POAC'13
Espoo, Finland

Proceedings of the 22nd International Conference on
Port and Ocean Engineering under Arctic Conditions
June 9-13, 2013
Espoo, Finland

Application of Genetic Algorithm to Ship Route Optimization in Ice Navigation

Minjoo Choi[1], Hyun Chung[1], Hajime Yamaguchi[2], Liyanarachchi Waruna Arampath De Silva[2]

[1]Korea Advanced Institute of Science and Technology, Daejeon, KOREA

[2]The University of Tokyo, Kashiwa, JAPAN

Abstract

Global warming by excessive carbon emission has made the earth's temperature increasing. It has led to worldwide climate changes and even unexpected natural disasters. On the contrary to those troubles, shipping and oil companies take the unfavorable conditions into new business chances to extend their business range because the rising temperature opens new money-making opportunities around the arctic area. This study's purpose is to develop an effective and reliable ice navigation algorithm for arctic sea route. We employ genetic algorithm to find an optimal navigation path in the ice infested environment. The algorithm was tested on an arctic area map constructed from the ice prediction model, which consists of longitude, latitude, ice thickness, ice concentration, and terrain elevation data. The simulation result provides an economical and safe route within a reasonable computation time.

1. Introduction

Since the global temperature started to rise by dramatic greenhouse gases' increase, the Arctic summer sea ice has been reduced very rapidly. In general, global warming is an unfavorable symptom which induces climate changes and following natural disasters. On the other hand, decrease in summer sea ice is opening a new shipping path crossing the arctic sea area. Compared with the existing shipping route between the Far East and Europe, the sea routes through the arctic area let the shipping distance be enormously less, resulting in the reduction of CO_2 emission as well. In this situation, the ice navigation system which guides ships to more economical and safer paths is considered as one of the most important systems to take this opportunity.

There were many researchers studying efficient route-searching problems for ships [1-4]. Their main focus was to find an economical path avoiding collisions on the sea surface. However, the research, considering ice-covered environment, has just started and only several papers have been published. I. H. Park designed the ice navigation algorithm for the Northern Sea Route shipping, using the Dijkstra's algorithm which is one of the graph search algorithms [5]. However, his model did not consider the ice model and the algorithm only searched the optimal path within limited nodes, using the historical data of them. In [6], the authors did the practical research which was tested at Baltic sea area. In their research, they integrated three models (ship transit model, ice model, and optimization model) to construct an ice navigation system. When they employed the optimization method, three methods were considered for the route optimization and the Powell's method was used because of its fast computation time. However, it has a critical problem which is that the Powell's method does not allow to escape from a local minima whereas its searching time is short.

To tackle the problems, we designed a suitable genetic algorithm (GA) for the ice navigation system and verified its performance. Usually, the Genetic Algorithm is a well known optimization approach as a powerful tool since it can be applied to diverse science and technology fields and it provides a reasonable result while it is slow to converge to an optimal solution. The GA, we suggested for the ice navigation system, has two main advantages: 1) to overcome the limitations the graph search algorithms have, 2) to have high possibility of searching an optimal solution. In addition, the algorithm's slow converge speed can be improved with special operators such as 'Deletion' and 'Repair' operators.

2. Modeling

For the path planning, the map structure and variables should be defined. In our work, two types map structures, 1) discrete 2) continuous, are employed. The discrete type only considers integer points as candidate nodes. It is the raw data from the ice prediction model which generates a grid type map where each cell contains ice information and geographical information. Second, continuous type is generated by interpolating the discrete type. This type of structure makes it possible to find more delicate paths than discrete type, since there are more possible candidate nodes in every cell. The maps have two types of obstacles, 1) static 2) dynamic. Whether a point is feasible or not (intersected by obstacles) is determined based on the information which the point includes. The goal of our work is to find an economical path which is constructed with sequential nodes where each node includes x, y coordinates, velocity and feasibility information [10]. Here, the economical path means that total distance, smoothness of the path, and traveling time are optimized and balanced. Of course, the path should be safe. The factors, described above, are variables in our model and they are named $\mathrm{T_{distance}}$, $\mathrm{T_{smooth}}$, $\mathrm{T_{time}}$, and $\mathrm{T_{clearance}}$. More detailed descriptions of our model will follow.

2.1 Objective function

An objective function is very significant to an optimization problem because it guides the direction of the algorithm. In our problem, the objective function considered four factors of a path. The first factor is $\mathrm{T_{distance}}$. Here, the $\mathrm{T_{distance}}$ means a total distance of all linkages in a path. The second factor is $\mathrm{T_{smooth}}$. It indicates a mean angle in the path. The third factor is $\mathrm{T_{time}}$ which is a total required time to travel from the first node to the last node. And the final factor is clearance. It reflects safety of the path. When a path is passing infeasible areas, the evaluation function of the chromosome would impose a penalty value on the $\mathrm{T_{clearance}}$ value.

Min

$$Z = w_1 \times T_{distance} + w_2 \times T_{smooth} + w_3 \times T_{time} + w_4 \times T_{clearance} \quad (1)$$

Where

$T_{distance}$: Summation of distances between each node.
T_{smooth} : Average of angle among three sequential nodes
T_{time} : Summation of travel time between each node.
$T_{clearance}$: Checking observance of safety distances from obstacles.

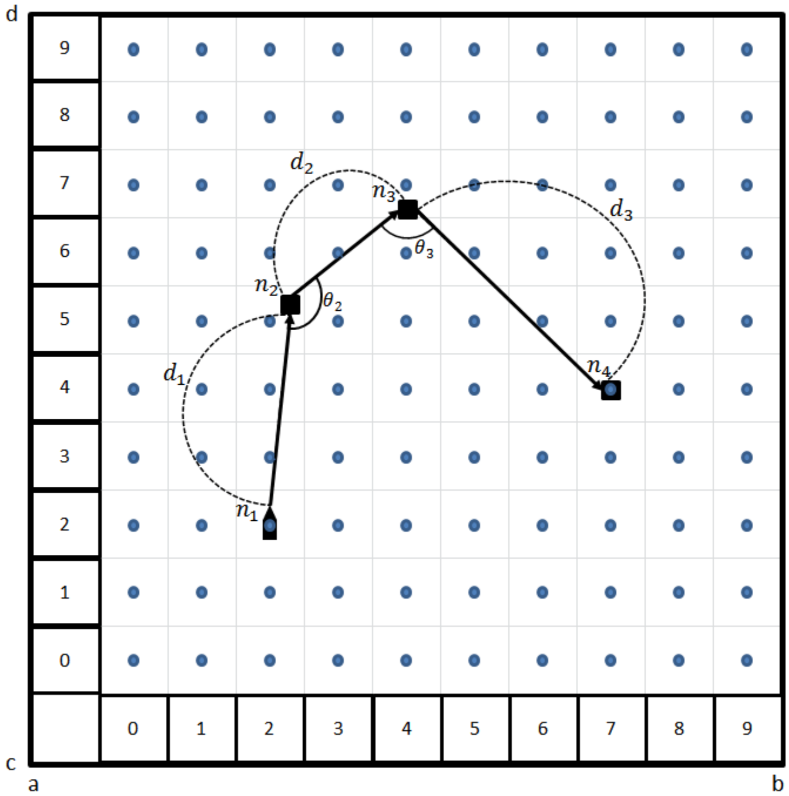

Fig. 1. An example path on the map.

Fig. 1 shows an example path on the map the algorithm was applied for. Each cell contains ice information (thickness and concentration) and geographical information (altitude, longitude, latitude). The information in the cell represents the environmental characteristics of the cell. In our work, we considered the grid map is continuous. Usually, an ice model generates the grid type of the map which is likely to consist of huge size of cells. In our work, the resolution was 20 km $\times$ 20 km. When it comes to the size of the cell, only considering integer points as the candidate nodes has a limitation to generate detail routes. For this reason, we modeled the environment to be continuous by interpolation and applied the suggested GA in the environment.

Fig. 2 is an example to explain the environment. If there is a node n_k, it would be surrounded by four integer points and the points would have the information which represents a cell. In this situation, the environmental characteristic of the node n_k is affected by the characteristics of the four points. When it comes to influence, we employed the invert distance weighting method which means the longer distance is the lesser influence to the node n_k. For example, the node n_k in Fig. 2 is affected by four surrounding nodes, $(i, j), (i + 1, j), (i, j + 1)$, and $(i + 1, j + 1)$. Among those points, the node n_k is closest to

$(\mathrm{i}+1, \mathrm{j})$ so that the environmental property of n_k is mainly subject to the point $(\mathrm{i}+1, \mathrm{j})$.

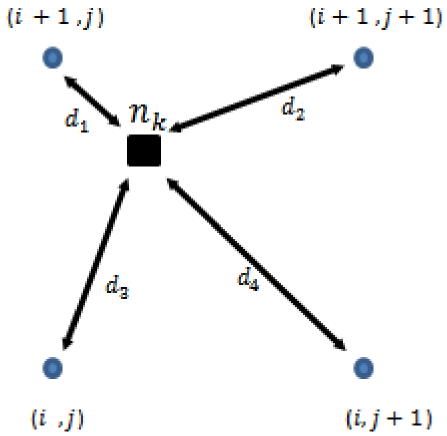

Fig. 2. An example node to describe interpolation

The first factor, $\mathrm{T}_{\mathrm{distance}}$, is $\sum_{i=1}^{n-1} d_i$ in the Fig. 1, where n is the number of node in a path. For the calculation of the distance between two nodes, the latitude and longitude information is used under the assumption the earth is a perfect spherical shape. The second factor, $\mathrm{T}_{\mathrm{smooth}}$, is the mean angle so that it is $\frac{\sum_{i=2}^{i=n-1} \theta_i}{n-2}$. This value describes the smoothness of the path. And, the third factor, $\mathrm{T}_{\mathrm{time}}$, is the travel time from n_1 to n_k which depends on two factors: the velocity and the distance between the nodes. In our work, we assumed that the velocity is in inverse proportion to the ice effective thickness of the path so that the velocity of an arc is determined between open water speed and the speed in the maximum ice breaking environment of the ship. The final factor, $\mathrm{T}_{\mathrm{clearance}}$, is to check the feasibility of a path. Here, the feasible path means it does not go past dangerous areas where the ice thickness is over capability of ice breaking or the water depth is shallow.

2.2 Constraints

Two main constraints are considered in this problem. The first constraint is that all nodes should be in the boundary of the map. Therefore, all coordinates of newly generated nodes or of the modified nodes should be within the range of the map boundary. Second, the ship should not sail on infeasible areas (obstructs) where the obstacles are classified as two types 1) static obstructs, 2) dynamic obstacles. Usually, the water level conditions are relevant in the static obstacles and the ice conditions are relevant in the dynamic obstacles. For example, the infeasible area occurs when an ice thickness value of a node, composing a path, is over the ship's icebreaking capability or the water depth is lower than a certain level. In this paper, a water depth below 10 m was chosen as the water depth constraint.

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

$$\mathrm{SF(t)} = \mathrm{E} - U_{j=1}^{k} O_{stat_j} - U_{j=k+1}^{l} O_{dyn_j}(\mathrm{t}) \tag{3}$$

where

E is the boundary of the environment.
SF(t) is the safety area (anti-collision).
[a, b] is x boundary of map.

[c, d] is y boundary of map.
$\mathrm{O}_{\mathrm{stat}_j}(j = 1, \ldots, k)$ is on-land area.
$O_{dyn_j}(t)$ is the area where the ice effective thickness is over the ship's ice breaking capacity.

3. Methodology

3.1 Genetic algorithm
To search the economical path, this paper employs the genetic algorithm (GA) which is one of the most famous meta-heuristics based on a population method. A genetic algorithm is a heuristic solution-search and an optimization technique, originally motivated by the principle of evolution through (genetic) selection [7]. The GA makes it possible to search a global optimum solution escaping from local optima and it is a robust method which can be applied for a wide range of complex practical problems in science and engineering fields. However, it is debatable whether GA is suitable for path planning or not, since GA is time-consuming and in some case, it is terminated before solutions reaches an optimal solution. To tackle the problem, researchers have introduced modified GA. J. Grefenstette employed Random Immigrant (RI) which changes environment to prevent rapid convergence of solutions by replacing a fixed percentage of population with newly generated chromosomes [8]. Ahmed, Hussenin, and Shawki devised a dynamic planner. The dynamic planner controls the ratio of RI and mutation to avoid early matured convergence when the solution improvement rate is slow [9]. Furthermore, they introduced repair and smooth operators which helps to improve GA application for path planning. Those operators increase convergence speed of GA. As mentioned above, a GA has both advantages and disadvantages. In this paper, we employed the GA for the path planning problem because of it's ability to follow the shortest path unlike grid boundary mapping constraints of Graph Search Algorithm s (GSA) such as A* algorithm and Dijkstra algorithm. For example, the number of available candidate node in a cell is different. In GSA, the number of node in a cell is fixed because possible nodes should be prefixed before the computation; on the other hand, there can be countless candidate nodes in the cell under the continuous environment. This is because the nodes in the path are generated while the computation is carried out. Fig. 3 describes an example case.

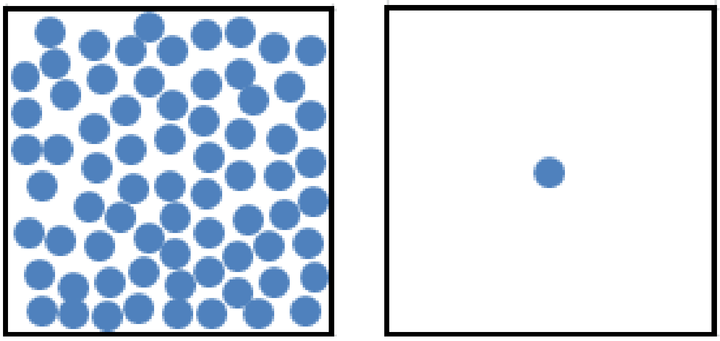

Fig. 3. The number of available candidate node in a cell

Because of the characteristic, the GSA and GA would have different feasible solution area which each algorithm can generates. Fig. 4, 5 describe example cases in each algorithm.

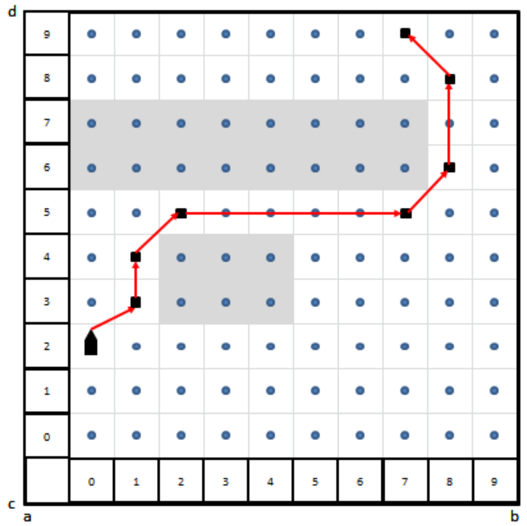

Fig. 4. Available path in GSA

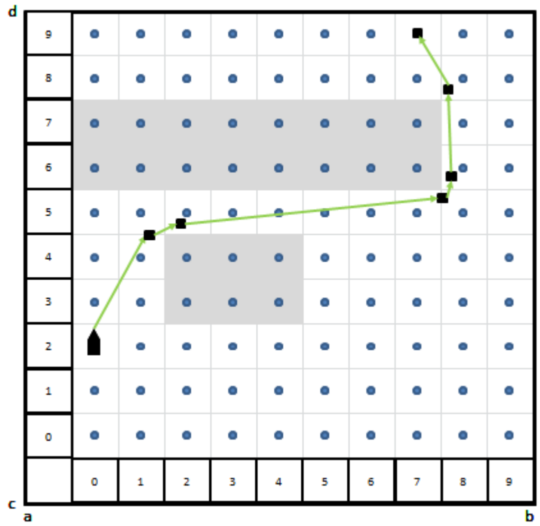

Fig. 5. Available path in GA

3.1.1 Chromosome structure

For a GA, how to represent a chromosome is the most important. In our work, a path is considered a chromosome and each gene, constructing a chromosome, consists of x, y coordinates, velocity and feasibility variables. This structure [Fig. 2] was mentioned by Roman Smierzchalski [10]. To generate an initial population, all values of a node, except for the first and last node, are given randomly within a feasible area. Also, the number of node in a path is decided by a random number; however, as a generation goes by, the size of a path can be increased or decreased by a fitness function.

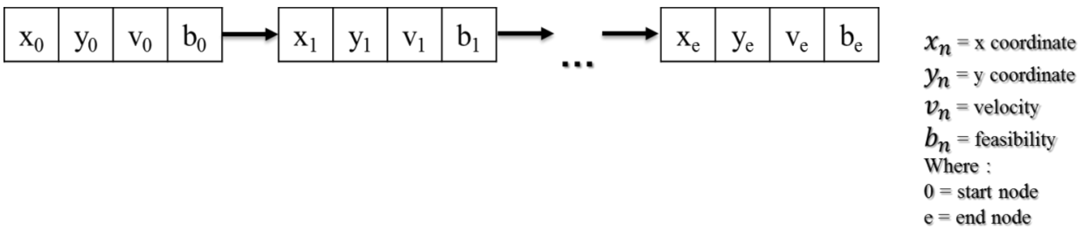

Fig. 6. Chromosome structure

3.1.2 Operators

In this paper, five genetic operators, which are crossover, mutation, reproduction, repair, and random immigrant (RI), were employed for the GA. The detail description about the operators would be explained below.

3.1.2.1 Crossover

Crossover operator is to generate child chromosomes by exchanging some parts of their parents chromosomes [11]. The main function of this operator is to generate qualified new chromosomes, believing qualified parents bear qualified their children. As the evaluation function filters poor quality parents, next population tends to consist of better chromosomes than previous populations. Various types of crossover operators have been proposed and implemented. In the typical ways, there are one-point crossover, two-point crossover, multi-point crossover, and uniform crossover. In this work, an one-point crossover was employed.

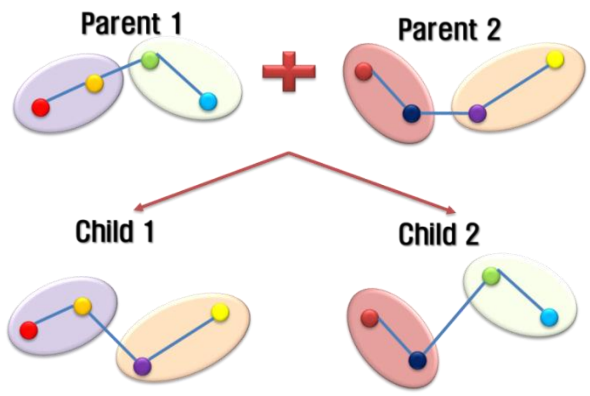

Fig. 7. One point crossover operator description [13]

3.1.2.2 Mutation

Mutation operator makes it possible to escape from a local optimality through a transition from a current solution to its neighborhood [11]. In our work, the operator was implemented by changing a node in a path into the other node not in the path.

Fig. 8. Mutation operator description [13]

3.1.2.3 Reproduction

Reproduction operator enables the most qualified chromosomes among the previous members to survive with pre-assigned probability. They are evaluated and sorted with newly created child chromosomes. If the ratio of reproduction operation is high, the global optima searching capability becomes low so that the ratio should be carefully determined by design of experiments.

3.1.2.4 Repair

Repair operator makes an infeasible connection between two nodes to be feasible connection. The operator replaces an infeasible node in the path with a new feasible node near the node.

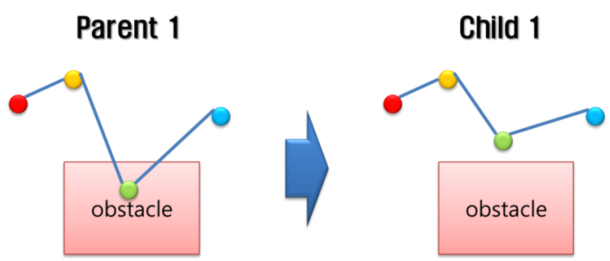

Fig. 9. Repair operator description [13]

3.1.2.5 Deletion

Deletion (smooth) operator improves the path by eliminating unnecessary nodes. For example, a middle node in three sequential nodes would be deleted if the quality of the path which only consists of first and third nodes is better than the quality of the original path which consists of first, second, third nodes. It makes the path smoother and simpler.

Fig. 10. Deletion operator description [13]

3.1.2.6 Random Immigrant

Random Immigrant (RI) is an operator which generates new chromosomes and replaces bad chromosomes in the population with the newly generated chromosomes [5]. This operator strengthens GA's global searching capability but weakens its convergence capability.

3.1.4 Dynamic planner

Dynamic planner controls the ratio of operators' performance probability. The proportion of RI and mutation among operations is changed depending on the convergence speed. For example, it decreases a crossover ratio and increases mutation and RI proportion to escape from the local area, when the convergence speed is slow.

3.2 Procedure

The suggested GA follows the procedure [Fig. 11]. The dynamic planner controls the selection ratio of mutation and RI, based on the improvement rate of the generations. Here, we considers RI as an operator of the GA, not the element of the dynamic planner.

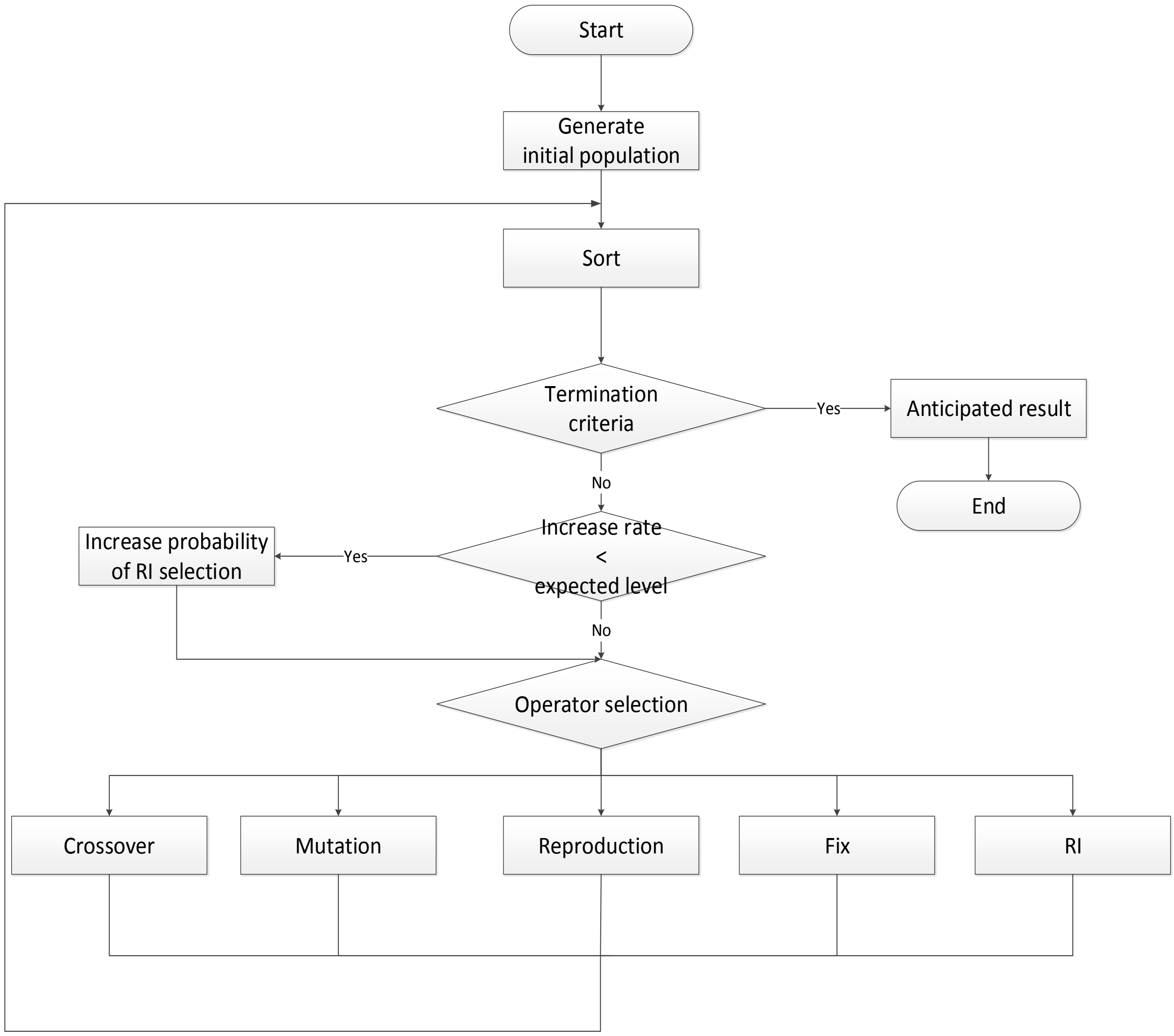

Fig. 11. Genetic algorithm procedure

3.3 Reference ship and Ship's speed in the ice-covered area

For the implementation, we employed a reference ship from The Northern Sea Route [12]. The ship is an icebreaking bulk/container ship. A detail specification of the ship is described below.

Table 1. Reference ship description

Feature / Ship type	Length(Lpp) x beam (B) x draft (m)	Cargo tonnage (metric tons)	Normal shaft Horsepower (MW)	Speed in open water (knots)	Icebreaking capability (m)	Route
40,000 Dwt Icebreaking bulk/container ship	186.1 x 27.5 x 12.5	36,000	28	14.5	1.85 m at 1.0 m/sec	Northerly route

From the Table 1, we could set the maximum speed and the minimum speed of the ship in the simulation. Within the available ice thickness range, the ship speed was considered as a linear relationship in the simulation. Fig. 12 describes the relationship between ice resistance and ice thickness [15]. In fact, it is not perfect linear relationship but we could get insight that the ice resistance is proportional to ice thickness from Fig. 12. Since only fixed ship's speed is considered, it is not needed to consider the relationship between ship's velocity and ice resistance. To take into account the ice concentration information, ice effective thickness information (ice thickness × ice concentration) was employed, instead of ice thickness. For example, if an ice effective thickness value in a grid is lower than 1.85 m, the speed of the reference ship was calculated by interpolation between 14.5 knot and 1.9438 knot (1.0 m/sec); otherwise, the grid was considered as an obstacle which imposed a penalty by the fitness function when the path was evaluated.

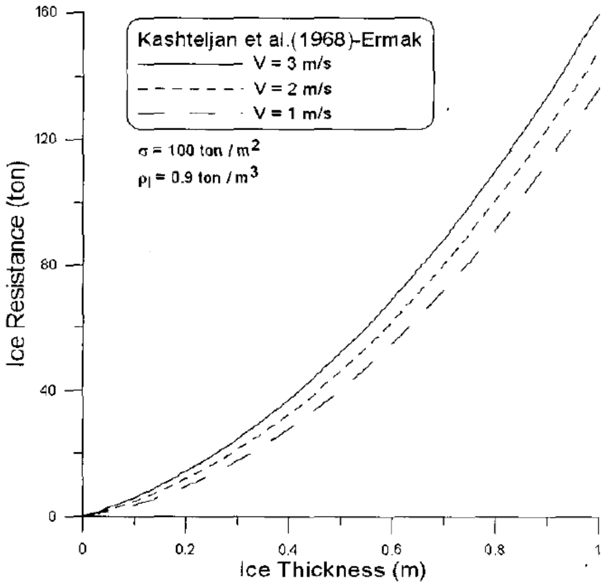

Fig. 12. The relationship between Ice resistance and Ice thickness with Kashteljan' eq [15]

3.4 Ice distribution map

We employed an Ice-POM numerical model to generate the ice distribution map for the simulation. The Ice-POM model is an ice ocean coupled model and the ocean part is based on Princeton Ocean model. Ice part is based on EVP (elastic viscous plastic rheology) and it takes account of floe collision and 0 layer thermodynamics model. This model was developed from Yamaguchi's laboratory in the University of Tokyo [14]. The map generated from the Ice-POM model is constructed based on grid cells. Fig. 13 is the 2003-Aug-01 snapshot from the model. In the map, the number of grid cell is 244 × 253, and the size of each grid is about 20 km × 20 km. Each cell contains ice information (thickness and concentration) and geographical information (altitude, longitude, latitude). In this problem, obstacles are subject to the level of ice effective thickness (ice thickness × ice concentration) and the level of altitude. To distinguish feasible region, we used color image scale. The ice effective thickness level is given by a color scale and it goes from $0_{\mathrm{m}}$to 6_{m} in steps on 0.5_{m}.

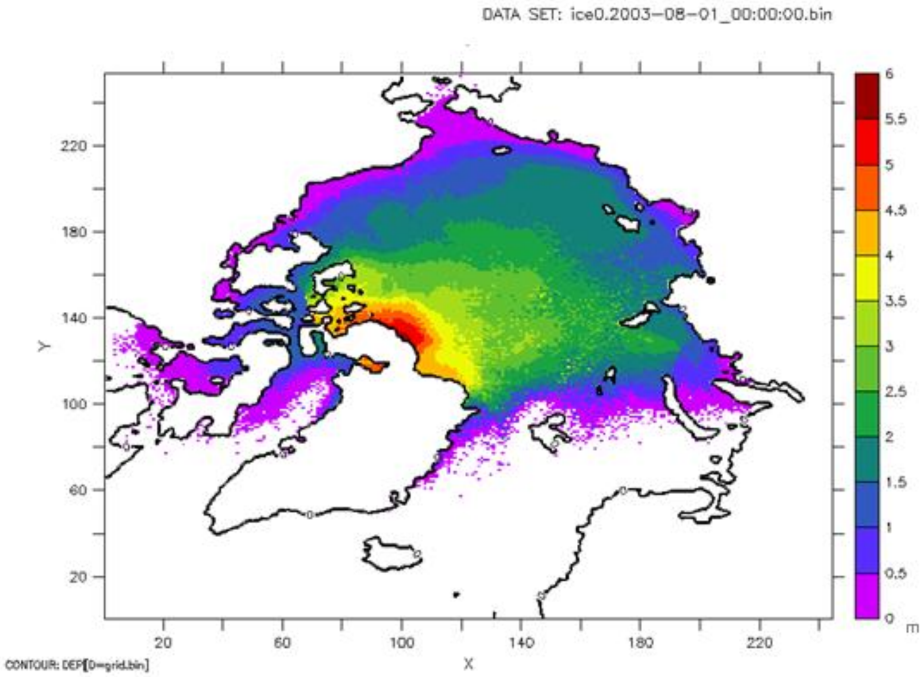

Fig. 13. The map generated by the ice model

: Ice thickness in shown in different colors, see the vertical bar

3.5 Parameter set

The performance of a GA significantly relies on how parameters are set. There are several parameters for the GA. First, population size is the number of chromosomes in a generation. As a population size increases, it increases the possibility of avoiding a local optima, but computation time also increases rapidly at the same time. In addition, the operators ratio has a significant effect on the result of implementation. Each operator has its own function. For example, Crossover makes the chromosomes converge and Mutation makes it possible to escape from a local optima. However, if the ratio of Crossover is too high, it is likely to converge on local optima; on the other hand, if the ratio of Mutation is too high, it is likely to diverge. For those reasons, to set a proper parameter ratio is important, and it should be decided carefully. We chose a parameter set through numerical trials.

Table 2. Parameter set for the experiments

Parameter set	
Population size	80
Termination criteria	No increase within 100 iterations or over 1000 iterations
Initial operators ratio	Crossover (50%) Mutation (37.5%) Deletion (6.25%) Repair (6.25%)

Result

We measured the algorithm performance in two cases 1) continuous environment with interpolation, 2) discrete environment without interpolation. Despite the concern that the convergence speed in continuous condition would be slower in comparison to discrete condition as it has more alternatives, the algorithms reached an optimal solution at the similar iteration [Fig. 14, 15]. In the figures, the objective value was the result from the equation (1) at every iteration. Here, the weight factors of the four components (distance, smooth, time, and clearance) in the equation have scaled values by numerous experiments to find more reasonable solution. From the experiments, we could get a weight factor set (0.8, 15, 10, and 2000). The objective function equation is described in the section 2.1.

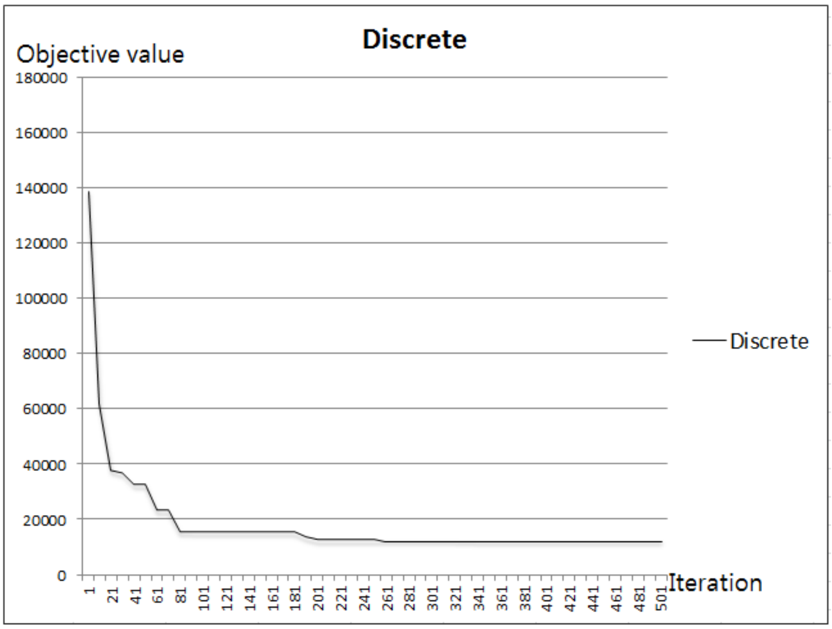

Fig. 14. Objective value decrease in the discrete case.

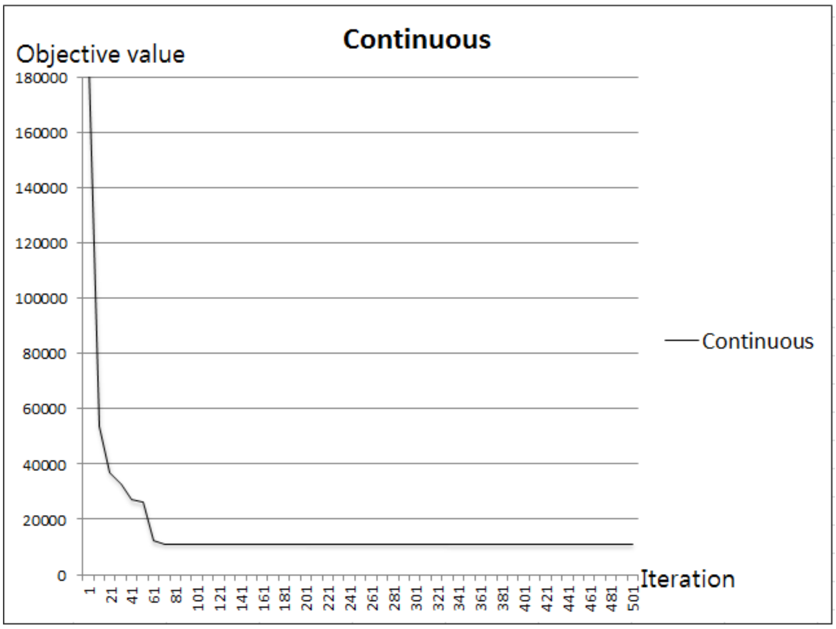

Fig. 15. Objective value decrease in the continuous case.

When it comes to the solution quality, the better solutions were found on the continuous condition [Fig. 16, 17]. Here, we set the start node (120, 222) and the end node (10, 60) because it is difficult to find feasible routes between the nodes. In each case, the detailed solution information is described in Table 3.

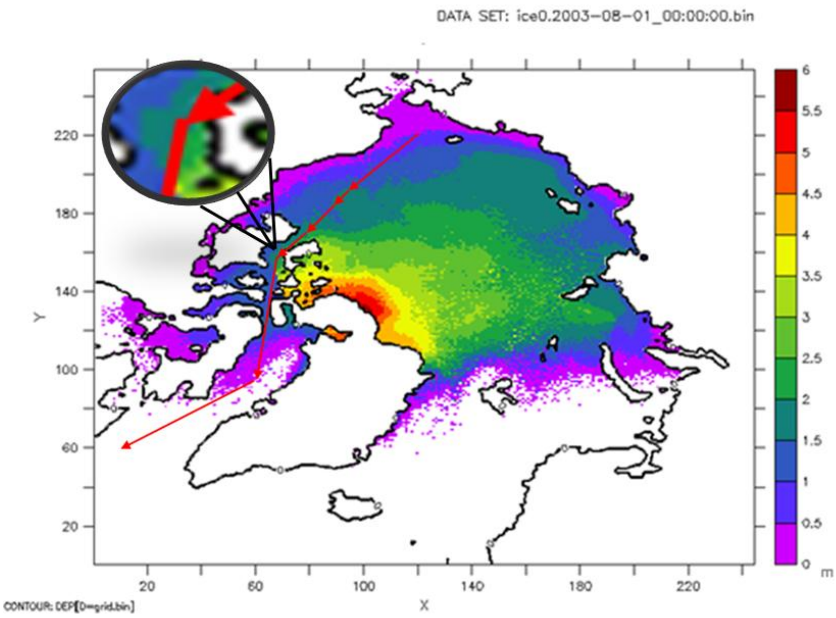

Fig. 16. The optimal path on the discrete environment

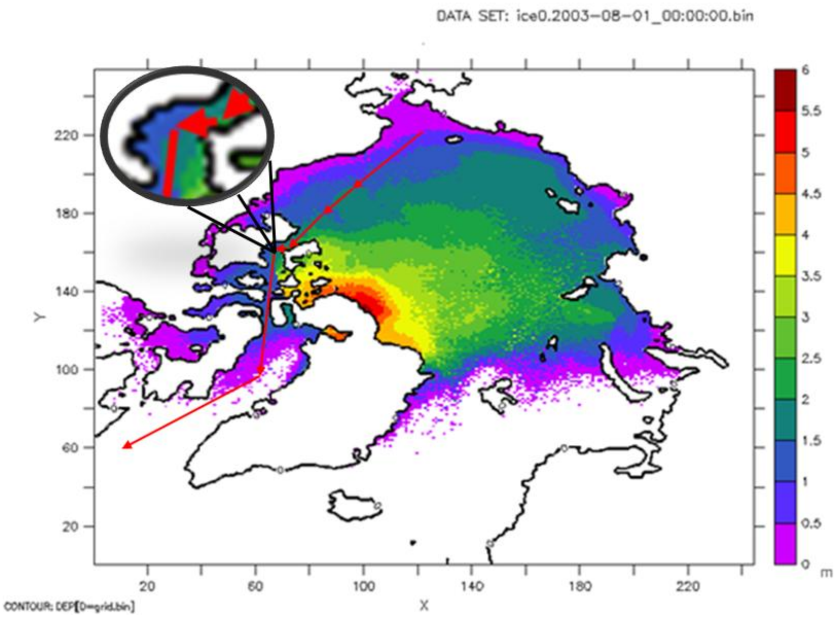

Fig. 17. The optimal path on the continuous environment

Table 3. The comparison table between discrete condition and continuous condition

	Discrete	Continuous
Optimal path	(120, 222) → (89, 194) → (87, 184) → (78, 171) → (67, 158) → (60, 96) → (10, 60)	(120, 222) → (94.3, 194.9) → (83.3, 181.1) → (70, 163.6) → (66.9, 161) → (60.8, 94.4) → (10, 60)
Total distance	9041.42 km	8428.87 km
Total time	314.095 hour	290.419 hour
Average angle	23.634 °	29.386 °
Total clearance	0	0

Conclusion

Path planning problem in an ice covered environment has been getting important at issue. To tackle the problem, the genetic algorithm was employed since it is a powerful to search a global area and suitable on the continuous map. Before the application, we modified the discrete map generated from the ice model into the continuous map by invert distance weighting method. In the continuous environment, the suggested algorithm generated reasonable solutions within a reasonable time. However, this study has several limitations. The map was static, not dynamic and stochastic. Thus, the next research will be carried out by the following two steps 1) the algorithm will perform on the dynamic environment. In the step, the sea ice behaves differently as time passed by so that the speed of the ship becomes an important factor, 2) next research will take uncertainty into account by introducing a stochastic model, the behavior of sea ice is not deterministic; therefore, the sea ice should be considered as an uncertain element. The stochastic model would be more practical and safer than the previous deterministic models. The other limitation is the GA does not ensure the optimality. To tackle this, we are planning to combine advantages of both a GA and a GSA by inserting some designed solutions in the initial population of the GA, such as the solutions from the GSA.

Acknowledgement
This research was partially supported by WCU (World Class University) program through the National Research Foundation of Korea funded by the Ministry of Education, Science and Technology (R31-2008-000-10045-0), and Human Resources Development of the Korea Institute of Energy Technology Evaluation and Planning (KETEP) grant funded by the Korea government Ministry of Knowledge Economy. (No. 20114030200040)

References

[1] V. I. Churkin and Y. I. Zhukov, Procedures for ship collision avoidance, in Proceedings of IEEE OCEANS Conference, Nice, France, September 1998, vol. 2, pp. 857-860
[2] M. Benjamin, J. Curcio, and J. Leonard, Navigation of Unmanned Marine Vehicles in Accordance with the Rules of the Road, in Proceedings of IEEE International Conference on Robotics and Automation(ICRA), Orlando, Florida, May 2006, pp. 3581-3587
[3] R. Szlapczynski, A New Method of Ship Routing on Raster Grids, with Turn Penalties and Collision Avoidance, The Journal of Navigation, vol. 59, pp. 27-42, 2006
[4] J. Witt and M. Dunbabin, Go with the Flow : Optimal AUV Path Planning in Coastal Environments, in Proceedings of the Australasian Conference on Robotics and Automation, Canberra, Australia, December 2008, pp. 1-9.
[5] I. H. Park, J. H. Nam, and K. S. Choi, A graphical approach to determine the optimal sea route of icebreakers in arctic region. The Korean Association of ocean Science and Technology Societies joint conference, Busan, June, 2011
[6] Ville Kotovirta, Risto Jalonen, Lars Axell, Kaj Riska, and Robin Berglund, 2008. A system for route optimization in ice-covered waters. Cold Regions Science and Technology 55 (2009) 52-62
[7] D. E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison Welsey, 1989 (GA definition)
[8] J. Grefenstette, Genetic algorithms for changing environments, In Proceedings of Parallel Problem Solving from Nature (PPSN-2), pp. 137-144, 1992
[9] Ahmed Elshamli, Hussenin A. Abdullah, and Shawki Areibi, Genetic Algorithm for Dynamic Path Planning, CCECE/CCGEI 2004, Niagara Falls, May 2004
[10] Roman Smierzchalski and Zbigniew Michalewicz, Modeling of ship trajectory in collision situations by an evolutionary algorithm, IEEE transactions on evolutionary computation, vol. 4, no. 3, September 2000
[11] J. Y. Choi and S.-S. Ko., Simulation-based two-phase genetic algorithm for the capacitated re-entrant line scheduling problem, Computers & Industrial Engineering, vol. 57, pp. 660-666, 2009
[12] Hiromitsu Kitagawa, Nobuo Ono, Hajime Yamaguchi, Koh Izumiyama, and Kazuhiko Kamesaki, The Northern Sea Route, SHIP & OCEAN FOUNDATION, 15-16, 2001
[13] Yangrong Hu and Simon X. Yang, A Knowledge Based Genetic Algorithm for Path Planning of a Mobile Robot, Proceedings of the 2004 IEEE international Conference on Robotics & Automation New Orleans, LA, April 2004
[14] L.W.A. De Silva, Hajime Yamaguchi and Jun Ono, 2013, Preliminary results of sea ice prediction in northern sea route using high-resolution ice ocean coupled model, Proceedings of the 28th international symposium on Okhotsk Sea and Sea Ice, Mombetsu, Japan, 127-130
[15] Kashteljan, V.I., Poznjak, I.I. and Ryvlin, A.J., Ice Resistance to Motion of a ship, Vol 1, sudostroyenie, Leningrad (Translation from Russian), 1968