual (a population member) is a set of trajectories (each trajectory corresponding to one of the ships involved in an encounter). A trajectory is a sequence of nodes, each node containing the following data:

- geographical coordinates x and y,
- the speed between the current and the next node.

3.2 *The evaluation of an individual*

The basic piece of data used during the evaluation phase of the evolutionary process is the average way loss computed for each individual (a set of cooperating trajectories). Some of the constraints also must be taken into account during the evaluation. This includes violations of ship domains and violations of stationary constraints: both must be penalized and those penalties – must be reflected in the fitness function. However, as for the other constraints, there are two possible approaches:

1 These constraints can be incorporated in the fitness function.
2 Meeting these constraints can be achieved by applying certain rules on various steps of the evolutionary process simultaneously:
   - when generating the initial population,
   - during mutation,
   - handling the constraints violations by fixing functions operating on individuals prior to their evaluation

The second approach has been chosen here, because of its faster convergence due to:

- its simpler fitness function,
- avoiding the production of individuals (during the mutation phase), whose low fitness function value can be predicted.

Violations of the first two constraints (stationary ones and ship domains) are penalized as follows. For each ship and its set of stationary constraint violations, an obstacle collision factor is computed as given by (4). For each ship and its set of prioritised targets a ship collision factor is computed as given by (3). The reason, why only collisions with prioritised targets are represented in evaluation is because the manoeuvres must be compliant with COLREGS. If a ship is supposed to stay on its course according to the rules, the collision is ignored so as not to encourage an unlawful manoeuvre. In case of collision with prioritised target, the author's measure – approach factor f_{min} (Szlapczynski, 2006b) is used to assess the risk of a crash. Approach factor has been defined as the scale factor of the largest domain-shaped area that is predicted to remain free of other ships throughout the whole encounter situation.

Each individual (a set of trajectories) is being assigned a value of the following fitness function (1):

$$fitness = \sum_{i=1}^{n} \left[tr\_fit_i \right], \tag{1}$$

where:

$$tr\_fit_i = \left(\frac{tr\_length_i - way\_loss_i}{tr\_length_i} \right) * sf_i * of_i, \tag{2}$$

sf_i – ship collision factor of the i-th ship computed over all prioritised targets:

$$sf_i = \prod_{j=1, j \neq i}^{n} \left(min(\,fmin_{i,j}, 1) \right) \tag{3}$$

of_i - obstacle collision factor of the i-th ship com puted over all stationary constraints:

$$of_i = \prod_{k=1}^{m} \left(\frac{360° - collision\_course\_range_j}{360°} \right) \tag{4}$$

n – the number of ships [/],

m – the number of stationary constraints [/],

i – the index of the current ship [/],

j – the index of a target ship [/],

k – the index of a stationary constraint [/],

$fmin_{i,j}$ – the approach factor value for an encounter of ships i and j [/],

$collision\_course\_range_j$ – the range of forbidden courses of the ship i computed for the stationary constraint j in the node directly preceding the collision. [/].

To detect the stationary constraint violations of an individual, all of the trajectories are checked against all of the constraints (which are modelled as polygons) and the collision points are found. Analogically, to detect the domain violations of an individual, all of its trajectories are checked against each other to find potential collision points. Unfortunately, applying ship domains instead of safe distances results in higher computational complexity and the process of evaluation consumes the majority of the evolutionary algorithm's computational time. Therefore it has been decided to invest some computational time in specialised functions (validations and fixing) and specialised operators, which speed up the convergence to optimal solution, thus decreasing the number of the generations – and consequently – decreasing the number of evaluations.

4 EVOLUTIONARY PROCESS

4.1 *Generating the initial population*

The initial population contains three types of individuals:

- a set of original ship trajectories – segments joining the start and destination points
- sets of safe trajectories determined by other methods,
- randomly modified versions of the first two types – sets of trajectories with additional nodes, or with some nodes moved from their original geographical positions.

The first type of individuals results in an immediate solution in case of no collisions, or in faster convergence in case of only constraint violations. The second type provides sets of safe (though usually not optimal) trajectories. They are generated by means of two methods: one operating on raster grids (Szlapczynski, 2006a) and the other planning a sequence of necessary manoeuvres (Szlapczynski, 2008). Finally, the third type of individuals (randomly modified individuals of the previous two types) is used to generate the majority of a diverse initial population and thus to ensure the vast searching space.

4.2 *Trajectory validations and fixing*

Representing all of the constraints in the fitness function would result in a very slow progress of the evolutionary algorithm. A good example here is the rule, according to which a course alteration should not be lesser than 15 degrees. Had this constraint been taken into account by the fitness function, slight course alterations, (for example about 5 degrees) would be penalized severely. On the other hand, individuals with no course alterations or with large course alterations would not be penalized. The individuals with no course alterations as well as those with large course alterations would likely be chosen for crossing and would spawn offspring, which again – would probably be penalized for slight course alterations. Therefore some of the constraints are applied as validating and fixing functions. Each trajectory of an individual is analysed and in case of unacceptable manoeuvres (such as slight course alterations), the nodes being responsible are moved so as to round a manoeuvre up or down to an acceptable value.

4.3 *Specialised operators*

The evolutionary operators, which have been used in the current version of the method, can be divided into the following groups.

1 Crossing operators: two types of crossing have been used, both operating on pairs of individuals and used to generate offspring:

   - an offspring inherits whole trajectories from both parents.
   - each of the trajectories of the offspring is a crossing of the appropriate trajectories of the parents.

2 Operators avoiding collisions with prioritised ships: three types of these operators have been used, all operating on single trajectories. If a collision with a prioritised ship has been registered, depending on the circumstances (coordinates of the collision point, way loss, number of target ships and number of nodes within a trajectory) one of the following operators is chosen:

   - node moving: the node closest to the collision point is moved away from it,
   - segment moving: two nodes, which are closest to the collision point are moved away from it,
   - node insertion: a new node is inserted between the two nodes closest to the collision point in such a way that the collision will probably be avoided,

   None of these operations guarantees avoiding the collision with a given target but they are likely to do so and therefore highly effective statistically and suitable for the evolutionary purposes.

3 An operator avoiding collisions with obstacles. a course alteration manoeuvre is made (a new node is inserted) in such a way, that the new trajectory segment does not cross a given edge of an obstacle (polygon).

4 Random operators: three types of these operators have been used, all operating on single trajectories. They are mostly used when a given trajectory does not collide with any prioritised trajectories; otherwise one of the abovementioned collision avoidance operators is more likely to be used. These random operators include:

   - node insertion: a node is inserted randomly into the trajectory,
   - node deletion: a randomly selected node is deleted,
   - nodes joining: two neighbouring nodes are joined, the new node being the middle point of the segment joining them,
   - node mutation: a randomly selected node is moved (its polar coordinates are altered).

   A trajectory mutation probability decreases with the increase of the trajectory fitness value (2), so as to mutate the worst trajectories of each individual first, without spoiling its best trajectories. In the early phase of the evolution all random operators: the node insertion, deletion, joining and mutation are equally probable. In the later phase node mutation dominates with its course alteration changes and distance changes decreasing with the number of generations. For node insertion and node mutation instead of Cartesian coordinates x and y, the polar coordinates (course alteration and distance) are mutated in such a way that the new manoeuvres are between 15 and 60 degrees. As a result, fruitless mutations (the ones leaving to invalid trajectories) are avoided for these two operators. Operating on polar coordinates (course and distance) instead of Cartesian x and y coordinates also makes it more likely to escape the local optimums because manoeuvres both valid and largely differing from the past ones are more likely to be generated.

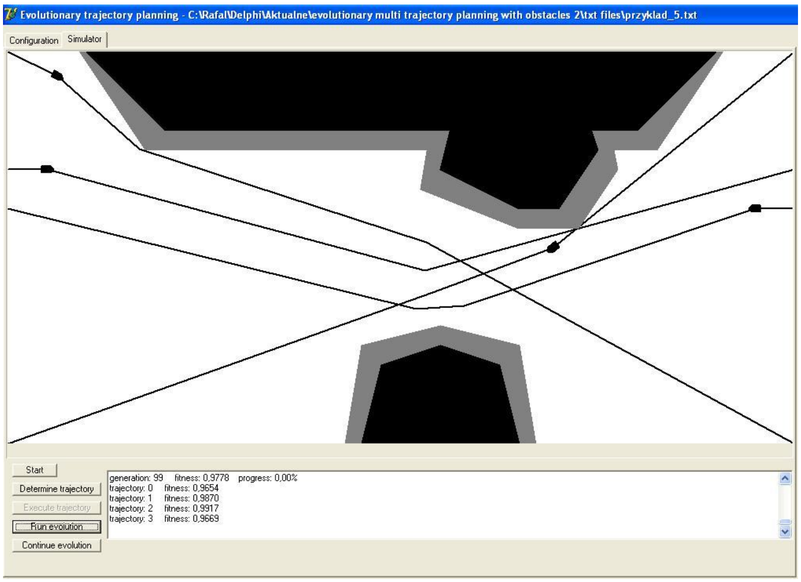
4.4 *Selection*

In the currently developed version of the method the truncation selection has been applied with the truncation threshold of 50%. Although this kind of selection means a loss of diversity, it has the benefit of a fast convergence to a solution. When combined with abovementioned, specialised operators (especially mutation using polar coordinates and operators aiming at collision avoidance), the solution, which the process converges to, is usually the optimal one.

4.5 *Stop condition*

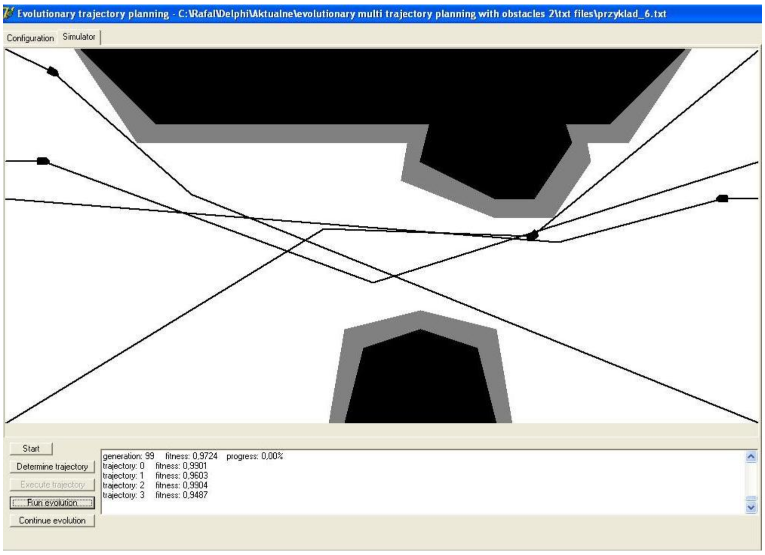
The evolutionary process is stopped if one of the following happens:

- the maximum number of generations is reached,
- the time limit is reached,
- further evolution does not bring significant improvement.


Evolutionary trajectory planning - C:\Rafal\Delphi\Aktualne\evolutionary multi trajectory planning with obstacles 2\txt files\przyklad_5.txt
Configuration
Simulator
Start
Determine trajectory
Execute trajectory
Run evolution
Continue evolution

```
generation: 99    fitness: 0,9778    progress: 0,00%
trajectory: 0    fitness: 0,9654
trajectory: 1    fitness: 0,9870
trajectory: 2    fitness: 0,9917
trajectory: 3    fitness: 0,9669
```

Figure 10 A final set of cooperating evolutionary trajectories with Fuji domain applied.


Evolutionary trajectory planning - C:\Rafal\Delphi\Aktualne\evolutionary multi trajectory planning with obstacles 2\txt files\przyklad_6.txt
Configuration
Simulator
Start
Determine trajectory
Execute trajectory
Run evolution
Continue evolution

```
generation: 99    fitness: 0,9724    progress: 0,00%
trajectory: 0    fitness: 0,9901
trajectory: 1    fitness: 0,9603
trajectory: 2    fitness: 0,9904
trajectory: 3    fitness: 0,9487
```

Figure 11 A final set of cooperating evolutionary trajectories with Coldwell domain applied.

5 SIMULATION EXAMPLES

Two examples - simulation results are shown in Figure 2 and Figure 3. In both cases the same scenario has been used, only with different ship domain model applied. Fuji domain has been applied in the situation depicted in Figure 2 and Coldwell domain – in the situation depicted in Figure 3. As a result, slight differences in sizes and shapes of the domains used have caused differences in the trajectories. Most notable one is that the ship starting in the upper right corner of the pictures had to perform an extra manoeuvre to the starboard in Figure 3, to avoid a collision with one of the other ships. The general tendencies of movement of other ships have remained unchanged however. All of the ships chose manoeuvres to starboard, unless course alteration to port board was forced by stationary constraints.

6 SUMMARY

In the paper an evolutionary approach to solving multi-ship encounter situations has been proposed. This approach is a generalization of evolutionary trajectory determining: a set of trajectories of all ships involved, instead of just the own trajectory, is determined. A method implementing this new approach has been developed. The method avoids violating the target ship domains and the given stationary constraints, while minimizing way loss and obeying the COLREGS. It also benefits from a number of author-designed specialized functions and operators, resulting in faster convergence to the optimal solution.

REFERENCES

Cockroft A.N., Lameijer J.N.F 1993: A Guide to Collision Avoidance Rules, Butterworth-Heinemann Ltd.

Coldwell T.G. 1982. Marine Traffic Behaviour in restricted Waters, *The Journal of Navigation* vol. 36: pp. 431-444.

Lisowski J.2005 Dynamic games methods in navigator decision support system for safety navigation, Proceedings of the European Safety and Reliability Conference vol. 2, pp 1285 – 1292.

Szlapczynski R. 2006a. A new method of ship routing on raster grids, with turn penalties and collision avoidance, *The Journal of Navigation* vol. 59, issue 1: pp. 27-42.

Szlapczynski R. 2006b. A unified measure of collision risk derived from the concept of a ship domain, *The Journal of Navigation* vol. 59, issue 3: pp. 477-490.

Szlapczynski R 2008. A new method of planning collision avoidance manoeuvres for multi target encounter situations, Journal of Navigation, 61, issue 2, pp. 307-321.

Śmierzchalski R. 1998 Synteza metod i algorytmów wspomagania decyzji nawigatora w sytuacji kolizyjnej na morzu. Prace Naukowe Wyższej Szkoły Morskiej w Gdyni.