Contents lists available at ScienceDirect

International Journal of Naval Architecture and Ocean Engineering

journal homepage: http://www.journals.elsevier.com/international-journal-of-naval-architecture-and-ocean-engineering/

Mission planning and performance verification of an unmanned surface vehicle using a genetic algorithm

Jihoon Park [a], Sukkeun Kim [a], Geemoon Noh [a], Hyeongmin Kim [a], Daewoo Lee [a, *], Inwon Lee [b]

[a] Department of Aerospace Engineering, Pusan National University, Busan, South Korea
[b] Department of Naval Architecture & Ocean Engineering, Pusan National University, Busan, South Korea

A R T I C L E I N F O

Article history:
Received 2 July 2020
Received in revised form
1 June 2021
Accepted 16 July 2021
Available online 19 July 2021

Keywords:
Unmanned surface vehicle
Vehicle routing problem
Genetic algorithm
Mission planning system
HILS

A B S T R A C T

This study contains the process of developing a Mission Planning System (MPS) of an USV that can be applied in real situations and verifying them through HILS. In this study, we set the scenario of a single USV with limited operating time. Since the USV may not perform some missions due to the limited operating time, an objective function was defined to maximize the Mission Achievement Rate (MAR). We used a genetic algorithm to solve the problem model, and proposed a method using a 3-D population. The simulation showed that the probability of deriving the global optimal solution of the mission planning algorithm was 96.6% and the computation time was 1.6 s. Furthermore, USV showed it performs the mission according to the results of the MPS. We expect that the MPS developed in this study can be applied to the real environment where USV performs missions with limited time conditions.

1. Introduction

This study contains the development of a Mission Planning System (MPS) for securing the autonomy of Unmanned Surface Vehicles (USVs) and verification of application to real mission situations through HILS.

According to the development of advanced science technology and along with the changes of the increasingly complicated maritime environment, research on USV has been actively conducted. After the September 11 attacks, the US Navy established a new marine strategy called Sea Power 21, which uses innovative information systems and networked advanced weapons systems to effectively prepare for asymmetrical threats at sea. Furthermore, in other countries, demand for USVs has been increased not only in the military sector but also in the private sector, including marine surveys, marine surveillance, search and rescue for distressed people, and assistance in marine pollution control (Park et al., 2017).

Research on MPS is required in order to secure the autonomy of the USV. The autonomous MPS has been studied not only in the marine field but also in the fields of ground-based robotics and aerospace domains. Nevertheless, a study of MPS for USVs was less focused and less invested than UAVs, UUVs, and UGVs (Zanoli et al., 2012). To address the lack of study of MPS for USVs, we developed a MPS that can be connected to USV's guidance and control system in this study. The MPS in this study includes hardware systems, and mission planning problem models and solvers as well.

This study is organized as follows. The optimization problem model and assumptions are described in Section 2. The solver of the mission planning algorithms based on genetic algorithms is explained in Section 3. In this section, we proposed the method of applying a 3-D population to a genetic algorithm for the MPS. The configuration of the simulation system is covered in Section 4, which contains the interaction of the MPS and USV systems, and includes building a simulation environment. The simulation for verification of the proposed mission planning algorithm and developed MPS is described in Section 5. Finally, the conclusion is contained in Section 6.

2. Optimization problem definition

2.1. Problem definition and assumption

In this study, the situation of mission planning optimization for

* Corresponding author.
E-mail address: baenggi@pusan.ac.kr (D. Lee).
Peer review under responsibility of The Society of Naval Architects of Korea.

https://doi.org/10.1016/j.ijnaoe.2021.07.002

a single USV agent with given multiple missions in the activity area was assumed to be solved. In a real mission environment, the USV has a limited operating time. Several factors affect to the operating time limit such as endurance of USV, customer's demand, urgent situation, etc.

The optimization problem for multiple missions of a vehicle can be expressed as a Vehicle Routing Problem (VRP) model (Pisinger and Ropke, 2007). For example, the Traveling Salesman Problem (TSP) model has been mainly used for single vehicle operating situations as described in this study (Little et al., 1963). The TSP is a general optimization model for multi-mission sequencing (Grefenstette et al., 1985). The main idea of the TSP is to minimize the resource consumption to visit all given depot nodes (mission points). However, the TSP model has a constraint: all depot nodes must be visited. This constraint is in conflict with the scenario assumed in this study. Due to the limited operating time, the USV may encounter situations that some missions that must be abandoned. If there is no constraint about visiting all mission points in the TSP, a solution is derived that the USV will not perform any mission for cost minimization.

Therefore, in this study, operating time was regarded as a major constraint. The problem was formulated to maximize the Mission Achievement Rate (MAR) by performing as many missions as possible within a given operating time. Prior to the formulation of the problem, several situations were assumed. The distance of the path between each mission point was given in the matrix form at the beginning of the operation. The distance matrix was derived through the path planning algorithm such as A* and RRT before the missions planning algorithm (Noreen et al., 2016). In this study, since it is assumed that there are no obstacles at sea level, the Euclidean distance between each mission point is derived. The USV was assumed to be sail with constant velocity, and the density of the seawater in the mission area was assumed to be uniform. Thus, it can be considered that the fuel consumption rate of USV is constant. Through this, problem modeling was performed by substituting all costs with time. In this study, we have focused on the verification of the mission planning system by simplifying the model as described above, and a future study will deal with a problem model closer to reality.

2.2. Objective function

$$minimize\ J = -MAR = -\frac{\sum_{i=1}^{n}\sum_{j=1}^{n} p_j x_{ij}}{\sum_{j=1}^{n} p_j} \times 100 \tag{1}$$

In this study, we set up an environment including given n-mission points, starting point, and arrival point. The objective function, MAR, consists of the product of the reward variable p_j and the binary variable x_{ij}, as shown in Eq. (1).

The subscript i and j are indices of mission points. p_j is a reward variable that is given when performing a mission at a mission point j. The reward variable of each mission is defined in advance. x_{ij} is a binary variable that determines whether a path connecting the previous mission point i to the current mission point j was passed or not. If the USV does not cross this path, mission j is regarded as not performed. The objective function is the MAR, which is expressed as the ratio of the sum of the reward variables of the mission points performed and the sum of the reward variables of all the mission points.

2.3. Constraints

$$\sum_{i=1}^{n}\sum_{j=1}^{n}(t_{move,ij} + t_{perform,j}) \cdot x_{ij} \le T_{limit} \tag{2}$$

Eq. (2) is a constraint representing the relationship between the T_{limit} and the time consumption to perform the missions where T_{limit} indicates the operating time given to the USV. T_{limit} is given as an initial condition at the start of the mission planning. The time consumption to perform the missions must not exceed the T_{limit} as shown in Eq. (2). The time consumption to perform the missions is the sum of $t_{move,ij}$ and $t_{perform,j}$. $t_{move,ij}$ is the time spent to move from mission point i to j which is related to the speed of the USV. However, since the speed and fuel consumption rate was assumed to be constant in this study, $t_{move,ij}$ can be considered to be directly proportional to the distance between mission points. $t_{perform,j}$ is the time spent to perform the mission at mission point j. Owing to unexpected factors that may occur during the mission, the $t_{perform,j}$ at each mission point cannot be fixed as constant. Therefore, in this study, the initial value of each mission point is assigned and random errors in a certain range is applied to the initial value for each simulation to consider the unexpected factors. However, in the simulation of this study, the error is set to 0 to maintain the same condition for each case.

$$\sum_{j=1}^{n} x_{0j} = 1 \tag{3}$$

$$\sum_{j=1}^{n} x_{j0} = 1 \tag{4}$$

$$\sum_{j=1}^{n} x_{ij} \le 1 \tag{5}$$

$$\sum_{j=1,\ j\ne i}^{n} (x_{ij} - x_{ji}) = 0 \tag{6}$$

$$t_{start,j} \ge t_{start,i} + t_{perform,i} + t_{move,ij} - T_{limit}(1 - x_{ij}) \tag{7}$$

In this study, the starting point and arrival points were regarded as the same place, which is called the base point. At the start of the mission, the USV must depart at the base point and return to the base point for the next voyage after the mission is over. These conditions are expressed as Eqs. (3) and (4). Eq. (5) is the constraint that describes that one mission point must be connected to the other. Eq. (6) is a constraint that prevents the selection of the past path again. Through Eqs. (3)–(6), the solution is guaranteed one circulation path. The constraint for avoiding sub-tours is expressed by Eq. (7). This equation is an expression for $t_{start,j}$, which means the mission starting time at point j (Miller et al., 1960).

3. Solver design by applying genetic algorithm

The genetic algorithm is one of the meta-heuristic optimization methods. It was proposed by Holland in 1975 and focused on the evolution of living organisms (Holland, 1992). Living organisms can survive by adapting to a given environment, with genetic diversity through crossbreeding (crossover) or mutations. A genetic algorithm is advantageous for modifying an objective function and adding constraints, which makes the application more flexible. As shown in Fig. 1, a genetic algorithm represents possible solutions in the form of chromosomes. When a solution is viewed as a set, each element constituting the solution can be expressed as a gene, and the set of possible solutions is called a population. The progress of the genetic algorithm is shown in Fig. 2. After generating an initial population, genetic operators such as selection, crossover, and mutation are applied to each gene included in the population. And then, the algorithm evaluates the fitness (the value of the objective function) in the population and explores the best solution.

3.1. Encoding and decoding methods

3.1.1. Expression of solution

In this study, for the problem encoding, each solution was expressed as a permutation vector. The permutation means the sequence of missions, and each element in the permutation represents a mission point. Fig. 3 gives an example of the encoding. The permutation length is equal to the number of initially given mission points. Among them, mission points abandoned by the USV were marked as 0 within the permutation.

3.1.2. Creating initial population for GA application

To apply the genetic operator, an initial population must be created. In a general mission sequencing optimization problem, the population is expressed as a 2-D matrix, a set of permutation vectors. However, in this study, we propose a 3-D matrix representation method to encode Eqs. (1) and (2). The population is expressed as a $m \times n*2$ size matrix as shown in Fig. 4, and the matrix A corresponding to $\{m,n,1\}$ is a set of m-permutation vectors of length n representing the mission sequence. Here, n indicates the number of mission points given and m indicates the size of the population, which is a configurable parameter related to the convergence performance of the genetic algorithm. Matrix B corresponding to $\{m,n,2\}$ is a binary matrix that determines whether to perform the mission or not. Finally, missions sequence matrix S is determined through the Hadamard product of A and B as the process is shown in Fig. 5.

The initial population was created as follows. Generate a random permutation vector consisting of natural numbers from 1 to n. This is repeatedly generated m-times to matrix form A. In the case of matrix B, each element was created by generating 0 or 1 with a 50% probability.

3.1.3. GA operators for selection, crossover, and mutation

In this study, Tournament Selection, Partially Mapped Crossover (PMX), and Swap Mutation were used for genetic operators. When the solution is expressed as a permutation vector, repeated genes occur in the crossover phase. To solve this problem, after the crossover section exchange, the repeated genes are modified as shown in Fig. 6 (Goldberg and Lingle, 1985).

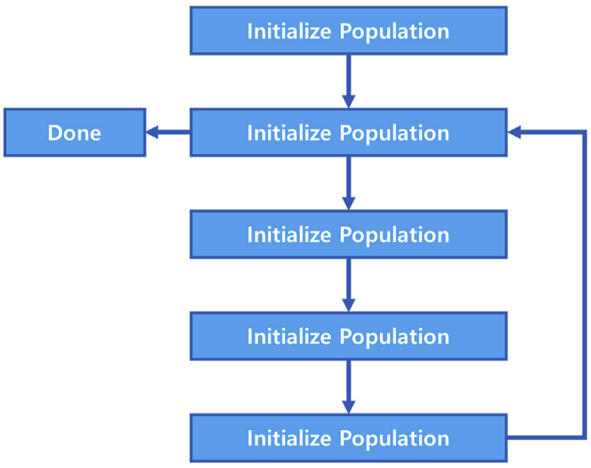

Fig. 2. Sequence of GA.

3.1.4. Handling constraints

In this study, a penalty variable, γ, is assigned to the infeasible solution to obtain a solution that satisfies the constraints. To implement this, the objective function, defined in Section 2 as Eq. (1) can be derived into Eqs. 8–10.

$$minimize\ J = -MAR = -\frac{\sum_{i=1}^{n} p_i x_i}{\sum_{i=1}^{n} p_i} \times 100 - \gamma T_{total,i} - T_{limit} \tag{8}$$

$$T_{total,i} = T_{move,i} + T_{perform,i} \tag{9}$$

$$T_{total,i} - T_{limit} = \begin{cases} T_{total,i} - T_{limit} & \ldots\ldots(T_{total,i} > T_{limit}) \\ 0 & \ldots\ldots(T_{total,i} \le T_{limit}) \end{cases} \tag{10}$$

γ is a constant variable that can be arbitrarily specified by the user. If the total mission execution time, $T_{total,i}$, exceeds the time limit T_{limit}, it is proportionally added to the objective function. As a result, the probability of survival of the solutions given a penalty variable is reduced. An example was shown in Fig. 7.

4. Hardware–in–loop system configuration

The HILS environment was constructed to verify the mission planning optimization algorithm proposed in this study. The entire system consists of three sub-systems: MPS (Mission Planning System), AGCS (autonomous guidance and control system), and a simulation environment. The interaction between each sub-system

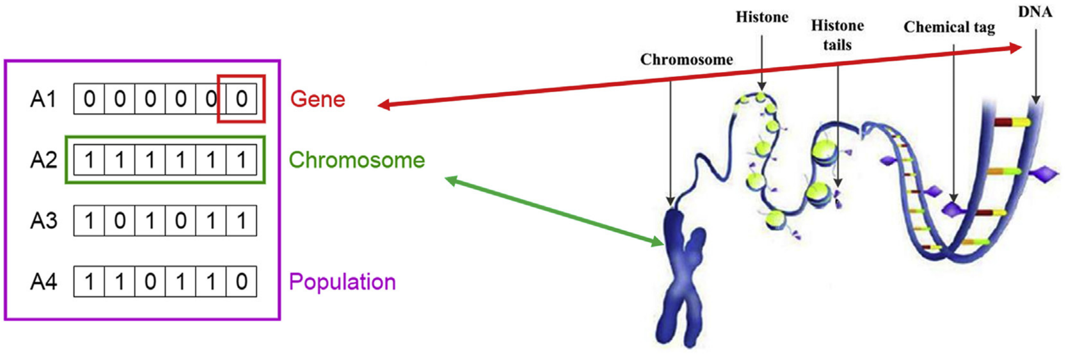

Fig. 1. Concept of GA.

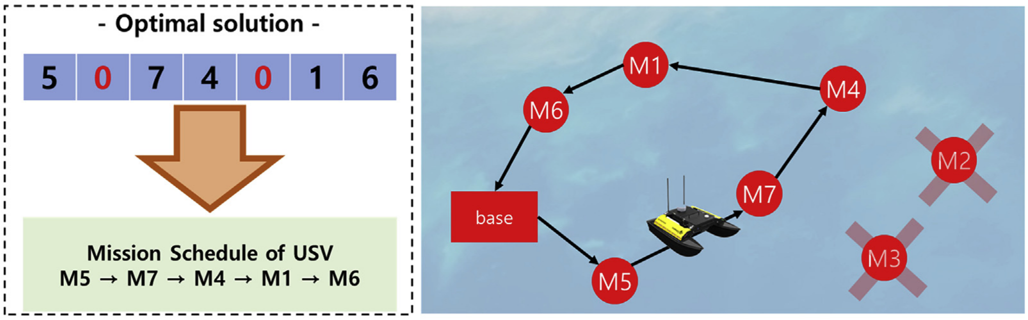

Fig. 3. Expression of the solution.

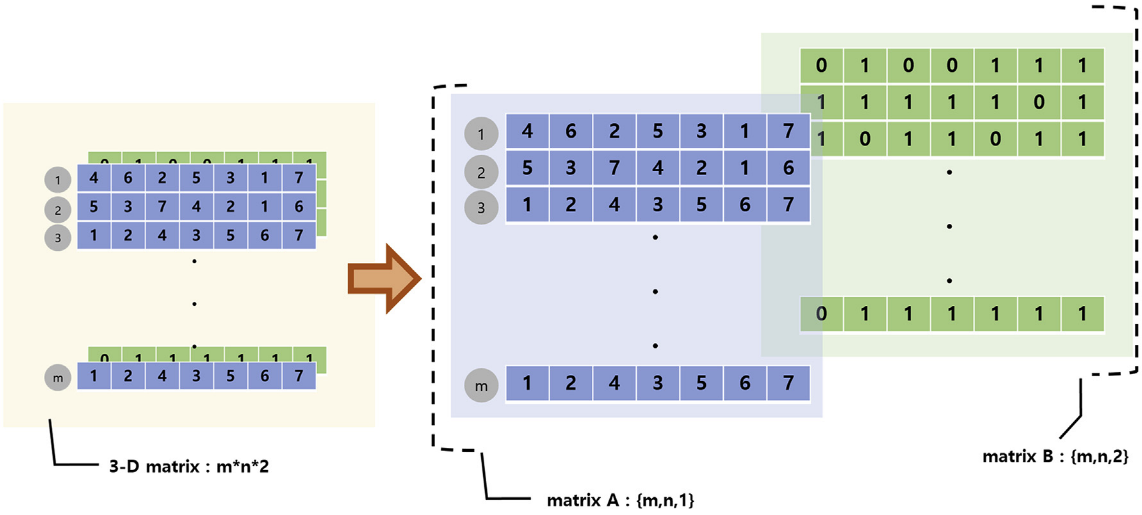

Fig. 4. Concept of 3-D population.

is illustrated in Fig. 8.

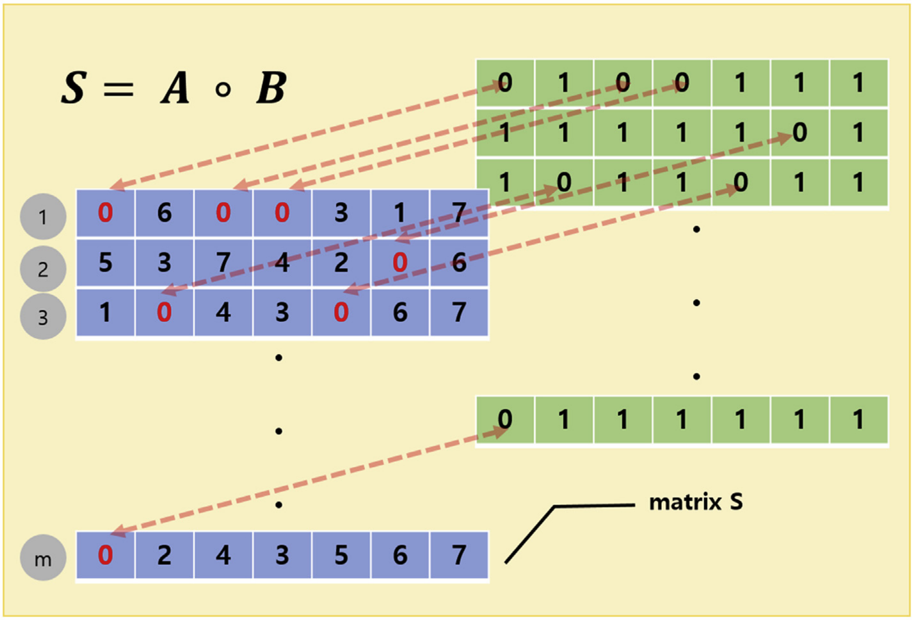

Fig. 5. Determination of sequence.

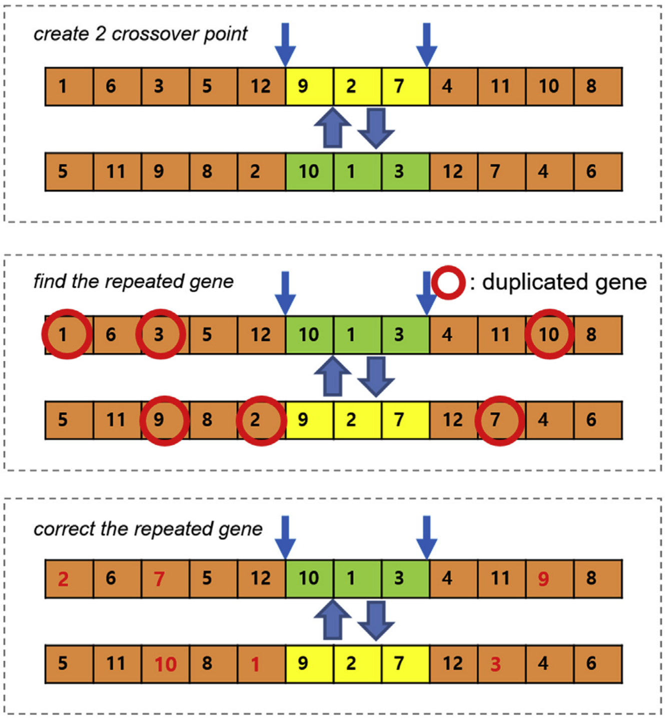

Fig. 6. Crossover scheme for permutation vector.

4.1. MPS (mission planning system)

MPS is a sub-system that operates a mission planning algorithm. It determines the sequence of the given missions to the USV based on the optimal solution of the mission planning algorithm and generates the guidance inputs where the form of guidance input is the coordinates of the mission points and it has been delivered to AGCS in the form of latitude and longitude.

In this study, MPS is composed of a laptop PC, and the mission planning algorithm is compiled in MATLAB. MPS uses one serial port for communication with AGCS (Wu et al., 2013).

4.2. AGCS (autonomous guidance and control system)

This section describes the guidance and control algorithm for the USV to perform missions according to the result of the mission planning. The algorithm is shown in Fig. 9. The optimal solution derived from the mission planning algorithm has a permutation vector form. The USV performs multiple missions according to its sequence (Hwang et al., 2017).

When the USV determines the current target mission, the location information of the mission is also determined. This information is transferred to the guidance algorithm shown in Fig. 10 in the form of latitude and longitude. The guidance algorithm generates attitude and velocity commands for the USV to reach the target point. The commands are transferred to the control algorithm shown in Fig. 11. In this study, the USV plant has a catamaran form, as shown in Fig. 12 in which attitude and speed are controlled by two thrusters.

In this study, Mbed was adopted as the computational processor for AGCS. Three serial ports are used for AGCS. First, one port is used for communication with MPS. Second, another port is used to communicate with GCS. The GCS console and AGCS are connected by a wireless RF modem and the operator can monitor the status of the USV in real-time. The last port is used to communicate with the simulation environment. In a real sailing environment, this port will be used to receive state feedback from the sensor module.

In this study, the state (position, velocity, attitude) of the USV can be directly received from simulation software as a substitution of the sensor measurements in the case of a real sailing environment. To exchange information between Mbed and the simulation system, a protocol was defined. As shown in Fig. 13, each protocol includes Start of Frame (SOF), End of Frame (EOF), a system ID, and checksum. AGCS receives the state of the virtual USV from the simulation system and transmits the control command value of each thruster of the virtual USV.

4.3. Simulation environment using ROS

Each thrust derived from the automatic navigation system is transferred to a virtual USV in the simulation environment, which results in USV movement. In the simulation environment, the state of the USV (speed, attitude, position, etc.) can be transmitted to the AGCS. Therefore, it is possible to replace the sensor function in the actual operation environment (Caccia et al., 2009).

The sailing simulation of the USV was performed using Robot Operation System (ROS) and Gazebo simulator. ROS is an open-source program that has several inter-processor communication or high-performance algorithms that can be easily used with the packages. The Gazebo is mainly used as a simulation program in ROS, which provides a 3D simulation environment and is suitable for collision avoidance or computer vision tests. It also provides various platforms such as USVs, UUVs, UGVs, and UAVs. In this study, a catamaran ship was selected as a USV platform because it is more stable than a mono-hull ship. The USV model was selected as the Heron model provided by Gazebo. The implementation in Gazebo is shown in Fig. 14.

The state of the USV was used for topics that were used for communication in ROS. ROS uses nodes, topics, and messages in communication. Nodes are the least executable term in the processor and communicate based on messages. Topics refer to this method of message communication. The optimal mission sequence is obtained through the control values by the generated genetic algorithm through the control system. The control value is passed to Gazebo via the ROS network and operated in Gazebo. The current status is returned with a topic such as/imu/rpy/filtered and navsat/fix.

5. Simulation and result

5.1. Verification of the mission planning algorithm

Fig. 15 shows the process to verify the optimality of the mission planning algorithm proposed in this study. Since the genetic algorithm is a probability-based optimization technique, we performed 2000 repetitions of the same problem to verify the optimization performance of the proposed method and evaluated the performance through the global optimal solution and the local minima derivation frequency. The global optimal solution used for comparison was derived through Integer Programming (IP). The comparative algorithm, IP always produces global optimal solutions but has a limitation that, unlike the genetic algorithm, it requires a long time for computation to approach problems mathematically.

The mission planning scenarios for the evaluation are configured as follows. i) It is assumed that 20 mission points are given in an area covered by one USV agent. (The corresponding coordinate information of the mission points in this study is shown in Table 1.) ii) The limited operating time is set to 3000 s (50 min). iii) According to the IP, the global optimal solution was found to have a 70% MAR (14 out of 20 missions) and this is shown in Fig. 16.

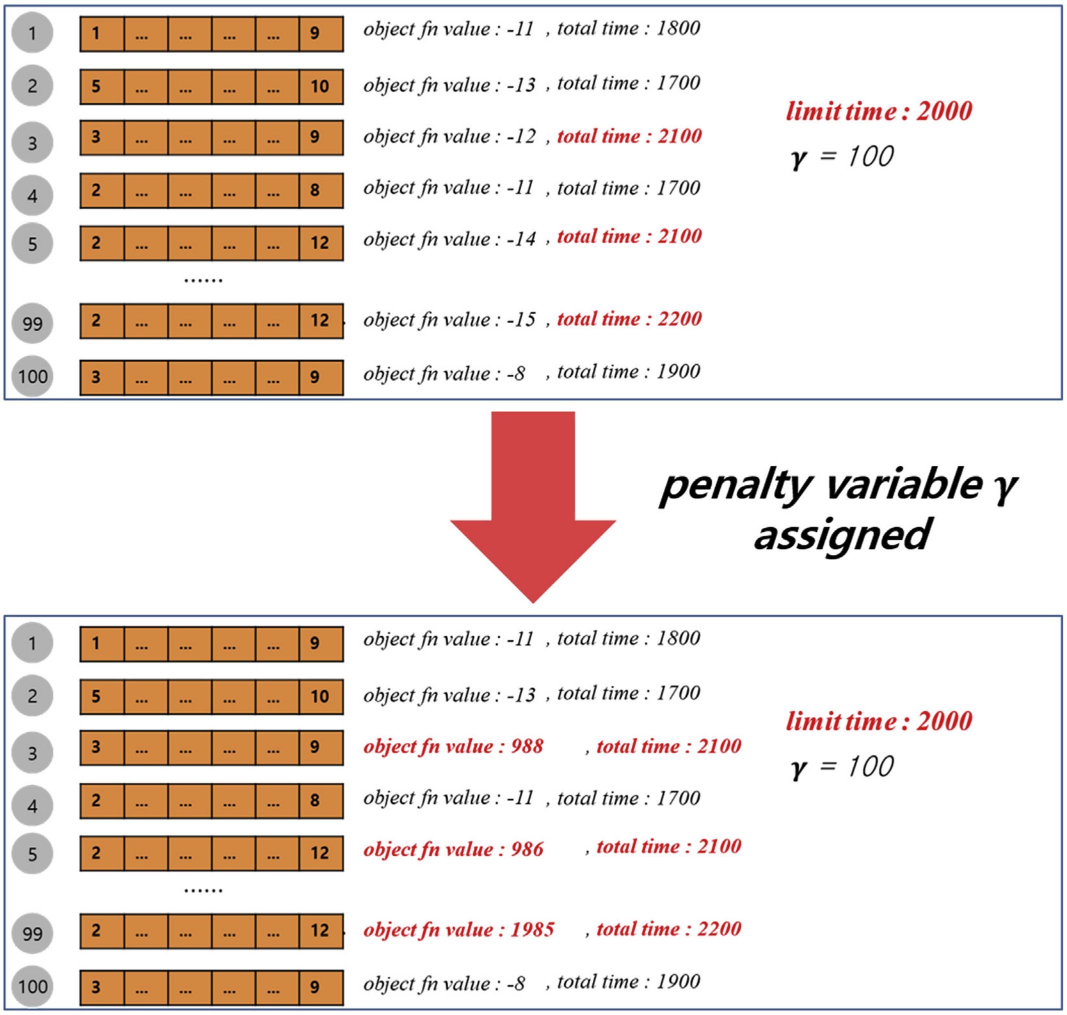
1 | 1 ... ... 9 | object fn value : -11 , total time : 1800
2 | 5 ... ... 10 | object fn value : -13 , total time : 1700
3 | 3 ... ... 9 | object fn value : -12 , total time : 2100
4 | 2 ... ... 8 | object fn value : -11 , total time : 1700
5 | 2 ... ... 12 | object fn value : -14 , total time : 2100
......
99 | 2 ... ... 12 | object fn value : -15 , total time : 2200
100 | 3 ... ... 9 | object fn value : -8 , total time : 1900
limit time : 2000
γ = 100
penalty variable γ assigned
1 | 1 ... ... 9 | object fn value : -11 , total time : 1800
2 | 5 ... ... 10 | object fn value : -13 , total time : 1700
3 | 3 ... ... 9 | object fn value : 988 , total time : 2100
4 | 2 ... ... 8 | object fn value : -11 , total time : 1700
5 | 2 ... ... 12 | object fn value : 986 , total time : 2100
......
99 | 2 ... ... 12 | object fn value : 1985 , total time : 2200
100 | 3 ... ... 9 | object fn value : -8 , total time : 1900
limit time : 2000
γ = 100

Fig. 7. Applying penalty variable.

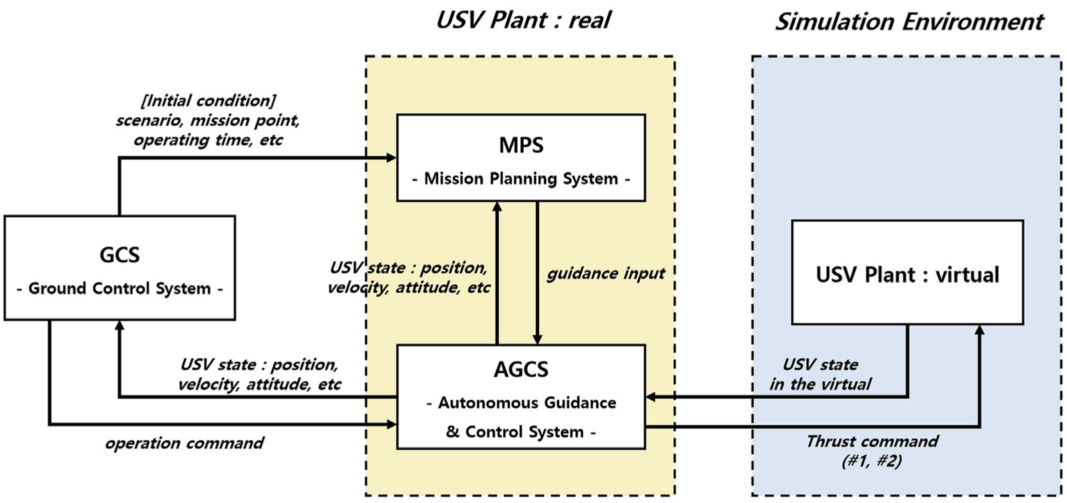
USV Plant : real
Simulation Environment
[Initial condition] scenario, mission point, operating time, etc
MPS
- Mission Planning System -
GCS
- Ground Control System -
USV state : position, velocity, attitude, etc
guidance input
USV Plant : virtual
AGCS
- Autonomous Guidance & Control System -
USV state : position, velocity, attitude, etc
operation command
USV state in the virtual
Thrust command (#1, #2)

Fig. 8. Entire system configuration.

The results of executing the proposed algorithm 2000 times are shown in Fig. 17 and Table 2. During the 2000 cases, the global best solution (the case with 70% MAR) was derived 1932 times. The local minimum (the case with 65% MAR) closest to the global optimal solution was 68 times, and the remaining cases were not detected. In other words, the optimization method presented in this study

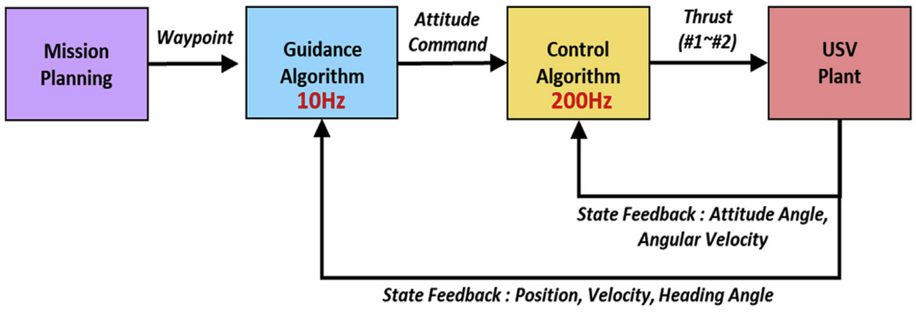

Fig. 9. Autonomous guidance and control algorithm.

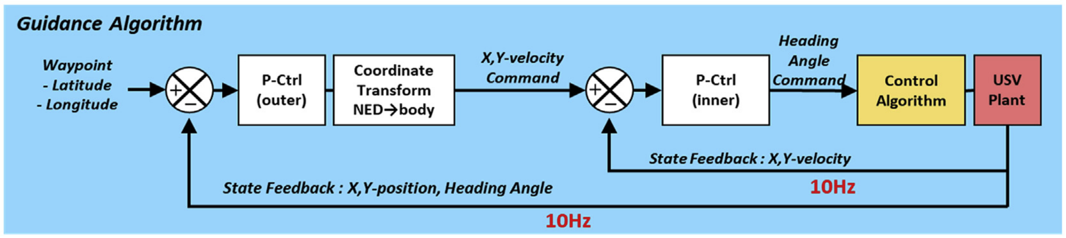
Guidance Algorithm
Waypoint - Latitude - Longitude
P-Ctrl (outer)
Coordinate Transform NED→body
X,Y-velocity Command
P-Ctrl (inner)
Heading Angle Command
Control Algorithm
USV Plant
State Feedback : X,Y-velocity
10Hz
State Feedback : X,Y-position, Heading Angle
10Hz

Fig. 10. Guidance algorithm.

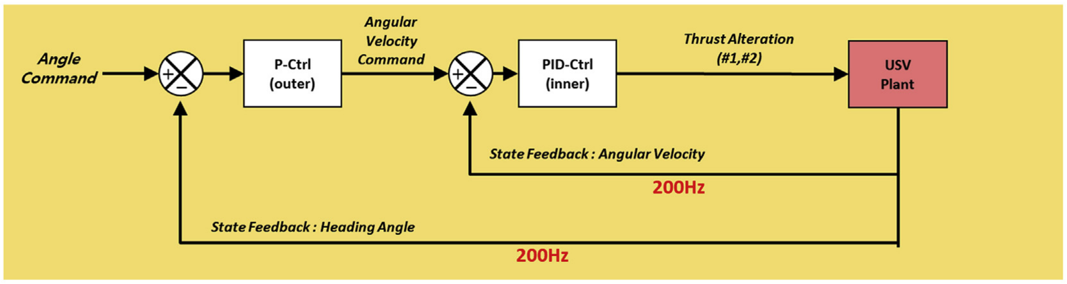
Angle Command
P-Ctrl (outer)
Angular Velocity Command
PID-Ctrl (inner)
Thrust Alteration (#1,#2)
USV Plant
State Feedback : Angular Velocity
200Hz
State Feedback : Heading Angle
200Hz

Fig. 11. Control algorithm.

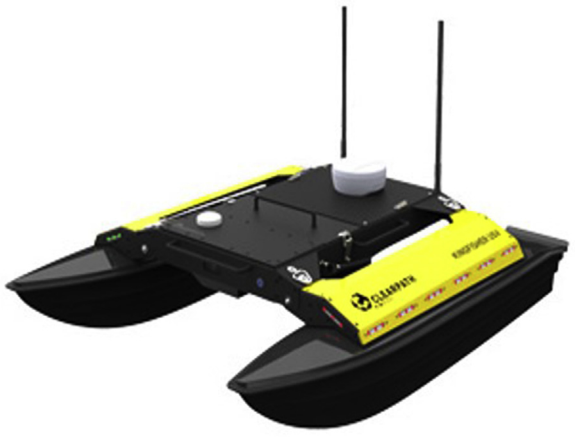

Fig. 12. USV plant in this study.

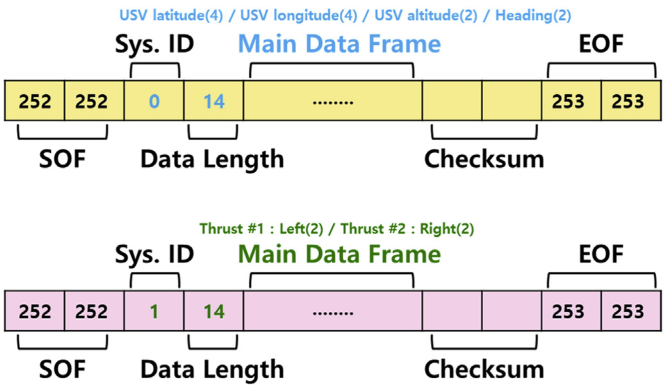

Fig. 13. Protocol between AGCS and simulation system.

has a 96.6% global optimal solution probability. The average operation time of the 2000 cases was about 1.6 s, which is 300 times faster than the IP method. Since the objective function of the mission planning optimization problem is to maximize the MAR, the order of execution may be different even for the same MAR because we set equal importance for all mission points.

5.2. *Verification of the mission planning system in HILS*

Two scenarios were set to verify the applicability of the

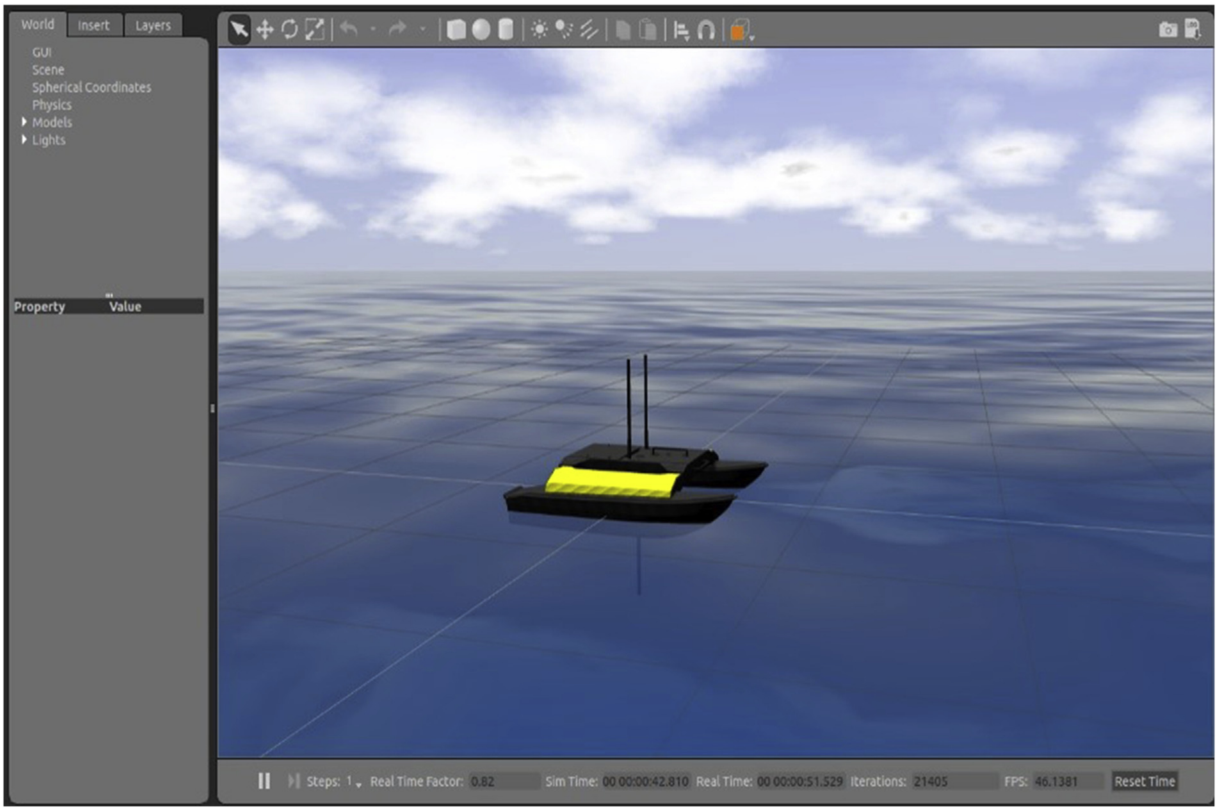

Fig. 14. USV plant in Gazebo simulator.

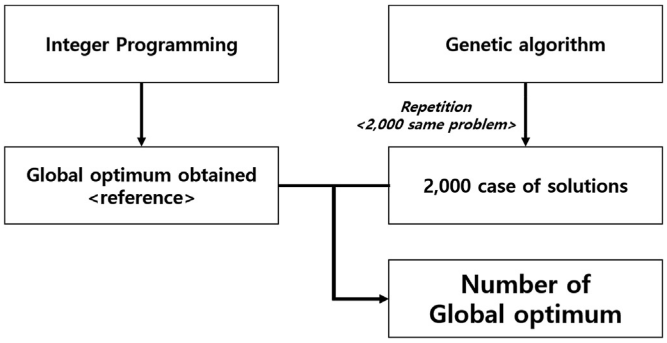

Fig. 15. Process of optimality validation.

proposed mission planning algorithm and system, as shown in Table 3. There are 20 mission points within the area covered by the USV, as shown in Fig. 18 and Table 1. The first scenario is mission planning optimization in an environment without time constraints, assuming that the operating time is sufficient to perform all missions. The optimization target is to minimize the resources (time or fuel) required to perform all missions. The second scenario is mission planning optimization with limited operating time, which is a more realistic situation than the first scenario. In this scenario, the limited operating time was given as 10,000 s.

The results of the first scenario are shown in Table 4 and Fig. 19. USV sailed all mission points as a result of the optimal solution of the mission planning algorithm calculated by MPS. The time to complete the mission was 14,235.67 s. The results of the second scenario are presented in Fig. 20 and Table 4. It shows that 14

Table 1
Coordinates of mission points information in this study.

	Chapter 5.1		Chapter 5.2	
	latitude	longitude	latitude	longitude
BASE	49.9	8.9	49.9	8.9
Mission 1	49.90006	8.900054	49.900022	8.900905
Mission 2	49.90014	8.9	49.899614	8.900326
Mission 3	49.900155	8.900154	49.899745	8.901015
Mission 4	49.900189	8.900296	49.899711	8.899561
Mission 5	49.900254	8.90028	49.900232	8.900256
Mission 6	49.900268	8.900514	49.900191	8.899618
Mission 7	49.900225	8.900578	49.900022	8.900428
Mission 8	49.900212	8.900752	49.899987	8.899076
Mission 9	49.900115	8.900731	49.899372	8.899907
Mission 10	49.900028	8.900809	49.89959	8.899156
Mission 11	49.899996	8.90065	49.899369	8.900449
Mission 12	49.900065	8.900554	49.899545	8.900884
Mission 13	49.899994	8.900535	49.900364	8.900868
Mission 14	49.899951	8.900608	49.900326	8.898652
Mission 15	49.899916	8.900382	49.899787	8.898792
Mission 16	49.900041	8.900318	49.899334	8.89936
Mission 17	49.899959	8.900232	49.900139	8.898486
Mission 18	49.899939	8.900028	49.900592	8.900138
Mission 19	49.899973	8.899827	49.900561	8.89935
Mission 20	49.90006	8.899902	49.89998	8.901388

missions specified in the optimal solution were performed. Within the time limit of 10,000 s, 14 of 20 missions were selected. The time to complete all selected 14 missions was 9699.83 s. Since the result of HILS satisfies the optimal solution as derived from MPS, it was proved that the developed MPS can be applied in a real environment with actual USV.

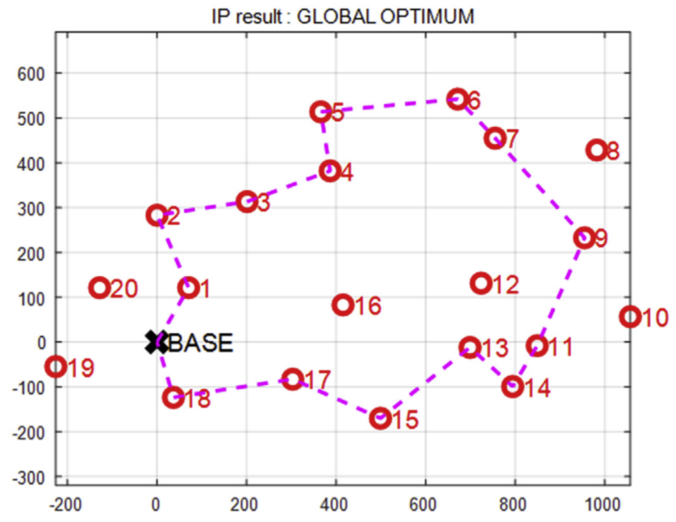

Fig. 16. Global optimum through IP.

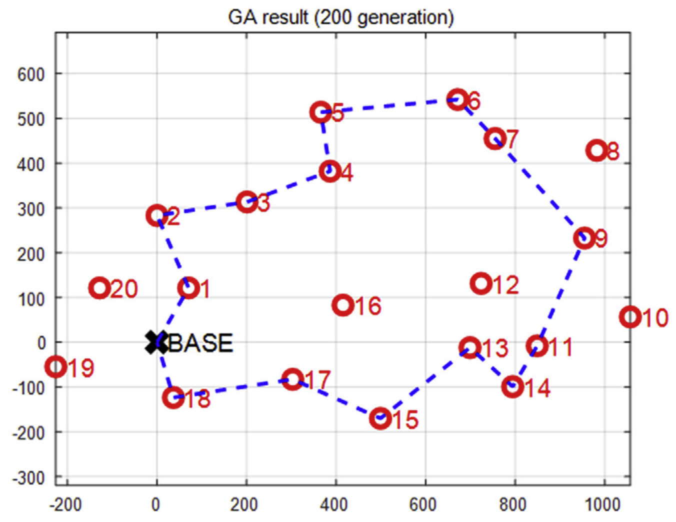

Fig. 17. Global optimum through GA.

Table 2
Result of GA.

GA: 2000 repetitions	
Global optimum	1932 cases (96.6%)
Local minima (error within 5.56%)	68 cases (3.4%)
Other local minima	0%
Average computation time per one iteration	1.6 s
IP: optimality comparison	
Computation time	about 8 min

Table 3
Initial condition.

	Scenario #1	Scenario #2
Obstacles	none	none
# of given missions	20	20
Time limit	inf	10,000s
Objective function	Minimize resource consumption	Maximize mission achievement rate

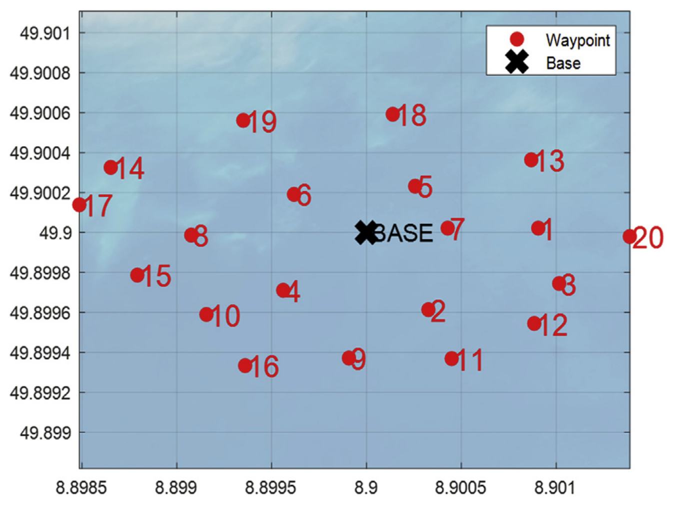

Fig. 18. Initial condition.

6. Conclusion

6.1. Summary

This study contains the process of developing MPS and mission planning algorithm for USV that can be applied in real situations and verifying them through HILS. According to this, scenarios and assumptions were set. Objective functions and constraints were defined to maximize the MAR within a given operating time. The genetic algorithm was adopted to solve the defined optimization problem model and we proposed a method using a 3-D population. This algorithm was implemented through MATLAB and was implemented into MPS to perform calculations.

To verify the optimality of the algorithm, the simulation was repeated 2000 times and the results were compared with the global optimal solution obtained through IP. As a result, 1932 global optimal solutions were derived (96.6%) out of 2000 executions, and the remaining 68 cases (3.4%) were 5.56% off from the global optimal solution. The computation time was 1.6 s, which was 300 times faster than the IP method. Through these results, it was confirmed about strength in the calculation time.

After verifying the optimality of the algorithm, MPS was installed on the actual USV platform. The Autonomous Guidance and Control System (AGCS) was developed for USV operation, and a guidance, navigation, and control algorithm was developed. The simulation environment was built using Gazebo and ROS and in the HILS environment, it was confirmed that the USV in the HILS performed the mission according to the given scenario from MPS.

In conclusion, by installing the mission planning algorithm and MPS developed in this study on the actual USV platform, we secured autonomy in the USV's mission planning. In addition, compared to the integer programming method, the calculation time is only 1.6 s, but it shows the ability to derive more than 95% of the global optimal solution, thus securing the possibility of real-time application of the USV.

6.2. Future works

In this study, the simulation was performed without obstacles. However, in real mission situations, several obstacles are fatal to the USV. Therefore, path planning to avoid them is essential. In addition, the model for the USV's fuel consumption rate or variables that occur during mission performance is simplified, and a more

Table 4
Results of case.

Case A: scenario #1		Case A: scenario #2	
Optimal solution	6, 19, 14, 17, 15, 8, 4, 10, 16, 9, 2, 11, 12, 3, 20, 1, 13, 18, 5, 7	Optimal Solution	6, 8, 10, 16, 9, 2, 11, 12, 3, 1, 13, 5, 7 (14 missions, 70%)
Total time	Inf.	Time Limit	10,000 s
Total Time	14,235.67 s	Total Time	9699.83 s

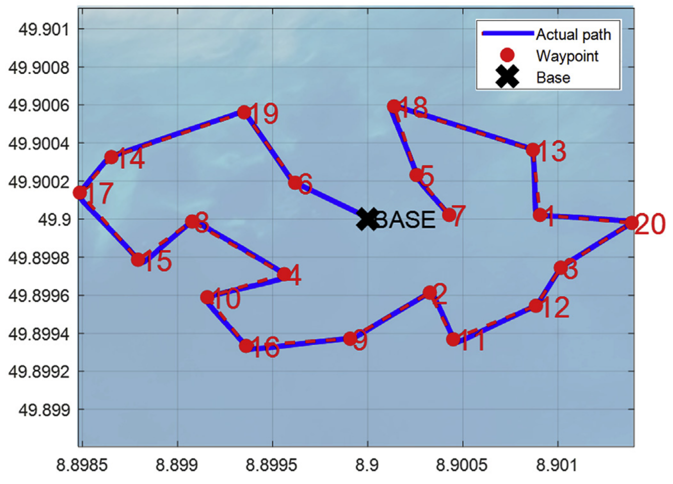

Fig. 19. Result of case: scenario #1.

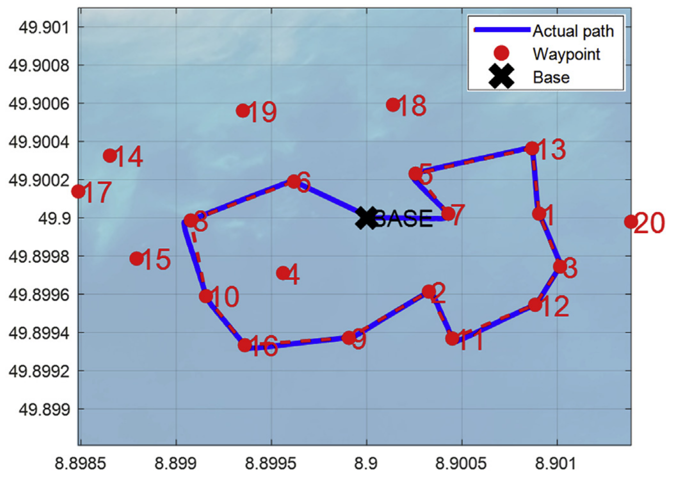

Fig. 20. Result of case: scenario #2.

realistic model expression method for this will be studied in future works.

The mission planning algorithm was executed on the MPS with a laptop PC, and the calculation time was 1.6 s. However, to secure the autonomy of the USV, it is essential to mount an on-board MPS. The research team plans to mount the algorithm on a small on-board PC for an actual sailing experiment.

When the same algorithm is executed on a small on-board PC, the calculation time is longer than that on a laptop PC. To improve this, it is necessary to maximize the calculation speed by parallel operation through a GPGPU to secure real-time mission planning. Ultimately, we intend to contribute to the advancement of USV technology by conducting sailing experiments in a real environment.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by the Technology Innovation Program (20003471, Source Technology Development of a MidAir Separation Reintegration System for Fixed Wing Parent and Child UAVs) funded by the Ministry of Trade, Industry & Energy (MOTIE, Korea).

This work was supported by a 2-Year Research Grant of Pusan National University.

References

Caccia, M., Bibuli, M., Bruzzone, G., Bruzzone, G., Bono, R., Spirandelli, E., 2009. Charlie, a testbed for usv research. IFAC Proceed. Vols. 42 (18), 97–102.
Goldberg, D.E., Lingle, R., 1985, July.). Alleles, loci, and the traveling salesman problem. In: Proceedings of an International Conference on Genetic Algorithms and Their Applications, vol. 154. Lawrence Erlbaum, Hillsdale, NJ, pp. 154–159.
Grefenstette, J., Gopal, R., Rosmaita, B., Van Gucht, D., 1985, July. Genetic algorithms for the traveling salesman problem. In: Proceedings of the First International Conference on Genetic Algorithms and Their Applications, vol. 160. Lawrence Erlbaum, pp. 160–168. No. 168.
Holland, J.H., 1992. Genetic algorithms. Sci. Am. 267 (1), 66–73.
Hwang, H.G., Kim, H.W., Kim, B.S., Woo, Y.T., Shin, I.S., Shin, J.H., Choi, B.W., 2017. A development of integrated control system for platform equipments of unmanned surface vehicle (USV). J. Korea Inst. Inform. Commun. Eng. 21 (8), 1611–1618.
Little, J.D., Murty, K.G., Sweeney, D.W., Karel, C., 1963. An algorithm for the traveling salesman problem. Oper. Res. 11 (6), 972–989.
Miller, C.E., Tucker, A.W., Zemlin, R.A., 1960. Integer programming formulation of traveling salesman problems. J. ACM 7 (4), 326–329.
Noreen, I., Khan, A., Habib, Z., 2016. Optimal path planning using RRT* based approaches: a survey and future directions. Int. J. Adv. Comput. Sci. Appl. 7 (11), 97–107.
Park, H.B., Heo, J.W., Oh, G.W., 2017. The Domestic Development Strategy through International Research Review of Unmanned Surface Vehicle. SASE Fall Conference, pp. 144–145, 2017.
Pisinger, D., Ropke, S., 2007. A general heuristic for vehicle routing problems. Comput. Oper. Res. 34 (8), 2403–2435.
Wu, B., Wen, Y., Huang, Y., Zhu, M., 2013. Research of unmanned surface vessel (USV) path-planning algorithm based on ArcGIS. In: ICTIS 2013: Improving Multimodal Transportation Systems-Information, Safety, and Integration, pp. 2125–2134.
Zanoli, S.M., Astolfi, G., Bruzzone, G., Bibuli, M., Caccia, M., 2012. Application of fault detection and isolation techniques on an unmanned surface vehicle (USV). IFAC Proceed. Vols. 45 (27), 287–292.