MULTI-SCALE AUTOMATIC ROUTE PLANNING ALGORITHMS FOR SEA-GOING VESSELS

Prof., Dr. Capt. Nguyen Viet Thanh
Dean of Navigation Faculty, Vietnam Maritime University
Email: thanhdktb@gmail.com
Prof., Dr. C/E. Luong Cong Nho
President of Vietnam Maritime University
Email: luongcongnho@vimaru.edu.vn
Dr. Nguyen Minh Duc
Navigation Faculty, Vietnam Maritime University
Email: nguyenminhduc@vimaru.edu.vn

Abstracts:

Collision Avoidance has always been one of the primary tasks of the ship officer, both in congested water and at the open sea. Under the stress of workloads onboard, the officer of watch might commit mistakes now and then and this may result in great losses to the ship, cargo, environment and even the human lives. On a larger scale, for a greener and more efficient shipping industry, weather routing has long been seen as a worthy solution. Thanks to the technology advancement, the track control system is - or will soon be - installed on board, together with different means of collecting target, or floating objects, motions objects as well as short and long-range weather information. Then, in this study, we will set the focus on the general structure of a route planning program in which different algorithms, e.g. isochrones method, can be applied to produce transoceanic routes basing on weather routing theories and then, in a smaller scale, a suitable algorithm might be used to generate collision avoiding routes for the ship while following the former. For collision avoiding purpose, an adaptive Bacterial Foraging Optimization (BFO) algorithm and different collision avoiding criteria such as the ship domain will be introduced and applied for a large amount of encountering cases at sea. Also, in order to apply the traffic rules, an appropriate cost function will be suggested so that an extra cost will be embed for a route that causes the own ship to break the rules while taking collision avoiding action. Later, the simulation result will be presented to show the effectiveness as well as the reliability of the adaptive BFO algorithm. As the required time for calculation is less than 10 sec for all the cases, it is quite possible for the algorithm to be applied in real time. The combination of these different scales of route planning algorithm will make it possible for a safe, effective and leisured navigation.

Keywords: Route planning, collision avoidance, weather routing, bacteria foraging optimization, ship domain.

1. Introduction

Thanks to the science and technology developments and their extensive application in weather forecasting service and in shipping industry, navigation has become much more relaxing. The officers onboard a modern ship have at their hands long range weather forecast with reasonable accuracy as well as a huge amount of information of the prevailing traffic condition around the ship, including the existence and motion of nearby targets as well as other static constraints. The problem, among others, they have to face now is how to use the available information properly so as to ensure a safer and more efficient navigation. Unfortunately, maneuvering has never been an easy task and traffic accident are still happening mainly due to human errors in judging the information and/or in making decision in an environment where they are burdened with all the paper works and overloaded with the information itself.

During the last several decades, different researches have been conducted on the ocean passage voyage designing as well as collision avoiding strategy planning subjects.

For ocean passage voyage, the Isochrones method and its modifications [1] have been widely in use by weather routing service providers. The main deterrence of these approaches is the phenomenon that some isochrone points might be stuck at landmass shore. The behavior is due to the inability of creating points for the following isochrones from a point surrounded by shorelines. Furthermore, the method is more or less based on Dynamic Programming method which is not suitable for time-varying environments, which is actually the case we normally witness at sea.

For collision-avoiding route producing purposes, the classical approach so far is to find a collision-avoiding course, i.e. a heading, for the own ship based on CPA/TCPA criterion [2][3]. This effectiveness of this strategy greatly reduces in situations where several target ships are involved. Another shortcoming is that the own ship maneuvering characteristics and traffic rules are not properly taken into consideration.

Aiming at stimulating a safer and more efficient shipping, the study focuses on the following main tasks automatically from all the available information:

- Planning long distance weather-route (weather routing)
- Producing optimal collision-avoiding strategy while following the weather route

The overall structure of the system is illustrated in the Fig.1. Taking into consideration the ship performance, i.e. the speed characteristic in waves, shoreline contours and the weather forecasts and changes in these forecasts, the optimal route for the own ship can be consistently calculated and updated.

While sailing on the above-mentioned route, the own ship has to take maneuver to avoid collision with other ships, floating objects as well as other dangerous or prohibited areas. Therefore, the actual track for the ship, i.e. a safe track in the proximity to the weather route, must be calculated whenever there exists a risk of collision. The target information can be acquired through the ship radar, the automatic identification system (AIS) or any other sensors such as a camera. Later, maneuvering commands can be transmitted to the ship steering-gear through an onboard local network in a task referred to as "tracking control", which has been a subject of extensive studies. At this scale, the system contains different blocks solving individual tasks, namely:

- The Safe-Passage-Checking block is on watch to detect any arising risk of collision, including the coming of a new TS, the changes in TS course and/or speed, and the deviation of OS from its safe track.
- The Optimal-Route-Generator block is to conduct real-time calculation of an optimal route for avoiding collision.
- The Tracking-Control block is to maneuver OS to follow the safe route calculated above.

Within the scope of this paper, we concern only with the collision risk detecting and route generating tasks, which are still in the early stage of development. An Adaptive Bacterial Foraging Algorithm (Adaptive-BFOA) will be suggested and applied in the current study for route-producing purpose.

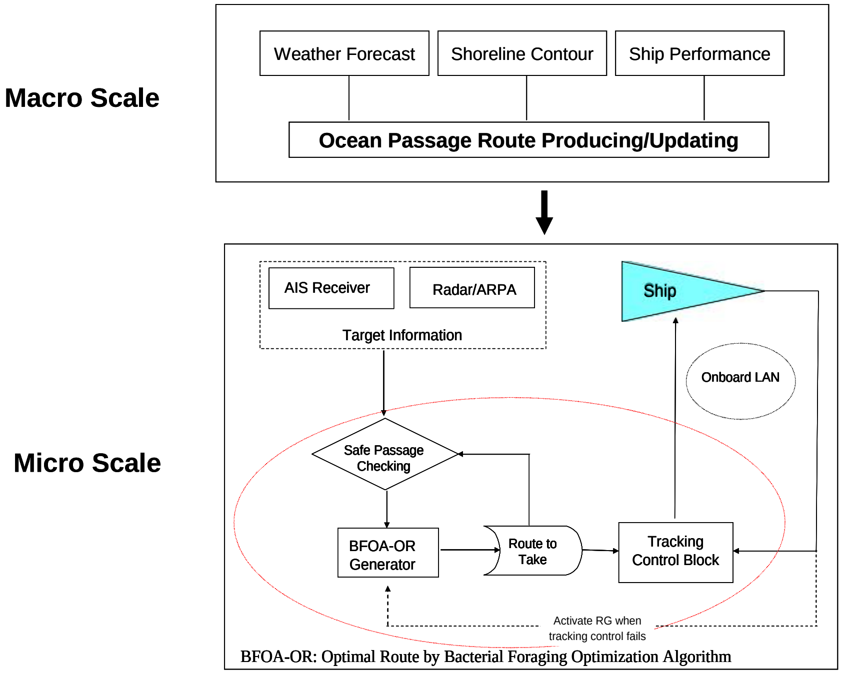


Figure 1 Overview of the Collision Avoiding Support System

2. Principles of Route Generation

2.1 General

Assume that the OS is currently at a point A on its weather-route when there arises the risk of collision due to the encounter with a TS as shown in Fig.2. To produce collision-avoiding route, a grid system between the point A and an end-point B selected on the original route of OS is built in the navigable water between these 2 points. The grid consists of grid-lines with a number of points on each line. Distance between the grid-lines, distance between points on a line as well as the number of points are treated as the designing parameters of the grid and should be chosen in a way to harmonize between the accuracy of the route to be determined and the volume of calculation required. For example, given the grid-width, if the number of points on each line is increased, the collision-avoiding route for our ship might be smoother but the volume of calculation will grow accordingly.

Restricted areas, e.g. military zones, are demarcated by suitable polygons such as the pentagons in Fig.2. Those polygons can be automatically or manually input before the voyage for the whole planned route and are kept in the voyage-database. They will later be recalled when the OS approaches these specific sea regions.

In the same manner, limiting lines can be manually input to limit the area around the planned route in which the ship should be maintained.

A safe route for the OS is the shortest route from the starting-point, via exactly 1 grid-point on every line to reach the end-point that does not cause the ship to enter a restricted area, to go out of limiting lines or to be in risk of collision with any TS.

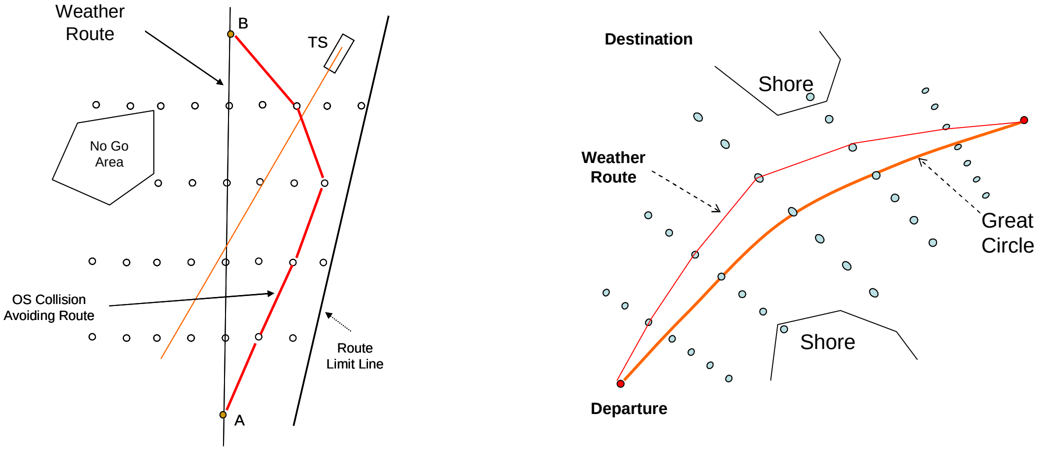


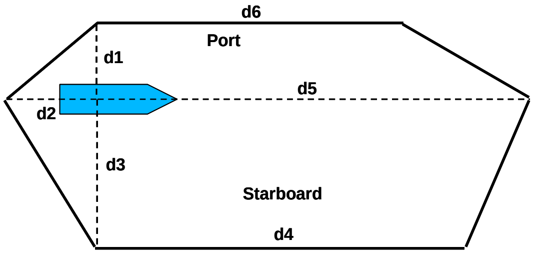
Figure 2 Route Producing Principles (Weather Route - left, Collision Avoiding Route - Right)

Similarly, to produce weather-route for the own ship, a suitable grid system is to be built around the great circle course from the departure point to the destination. A route for the ship must be through one grid point on each grid line and the shore contours (depth contours should be used instead) must be kept clear away. Grid lines are chosen in a way that lines in the middle section are longer than those nearer to the two ends. This helps increase the flexibility of the algorithm in computing optimal weather route.

2.2 Collision Risk Assessing Criteria for Collision Avoidance

2.2.1 Collision Risk Assessing Criteria

In order to verify if the collision-risk does exist and/or a route is safe for the OS, given TS movements, suitable risk assessing criteria must be employed. Taking into consideration the dimension of the water available for the OS to maneuver, different criteria may be used for assessing collision risk for the ship-to-ship encounters at sea, including the Ship Domain, the Bumper Model, the OZT (obstacle zone by target), etc. In this paper the Ship Domain is introduced and employed to conduct simulation. The Ship Domain, suggested by R. Smierzchalski et al in their work [4], is a region of hexagon shape surrounding the ship as shown in Fig.3. The appearance of a navigational constraint in the vicinity of the domain contour means the appearance of a navigational risk. The risk increases as a result of the decreasing distance to the detected constraints. Sizes of a ship domain on its course are computed from its length and speed, together with the selected minimum time and distance to the closest point of approaching (TCPA, DCPA) as the followings (see Fig.4).



$$d1 = DCPA/4$$
$$d2 = DCPA/4$$
$$d3 = B \times V^{0.44}/2$$
$$d4 = TCPA \times V/2$$
$$d5 = (L \times V^{1.26} + 30 \times V)/2$$
$$d6 = TCPA \times V/2$$
$$where: L,\ B\ are\ ship\ lenth\ and\ breath\ [NM],\ V\ is\ the\ ship\ speed\ [kts]$$

Figure 3 Ship Domain

As the name itself implies, domains of OS and TS should never overlap each other. The values of TCPA, DCPA parameters can be decided by the watch keeping officer basing on his own experience and the weather condition.

2.2.2 Limits of Weather

In considering the safety of the ship and the cargo, special attention should be paid not only on the speed reduction but also on the ship rolling, pitching and heaving characteristics in heavy weather. For example, in a given sea state, the ship heading should be chosen so as avoid roll resonance or parametric roll resonance. Then, a path from one way-point to another is deemed viable only if on that path, the ship stability and strength are ensured and it is free from all the notorious adverse effects.

In this paper, the following two indexes are employed:

- Limit wave height
- Limit roll resonance frequency

2.3 Route Cost Evaluation

2.3.1 Collision-Avoiding Route Cost

A route for the OS to avoid collision should be, first and foremost, safe. Additionally, it should also be economical, i.e. time required for the ship to reach destination by that route be small, and the traffic rules (as set out in Colreg 72) be satisfied as far as the circumstance allows.

Keeping in mind all the above factors, in this study, we suggest the formula for evaluating the cost of a route as followings (1):

$$Q = \left(\sum_i T_i\right) \times \left(1 + \sum_j K_j\right) \tag{1}$$

$i = 1\ to\ Number\ of\ Connections\ on\ the\ Route$

$j = 1\ to\ Number\ of\ TS\ involved\ in\ the\ situation$

T_i : $Time\ required\ to\ travel\ connection\ i^{th}\ on\ the\ Route$

K_j : $Additional\ cost\ due\ to\ rule\ infringement\ when\ avoiding\ TS\ j^{th}$

In (1), the connection ith is a part of the route that connects a point on line ith and a point on line (i+1)th of the grid. The first part of the formula therefore expresses the total-time needed for the OS to reach the end-point by following the route. The second part of (1), on the other hand, represents the added penalty (i.e. added cost) of a route if on that route, the OS infringes upon the rules of taking action to avoid collision with the target jth.

The choice of the coefficient K reflects the level of pressure on the ship officer if a rule-violating strategy to avoid collision is selected. From the onboard experience and simulation, the value of K in (1) for different encountering case between the OS and a certain TS has been chosen as the followings [8]

(1) Passing a Head-on TS on OS Starboard side: K = 0.2.
(2) Passing TS Crossing from Starboard side on OS Starboard side (OS give-way): K = 0.05.
(3) Turning to Port while TS is crossing from Port side (OS Stand-on): K = 0.1.

(4) Turning to Starboard while TS is behind the OS traverse axis on starboard side and overtaking: K = 0.2.

(5) Turning to Port while TS is behind the OS traverse axis on port side and overtaking: K = 0.2.

(6) Otherwise, K = 0.0.

The problem is now to determine a safe route that minimizes the value of Q, i.e. an optimization problem.

2.3.2 Weather-Route Cost

One of the prominent advantages of the Adaptive-BFO algorithm (Section 3) is its flexibility for choices of the cost function. However, in this paper, the typical minimum time route criterion is applied and then, the cost function has the following simple form:

$$Q = \sum_i T_i \tag{2}$$

T_i : *Time required to travel connection* i^{th} *on the Route*

2.4 Own Ship Maneuverability

For the generated collision-avoiding route to be viable, the maneuvering characteristics of the OS must be properly taken into account, including the OS speed, speed reducing and increasing due to rudder actions, radius of the turning circle for different rudder angles. Being highly non-linear [5], these characteristics must be defined in advance for the OS by sea-trial. Then, the result should be tabulated for quick access to draw the ship position at a given time if a certain action is taken.

In order to compute an optimal weather route, the ship speed reduction in different conditions of wave height and direction must be applied. For our ship model, the speed characteristic is as shown in Fig. 4 for different angle of wave attack, namely 0°, 60°,120°,180°.

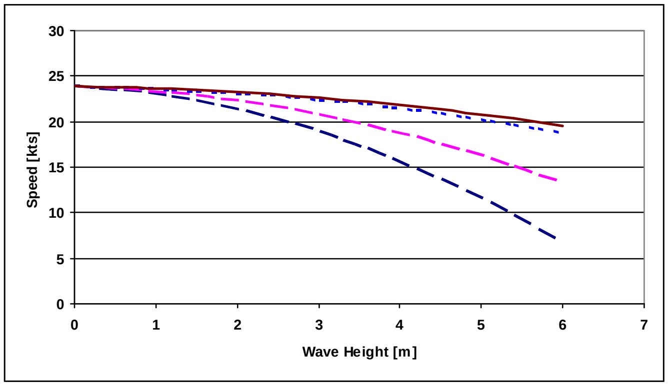


Figure 4 Ship Speed Reduction in Wave

3. Collision-Avoiding Route Generation by Adaptive-BFOA

The optimization problem in route producing is a combinatorial problem which highly non-linear and therefore can not be solved analytically. Then, rather than looking for the true optimal, an evolutionary algorithm is employed here in searching for a solution (a route) with acceptable accuracy. An adaptive-bacterial-optimization-algorithm is therefore constructed and applied in this study.

3.1 Classical Bacterial Foraging Optimization Algorithm

First introduced by Passino [7] in 2002, BFOA has been the subject of many researches in the last several years and generally considered a promising solution for a variety of distributed optimization. The algorithm is a population-based numerical optimization method which is simple but powerful. Applying BFOA, the optimal searching process can be achieved by letting a swarm of bacteria undergoing the following four fundamental steps:

- Chemotaxis: A process simulating the movement of an E.coli cell via swimming and tumbling actions. Swimming is the action of a bacterium to slide in a certain direction in searching for the region of better nutrient concentration. Tumbling, on the other hand, is the

action of starting the search in a new direction.
- Swarming: The grouping behavior of the bacteria swarm that causes them to aggregate to traveling groups. It affects the distribution of the swarm (solution set) in the solution space.
- Reproduction: A process in which less healthy bacteria die while the healthier ones reproduce so as to keep the total number of bacteria fixed. In our application it is equivalent to the duplication of search in more favorable regions.
- Elimination and Dispersal: A process imitating the phenomenon that sudden changes in the environment cause bacteria in certain region dead while the same number of bacteria are added and randomly distributed over the search space. It helps the algorithm to diversify the search all over the solution space. The search therefore will not be trapped in local optimum.

3.2 Route Generation by BFOA

As mentioned earlier, the overall route-generating process can be summarized as shown in the flow chart in Fig.5

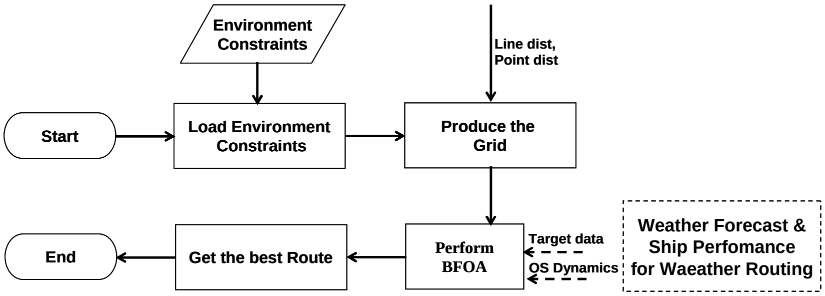


Figure 5 Flow Chart of Route Generation

The collision risk assessment criteria or the limits of weather in section 2.2 are used for checking the safety of a route and a suitable cost function in section 2.3 is applied to judge the fitness level for that route.

A bacterium position (or a solution, a route equivalently) is a combination S of the grid points (one point on each line) as denoted in (2). The search direction (or the tumble of a bacteria) can be expressed as the vector V in (3) and a swim in a direction can be carried out as in (4) with D to be the length (or distance) of swim.

$$S = \left[p(1),\ p(2), \ldots,\ p(i), \ldots,\ p(N)\right] \qquad (2)$$

$$V = \left[v(1), v(2), \ldots,\ v(i), \ldots, v(N)\right] \qquad (3)$$

$$S' = S + V.D \qquad (4)$$

$$\begin{aligned}
&\textit{where}\\
&\quad p(i) \in \textit{[1 to Number of points on grid line } i^{th}\textit{]}\\
&\quad v(i) = 0 \textit{ or } v(i) = \pm 1\\
&\quad \textit{N: Number of grid lines}\\
&\quad D : \textit{the seletected swim \_ length}
\end{aligned}$$

Due to the high dimensionality of the problem, i.e. large N, there are a huge number of combinations of 0 and 1 in vector V. In this study, the choice of V is limited to the following combinations (5).

$$\begin{aligned}
V_1 &= [0,\ldots,0,1,0,\ldots,0]\ \textit{i.e. } v(i) = 1 \textit{ if } i = k;\ v(i) = 0 \textit{ otherwise}\\
V_2 &= [0,\ldots,0,1,1,0,\ldots,0]\ \textit{i.e. } v(i) = 1 \textit{ if } k \le i \le k+1;\ v(i) = 0 \textit{ otherwise}\\
V_3 &= [0,\ldots,0,1,1,1,0,\ldots,0]\ \textit{i.e. } v(i) = 1 \textit{ if } k \le i \le k+2;\ v(i) = 0 \textit{ otherwise}\\
V_4 &= [0,\ldots,0,-1,0,\ldots,0]\ \textit{i.e. } v(i) = -1 \textit{ if } i = k;\ v(i) = 0 \textit{ otherwise}\\
V_5 &= [0,\ldots,0,-1,-1,0,\ldots,0]\ \textit{i.e. } v(i) = -1 \textit{ if } k \le i \le k+1;\ v(i) = 0 \textit{ otherwise}\\
V_6 &= [0,\ldots,0,-1,-1,-1,0,\ldots,0]\ \textit{i.e. } v(i) = -1 \textit{ if } k \le i \le k+2;\ v(i) = 0 \textit{ otherwise}\\
V_7 &= [0,\ldots,0,1,0,\ldots,0,-1,0,\ldots,0]\ \textit{i.e. } v(i) = 1 \textit{ if } i = k_1; v(i) = -1\\
&\qquad\qquad\qquad\qquad\qquad\qquad \textit{if } i = k_2;\ v(i) = 0 \textit{ otherwise}\\
&\textit{where } k,\ k_1, k_2 \textit{ are random value produced at each tumble} \qquad (5)
\end{aligned}$$

Then, a tumble is a probabilistic choice of a vector V from the above set, with V1 and V4 are intentionally chosen more frequently than V2 and V5. In turn, the later vectors are used slightly more often than the rest. After each tumble, the bacterium undergoes one or several swims, depending on the success of the search in that direction.

The effect of the chemotaxes is illustrated in Fig.6 for a case of collision-avoiding consideration. It can be easily seen here that the swarm (6 bacteria) are initially scattered over the solution space i.e. the grid. After several chemotaxes, they tend to converge to their local optimum, representing different locally optimal strategies which the OS can employ to avoid collision with the given TS.

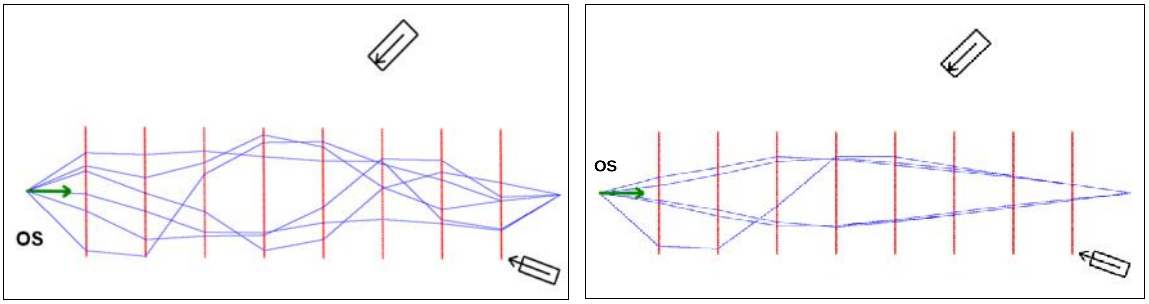


Figure 6 Random Bacteria Positions (routes – left) and Improved Positions after Chemotaxis (right)

3.3 Adaptive BFOA Pseudo-Code for Route Generation

The overall route-producing algorithm can be illustrated by the following pseudo-code: It should be noted that to improve the searching performance of the algorithm, a swim-length adapting scheme (*) has been employed, in which the swim-length is modified after a number of unsuccessful chemotaxes. This allows the bacteria to reach the optimal quickly in the early stages and then approach this optimal steadily. Additionally, the bacterium is not killed until it has performed a number of chemotatic moves (**). This is to ensure that all regions of the grid can be thoroughly explored.

```
a) Initialization:
Initialize_Grid(N_line, N_point, D_point);
For bac = 1 to Ns
    Initialize_Bacterium(B(bac));
Next bac

b) Evolution
For cycle = 1 to N_cyc
        For bac = 1 to Ns
                For chemo = 1 to Nc
                        PerformChemotaxis(B(bac));
                Next chemo

                If (N_fail > N_to_[lar/med/sma]) then (*)
                        ConvertSwimLengthFrom[large/medium/small]_To[medium/small/large]();
                End if
        Next bac

    Sort_Bacteria&RouteCost_Ascendingly(B(Ns), Q(Ns));

        For die_no = 1 to Nr
                If (ChemotaxisCount(B(die_no)) > N_die) then (**)
                        Kill_bacterium(B(die_no));
                        B(die_no)=Reprocude(B(Ns - die_no));
                End if
        Next die_no

        For disperse_no = 1 to Nd
                rand = produce_random_interger();
                Initialize_Bacterium(B(rand));
                Next disperse_no
Next cycles

c) Termination
Sort_Bacteria&RouteCost_Ascendingly(B(Ns), Q(Ns));
Return B(1);
```

Where the variables and designing parameters are defined as:

N_line: Number of lines on the grid

N_point: Number of points on a grid line

D_point: Distance between points on a line

N_cyc: Number of cycles in the algorithm i.e. number of generations of the bacteria population.

Ns: Number of bacteria in the population

B(Ns): Bacteria population (Ns members)

Q(Ns):Route Cost of the bacteria (Ns routes)

Nr: Number of bacteria died/reproduced in a cycle

Nd: Number of bacteria eliminated/dispersed in a cycle

Nc: Number of chemotaxes of a bacterium in a cycle

N_to_lar: Number of unsuccessful chemotatic move before converting the move-length from small to large

N_to_med: Number of unsuccessful chemotatic move before converting the move-length from large to medium

N_to_sma: Number of unsuccessful chemotatic move before converting the move-length from medium to small

N_die: Number of chemotatic moves of bacteria before maturing

N_fail: Number of unsuccessful chemotatic moves of a bacterium

4. Simulation Studies

The Adaptive-BFOA has been applied for a set of scenarios to verify its efficiency. In this paper, only several scenarios will be shown as illustrations for the whole idea. The overall safety checking and route producing program has been coded by VB language for both the collision-avoiding route and optimal weather .route.

4.1 Collision-Avoiding Route Generation

a) Scenario 1: This scenario is an encounter case at open sea in which the OS has to take action to avoiding collision with the 3 crossing-TS. The ship domain is applied here, instead of the bumper model for risk assessing. It can also be seen here that the bacteria population initially scattered all over the grid and then gradually converge to the region of the best solution. As the encounter is at open sea, the grid can be largely widened to make more room for the OS to maneuver. The generated collision-avoiding route is appropriate from the experienced seamanship viewpoint and properly satisfies the traffic rules.

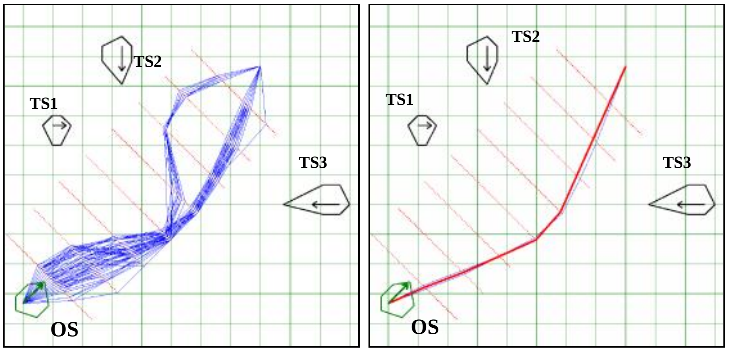


Figure 7 Positions of bacteria (Routes) in the first generation and after the evolving process for a traffic scenario

b) Scenario 2:

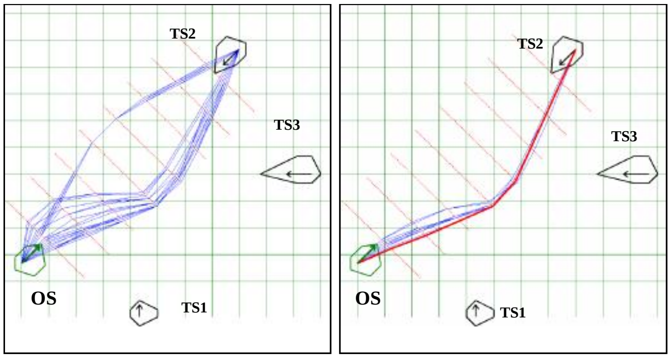


Figure 8 Positions of bacteria (Routes) in the first generation and after the evolving process for a traffic scenario involving a head on vessel

The scenario is another encounter case which involves a head-on TS and 2 others crossing from the starboard-side. In this scenario also, the traffic rules have been properly satisfied and the generated route is appropriate. The time required for calculation in both case 3 and 4 is less than 15 [sec]. The algorithm is therefore applicable in real time.

4.2 Weather Route Generation

In this scenario, weather forecast of up to 72 hours is used to compute the optimal weather route for our model ship from the departure point (20.2N,112.6E - upper end) to the destination (10.6N,110.9E – lower end). Weather data is extracted from the forecast database maintained by the Research Institute for Sustainable Humanoshere (RISH), Kyoto University, Japan. The forecast was produced on August 08th 2012 for 6-hour intervals. Routes produced by the bacteria population after the first cycle is shown in Fig. 9 (left). It is obvious that the population is distributed over the grid and therefore the whole grid has been searched.

After 3 cycles (Fig.9 – Right), all the bacteria further converged in a narrow region which is in fact the region around the optimal route. Due to the effects of the sea state, the optimal route deviates largely from the shortest route between the 2 positions (given the shore contours).

5. Conclusion

In this paper, an Adaptive-BFOA has been proposed and applied for optimal weather-route and collision-avoiding route generation respectively. Using Adaptive-BFOA, the solution is searched on a grid constructed from environmental constraints and other parameters decided by the ship officer.

To improve the efficiency of the searching algorithm, several modifications have been applied to classical algorithms suggested by other authors working on BFOA, taking into consideration the nature of marine traffic:

- A move-length adapting algorithm is suggested to improve the convergence speed and capacity of the bacteria to jump out of attractive region of local optimum.

- Bacteria are not replaced (die) before a certain number of chemotatic moves have been tried for them. The algorithm therefore does not miss a promising region.

The proposed Adaptive-BFOA is efficient for route-producing purpose. By the use of the route cost definition, the rules of the road can be properly taken into account. The solution is available within an acceptable time limit. Then, it is possible to apply in real time. Simulations have also shown that the Adaptive-BFOA is very robust and reliable, given the diversity of marine traffic environment.

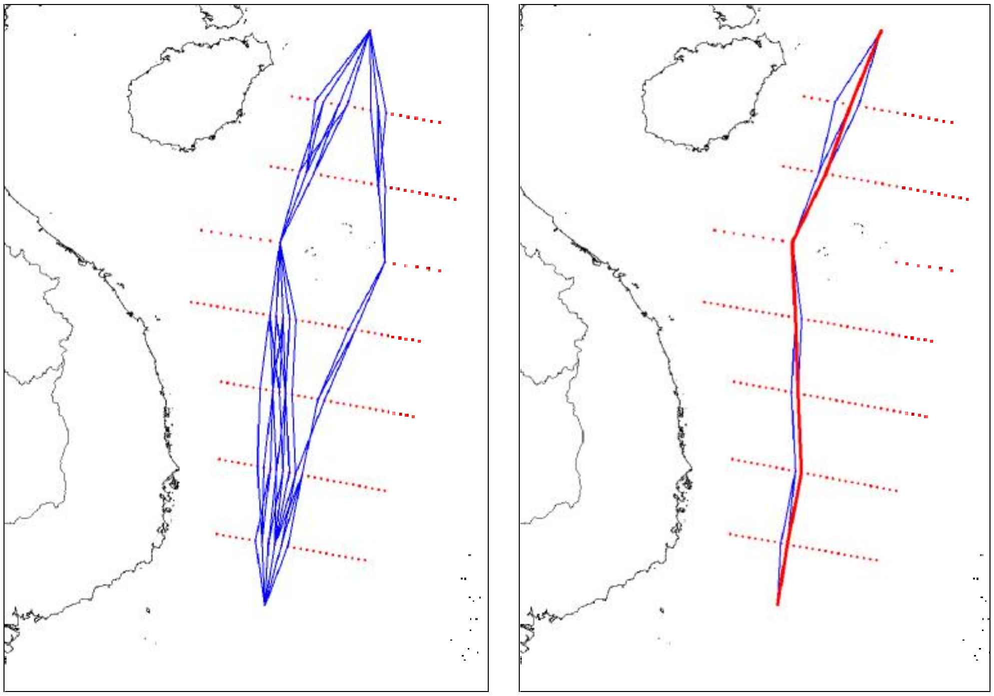


Figure 9 Weather Route Sample

6. References

[1] Hagiwara, H., *"Weather routing of (sail-assisted) motor vessels"*, PhD thesis, 1989, Delft

[2] N. Ward, S. Leighton, *"Collision Avoidance in the e-Navigation Environment"*, http://www.gla-rrnav.org/pdfs/ca_in_enav_ paper_iala_2010.pdf

[3] W. Lang, *"Ship collision avoidance route planning tool"*, http://www.bairdmaritime.com/index.php?option=com_content&view=article&id=5288:ship-collision-avoidance-route-planning-tool-&catid=78&Itemid=75

[4] R. Smierzchalski, *"Evolutionary algorithm in problem of avoidance collision at sea"*, Proceedings of the 9th International Conference, Advanced Computer Systems, 2002

[5] I. Fossen, *"Marine Control Systems"*, Marine Cybernetics, ISBN 82-92356-00-2, 2002

[6] M. D. Nguyen, *"A study on an integrated observing and collision avoiding support system for merchant ships"*, Doctor Thesis, Tokyo University of Marine Science and Technology, 2012

[7] K. M. Passino, *"Biomimicry of bacterial foraging for distributed optimization and control"*, IEEE Control Systems Magazine, Vol. 22, pp.52–67, 2002.