



Mission Planning for Underwater Survey with Autonomous Marine Vehicles

Junwoo Jang¹, Haggi Do¹ and Jinwhan Kim²

¹Graduate Student, Department of Mechanical Engineering, KAIST, Daejeon, Korea

²Associate Professor, Department of Mechanical Engineering, KAIST, Daejeon, Korea

KEY WORDS: Autonomous marine vehicles, Persistent autonomy, Multi-robot system, Mission planning, Constrained planning

ABSTRACT: With the advancement of intelligent vehicles and unmanned systems, there is a growing interest in underwater surveys using autonomous marine vehicles (AMVs). This study presents an automated planning strategy for a long-term survey mission using a fleet of AMVs consisting of autonomous surface vehicles and autonomous underwater vehicles. Due to the complex nature of the mission, the actions of the vehicle must be of high-level abstraction, which means that the actions indicate not only motion of the vehicle but also symbols and semantics, such as those corresponding to deploy, charge, and survey. For automated planning, the planning domain definition language (PDDL) was employed to construct a mission planner for realizing a powerful and flexible planning system. Despite being able to handle abstract actions, such high-level planners have difficulty in efficiently optimizing numerical objectives such as obtaining the shortest route given multiple destinations. To alleviate this issue, a widely known technique in operations research was additionally employed, which limited the solution space so that the high-level planner could devise efficient plans. For a comprehensive evaluation of the proposed method, various PDDL-based planners with different parameter settings were implemented, and their performances were compared through simulation. The simulation result shows that the proposed method outperformed the baseline solutions by yielding plans that completed the missions more quickly, thereby demonstrating the efficacy of the proposed methodology.

1. Introduction

The main objective of research on autonomous systems is to allow a complex unmanned system to operate autonomously without user intervention for a long period. The ability to adapt to complex and variable environments is required for the stable operation of autonomous systems for several hours or days without the intervention of users from the outside in real-world environments. To make decisions such as whether to continue a mission under various emergency situations and redesign the procedures for carrying out the mission, a complex autonomous system needs to be able to analyze its situation at a high level and behave accordingly.

Such decision making is more difficult in a marine environment, which experiences significant external disturbances. These disturbances affect the dynamics of a complex autonomous system moving in the environment, increase the system's uncertainty. Furthermore, an unmanned system has a vast operating range in a marine environment, whereas the capabilities of the sensors that can be used in recognition systems are limited. In particular, the

impermeability of water to electromagnetic waves significantly hinders communication underwater, and this communication difficulty makes the operation of complex maritime systems more difficult, increasing the need for research.

Various studies on autonomous marine vehicle (AMV) systems have been conducted, but they mainly considered the navigation, guidance, and control algorithms of marine vehicles focusing on the development of partial autonomy technology. Those studies presuppose intermittent user interventions at times when complicated decision-making is required in the course of performing the mission given to the unmanned system. However, the mission planning capability for specifying and formulating abstract missions for unmanned systems is required, in addition to traditional autonomy technology elements, to build the ultimate AMV that can perform complex missions for a long period without user intervention (Kim and Lee, 2018).

In this study, we developed a series of procedures for planning long-term autonomous missions using a complex autonomous system consisting of an autonomous surface vehicle (ASV) and multiple

Received 13 December 2021, revised 10 January 2022, accepted 10 January 2022

Corresponding author Jinwhan Kim: +82-42-350-1519, jinwhan@kaist.ac.kr

© 2022, The Korean Society of Ocean Engineers

This is an open access article distributed under the terms of the creative commons attribution non-commercial license (<http://creativecommons.org/licenses/by-nc/4.0>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

autonomous underwater vehicles (AUVs). The following mission scenario was considered to show the usefulness and characteristics of this complex autonomous system, which increased the performance efficiency and possibility of successfully completing a mission by using multiple AMVs simultaneously. In this scenario, the complex system departed from the coast, surveyed the ocean floor terrain in distant areas, and then returned. Bathymetry surveys can only be carried out by underwater vehicles. However, because each AUV had a limited usable battery life, it had to be loaded on the ASV and moved to the target area. Furthermore, because there were multiple regions to survey, the AUVs needed to recharge their batteries through the ASV. In this mission scenario, (1) multiple autonomous systems were used, (2) the type of actions that could be selected varied depending on the system, and (3) it considered resource limitations, thus the problem is regarded as a heterogeneous/multiple system problem.

A mission planning algorithm that could deal with the concurrency and complexity of the mission was required to solve the given mission scenario. Many examples of studies on mission planner algorithms can be found because they are applied in various fields, but it is difficult to generalize these algorithms because each application field has different problem characteristics. To overcome this limitation, this study attempted to use the planning domain definition language (PDDL), which is a language tool developed to solve planning problems generally. The PDDL can solve mission planning problems in various fields by describing them using a unified language system (Moon, 2021). However, it may be inefficient to use the PDDL for an actual mission because of the characteristics of the PDDL. When a mission's difficulty is high, the PDDL is often satisfied with producing any mission plans that can be used to achieve the goal rather than producing an optimized mission result.

This paper proposes the adoption of an optimization process using a dynamic planning method to supplement the disadvantage of mission planning using the PDDL. Unlike the planning methods with the PDDL, the optimization techniques studied in the operation research (OR) field can effectively solve traveling salesman problems (TSP) and vehicle routing problems based on predetermined graph structures. The part that optimizes the traveling path in a complex autonomous system is separated and solved using a conventional dynamic programming method from the OR field. Then, based on the optimized path result produced, constraints are added to the PDDL to reduce the search region to be explored in the PDDL. The proposed method was evaluated by applying it to mission scenarios with various levels of difficulty, confirming that better and faster mission planning results can be produced compared to the conventional mission planning method.

2. Developed Mission Planner for AMVs

2.1 Mission Planner in Marine Environment

The mission planner structurally separates the roles of recognizing the surrounding environment and situation, setting mission objectives,

and performing the mission, and simplifies the flow of data, thereby helping the system to operate successfully. The importance of developing a systematic mission planner increases with the complexity of the mission. Hence, we can find examples of using a mission planner in various studies that require actual system operation.

One study proposed a system architecture to build an autonomous system with multiple small vessels and perform mission planning in a marine environment (Elkins et al., 2010). It was designed to separate the decision-making required for performing a mission in the action engine into various levels and facilitate a broad spectrum of decisions from high levels such as collision avoidance to low levels such as the control input values of the actuator. An architectural system that could respond to various situations was designed using a high-level action-based method, and then used to make decisions and control, small ASVs. Nevertheless, this action-based decision-making system had the disadvantage of only being able to respond to situations that the developer could anticipate in advance.

The T-Rex mission planner was proposed to design a mission planning architecture using an AUV and output plans based on the time (McGann et al., 2007). Because it is difficult for an AUV to communicate smoothly, high autonomy is required when using an AUV to survey a marine environment. To deal with the high uncertainty in an underwater environment, a goal-oriented planning method was adopted, which allowed it to perform a mission without clearly defining specific actions. However, the algorithm lacks scalability. Thus, it had the drawback of being difficult to apply to large-scale problems.

Later, a mission planner that could be applied to multiple AUVs was developed, supplementing the T-Rex problem (Py et al., 2016). That study took into account a case where the autonomy systems were heterogeneous, thus considering the communication between an unmanned aerial vehicle and AUVs. An artificial intelligence-based mixed-initiative mission planner was introduced to perform the mission distribution and scheduling of multiple AMVs, and the overall mission planning was performed based on a centralized approach through communication between the AMVs.

A study was conducted on the construction of the ROSPlan mission planner to manipulate structures directly using AUVs (Palomeras et al., 2016). Unlike missions such as collecting environmental samples, missions that involve physically direct intervention require continuous planning because the increased uncertainty of changes based on the action. In this study, one AUV was used to perform the task of turning a valve. In this task environment, the shape or form of the surrounding structures could vary, and it had to be able to cope with the situation where the task was not completed within a set period. Therefore, scheduling was even more important. The plan included the estimated time of completing each task, and if a decision made at the high level was not performed at the low-level control, taking more time than planned, then it was regarded as a failure, and re-planning was performed so that the mission could continue.

The four studies introduced above represent mission planners that

can easily be applied to other studies by expanding their structures, which means that their use can be considered first. In the case of action-based mission planners, it is difficult to evaluate the efficiency of a mission plan because the action decisions vary depending on the developer's capability. Mission planners that used the PDDL have been rated higher in relation to the plan's specificity and mission delivery compared to other mission planners (Martinez et al., 2020). Furthermore, the existence of various PDDL variants is advantageous. Out of these, an appropriate one can be selected based on the situation of the problem to be solved. Therefore, the generalizability and adaptability are high.

2.2 Development of Efficient Planner

Path planning accounts for a large proportion of the mission planning for a mobile robot. If the mission that the vehicle has to perform after moving to the mission's region is not complex, the entire mission planning problem may be replaced by a path planning problem. Path planning is usually divided into a task allocation problem that specifies the mission's region to be visited by multiple robots and a scheduling problem that determines the sequential order for performing the allocated tasks. Many studies have been conducted to solve these two coupled problems in combination.

Temporal planning can be optimized using a genetic algorithm (Miloradović, et al., 2017). If heterogeneous autonomous systems are used, there is a high possibility that the constraints will have high non-linearity. Therefore, the process of finding the solution using a genetic algorithm may be efficient. Because the sensitivity to mutation is high and can easily diverge as a result of the characteristics of the problem in mission planning, the proposed algorithm considers this to maintain a certain number of entities that have no mutation among the entities that are evolving. In addition, it sometimes uses prior knowledge of the problem for mutations. Based on this, a solution close to the optimal solution can be quickly found.

In one attempt to find the optimal solution, the mission planning problem was defined as an OR-based problem (Tsiogkas and Lane, 2018). A constraint was applied where each autonomous vehicle possessed limited resources, and an attempt was made to find a plan where as many tasks as possible could be performed while complying with the constraint on the operating time. Despite utilizing an orienteering problem for the problem definition, the re-planning and environment information were based on the mission planning algorithm's goal.

Another study used the PDDL for mission planning after allocating tasks to increase the efficiency of the mission planning for multiple AUVs (Carreno et al., 2020). Even if the problem was only described using the PDDL, concurrency and heterogeneity could be achieved. Although the efficiency of the mission planning and success rate of the mission plan were low, task allocation that considered the heterogeneity, which was proposed in the paper, improved this significantly.

As the above previous studies have shown, the use of research in the

OR field has the advantage of allowing the mission performances of multiple mobile robots to effectively be optimized, but it is difficult to specify and formulate the mission, such as defining actions at a high dimension, like the PDDL. Combining a conventionally used algorithm with high optimality and the PDDL, which can be used to formulate abstract missions, as in Carreno et al. (2020), has recently been studied by multiple researchers because these two independent research fields can complement each other (Muñoz et al., 2016; Silver et al., 2020; Kim et al., 2019).

3. Mission Planning Using PDDL

The PDDL is a language system developed to standardize mission planning algorithms. It was developed based on the Stanford research institute problem (STRIPS) and action description language (ADL), which are conventional mission planning languages. The PDDL by itself does not provide a mission planning function but makes it possible to apply the same mission planning algorithm to mission planning problems in various fields by defining them with a unified grammar. However, various versions of the PDDL and extended PDDLs have been developed based on the characteristics required in various mission planning problems.

3.1 PDDL Variations

PDDL 1.x, which is the most basic version of the PDDL, is driven by a logical modeling method and based solely on the true or false values of given propositions. We define each variable's state value and the action model in advance. The action model can change the states of variables under the given condition, and only the basic logical operators (and/or) can be used in the given condition. The mission planning algorithm finds a sequence of actions to reach the target state from the initial state, where the basic goal is reaching it through a minimum number of actions. Depending on the version, it is also possible to define the cost for each action in advance and set a goal of minimizing the sum of the cost functions of the action sequence.

PDDL 2.1 significantly improved the expression capability of the mission planning language through the adoption of time and numeric variables, which could not previously be represented (Geffner, 2003). It adopts period parameters for modeling the time required for the tasks to produce a plan based on time, and uses three new points to apply the conditions of the operation start time, operation termination time, and operation duration. For variables that change as the result of an action, the effect can also be applied at the start and end of the action in a similar way. Numeric variables enable the expression of continuous resources such as batteries and materials, facilitating the planning of more realistic mission scenarios.

PDDL 3.0 allows the adoption of preferred constraints in mission plans. In other words, it was extended to allow the language to reflect a user's preference, even if various actions are possible. In each version of the PDDL, every action changes the state variables deterministically. In the probabilistic PDDL or relational dynamic

influence diagram language (RDDL), however, the action model is defined probabilistically, which makes it possible to describe problems involving a partially observable Markov decision process (POMDP).

3.2 PDDL 2.1 Definition

PDDL 2.1 is a typically used for the mission planning algorithms of heterogeneous and multiple systems because it supports a function for planning the concurrent action commands of multiple agents. PDDL 2.1 aims to minimize the time required for the mission plan and uses a tuple consisting of $\langle I, A, G, M \rangle$.

- I refers to the initial state and consists of a true/false logical variable and numeric variable (v).
- A refers to the possible action set, where each action is defined by the prerequisites, execution effects, and action's duration. ($\langle dur, pre_+, pre_-, pre_-, eff_+, eff_- \rangle$) The prerequisites are divided into conditions that must be satisfied before, after, and while the action is being performed. For the execution effects, the variables that change before and after the action starts can be defined.
- G is the goal state and refers to the variable condition that is to be finally reached.
- M is the cost function to be optimized and is defined as the total time required to perform the mission.
- The logical variable for each prerequisite and effect can use only the basic logical operators.
- The following operators can be used by the numeric variable for each prerequisite and effect:

$$pre : \langle f(v), op, c \rangle, s.t. op \in \{ \leq, <, =, >, \geq \}, c \in \mathbb{R} \quad (1)$$

$$eff : \langle v, op, f(v) \rangle, s.t. op \in \{ \times, +, =, -, \div \} \quad (2)$$

When a mission planning problem is described as above through PDDL 2.1, the mission planning algorithm finds a series of action sequences to reach the goal state from the initial state. The algorithm tries to find a solution that minimizes the time required while accomplishing the mission as represented in eqn. (3), but when it is difficult to find the optimal solution because the problem's complexity is large, it may aim to find only an adequately satisfactory solution.

$$\text{argmin} (M(A_{sol})) \quad s.t. A_{sol} = \{ a_{1:n} \mid a_1 \in A, a_n \circ \dots \circ a_1 (I) = G \} \quad (3)$$

3.3 PDDL 2.1 Planner

To solve a problem described using PDDL 2.1, a mission planning algorithm that analyzes the problem type and searches for a solution must be used. The mission planning algorithms that can be used in marine environment mission planning are as follows. Continuous linear planner (COLIN) (Coles et al., 2012) is a heuristic-based

mission planner that proceeds while examining the survey region in the forward direction, and it facilitates the inference of continuous linear changes. Here, heuristic refers to a decision-making method set based on experience and prior knowledge. The linear programming method and fast forward type-search algorithm have been combined in the COLIN algorithm's practical implementation process. Forward-chaining partial-order planning (POPF) (Coles et al., 2021) was constructed based on the COLIN algorithm, and it uses the idea of proceeding with the mission plan by partially dividing each stage in the process of searching for the solution. It aims to find a solution that satisfies the given condition by finding the constraints that must be solved sequentially, rather than actions that are executed after certain conditions have already been satisfied. Optimizing preferences and time-dependent costs (OPTIC) (Benton et al., 2012) manages the interaction between the preferred time constraints and the mandatory time constraints occurring in the mission planning stage based on the integer programming method.

4. Methodology

4.1 Mission Scenario

Fig. 1 shows a complex system that includes an ASV and AUVs, which is currently being developed in a project involving the source technology for AMVs. This paper considers various scenarios to produce mission planning results. In this research and development, we used a complex system consisting of an ASV and three AUVs loaded on the ASV to carry out a mission that included a bathymetry survey and creation of a digital map at a depth of more than 2,000 m. A buoyancy-controlled AUV conducted a survey independently over a broad region and long period of time, while the bathymetry survey was conducted using two power-controlled AUVs. The ASV had the role of transporting and recharging the AUVs. The main actions of this complex autonomous system consisted of the ASV traveling to the survey point, the deployment and docking of the AUVs, and the bathymetry survey by the AUVs. The goal was to perform this mission autonomously over an operating distance of up to 1,000 km and a period of up to seven days. The mission planner for the complex system needed high reliability to perform the mission autonomously for a long period.

4.2 PDDL Description

The problem was described using the grammar of PDDL 2.1 based on the presented scenario. The PDDL consists of domain files that define the rules of a problem to be solved and problem files containing the environment setting information. The problem files are composed separately. Thus, it has the advantage of being able to easily solve a variety of problems in the same domain.

The state variables and action model were defined in the domain file according to the PDDL 2.1 grammar. The state variables included the complex system's navigation information, environment information, battery status information, and fault diagnostic information. The

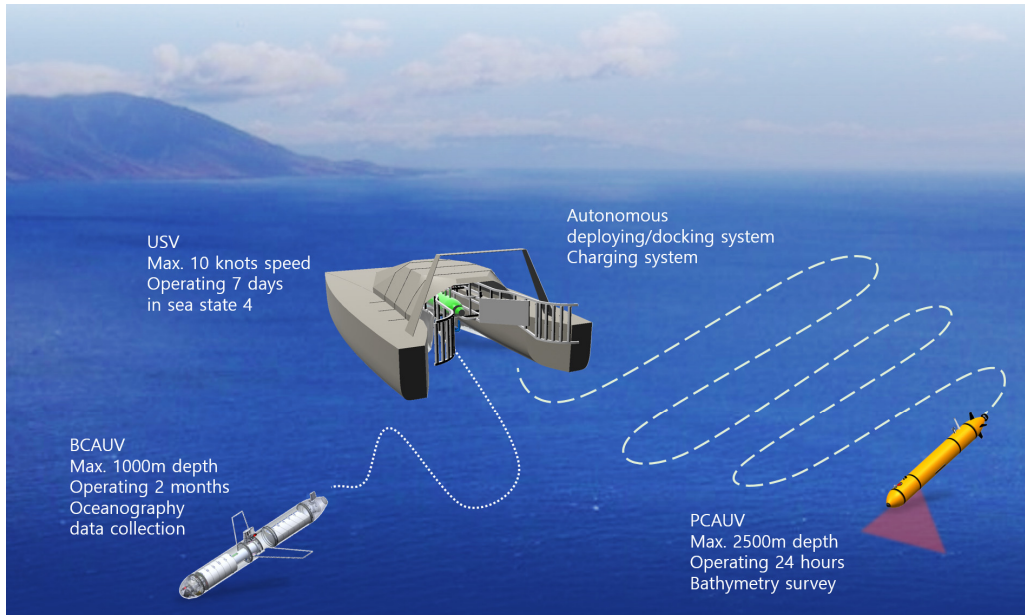


Fig. 1 Example of a mission using an unmanned surface vehicle and autonomous underwater vehicle in the marine environment (Courtesy of KRISO)

Table 1 Part of domain files and problem files following PDDL 2.1 format

Domain file	Problem file
(:types AUV waypoint region)	(:init
(:predicates	(= (distance wp1 wp2) 200)
(located ?wp - waypoint)	(= (distance wp1 wp3) 320)
(AUVlocated ?wp - waypoint ?a - AUV)	(= (regionSize r1) 60)
(surveyed ?r - region)	(= (regionSize r2) 60)
(onUSV ?a - AUV)	(movable wp1 wp2)
...	(movable wp2 wp3)
(:functions	(USVavailable)
(battery ?a - AUV)	(AUVavailable a1)
(consumption)	(onUSV a1)
(surveyspeed)	(= (surveySpeed) 5)
(USVspeed)	(located wp1)
...	...
(:durative-action move	(:goal (and
:parameters (?wpi ?wpi - waypoint)	(surveyed r1)
:duration (= ?duration (/ (distance ?wpi ?wpi) (USVspeed)))	(surveyed r2)
:condition (and	(located wp1)
(at start (USVavailable))	(onUSV a1)
(at start (located ?wpi))	(onUSV a2)
(at start (movable ?wpi ?wpi))	(BCgone)
:effect (and	
(at start (not (located ?wpi)))	
(at start (not (USVavailable)))	
(at end (USVavailable))	
(at end (located ?wpi))	
(at end (increase (total-distance) (distance ?wpi ?wpi)))	

regions to be surveyed and waypoints were also set up. In the case of the exploration region survey by the AUVs, we assumed that there were pre-planned survey patterns that would make it possible to conduct the survey once the AUVs arrived at the survey region. The

ASV has to transport the AUVs to the waypoints included in the survey region and then deploy them. Because there could be multiple waypoints in the survey region, as well as waypoints outside the survey region, the survey region and waypoint variables were set

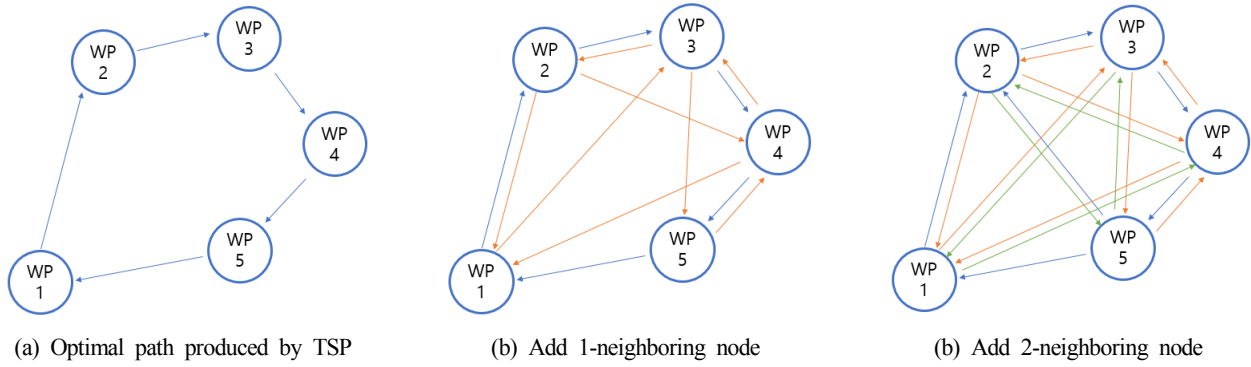


Fig. 2 Optimal path obtained from solving the traveling salesman problem and additional movable paths.

separately. The action model included the travel of the ASV, deployment and docking of the AUVs, exploration region survey by the AUVs, and recharging of the AUVs at the ASV.

- move (?wpi, ?wpf):** The ASV moves from Waypoint $?wpi$ to Waypoint $?wpf$.
- deploy (?a, ?wp):** The $?a$ -th power-controlled AUV is deployed at Waypoint $?wp$.
- BC_deploy (?wp):** The buoyancy-controlled AUV is deployed at Waypoint $?wp$.
- survey (?a, ?r, ?wp):** The $?a$ -th power-controlled AUV surveys exploration region $?r$, starting from Waypoint $?wp$.
- dock (?a, ?wp):** The $?a$ -th power-controlled AUV is docked at Waypoint $?wp$.
- charge (?a):** The $?a$ -th power-controlled AUV mounted on the ASV is recharged.

The problem file contained information about the initial state of the mission and goal state to be reached. The initial state information included the values of all the state variables used in the domain file, such as the traveling speed of the ASV, charging speed of the AUV, survey region's size, and locations of waypoints. The goals were set to deploy the buoyancy-controlled AUV based on the scenario, complete the survey in all the designated regions, and return to the initial starting position with the power-controlled AUVs. Table 1 shows an example of domain files and problem files created.

4.3 Constrained PDDL Using TSP

The problem presented in the scenario could not simply be replaced by a traveling path planning problem because it included the functions for deploying, docking, and charging the AUVs. However, as the number of survey regions increased, the optimality of the planned mission was greatly affected by the ASV's mission routes. Because the ASV had two AUVs that could perform surveys, the ASV could travel repeatedly to deploy and recover these AUVs at different areas, which allowed the mission to be carried out faster. However, because this required the AUVs to be fully charged for surveying, the ASV's optimal travel path could be different from the optimal travel path produced when only the

reduction of the traveling distance was considered. The optimal travel path that reduced the traveling distance could be used because it significantly reduced the cost function result for traveling.

When waypoints were given in the problem, the solution that minimized the traveling distance was found through the OR tool that solved the TSP. In the case of TSP optimization, even if more than 100 waypoints were given, there was no significant increase in the computation time for the entire mission plan because the optimization could be performed in less than 1 s. Afterward, a movable variable was added to the problem file so that the ASV could move along the optimal path found. In this case, the number of possible cases of the move action that the ASV could take at the waypoint where the ASV was located was reduced from (Total Number of Waypoints - 1) to 1, which meant the survey regions could be reduced significantly.

Because the optimal path of the ASV found by the TSP was different from the optimal path of the problem that was to be solved originally, moveable paths could be added to compensate for this. In this paper, we add the condition that it is possible to move to a nearby waypoint in the optimal path sequence produced in the TSP with heuristics. Fig. 2 shows the moveable paths added according to the neighboring range when it is possible to move to the neighboring waypoints in the optimal path sequence.

5. Experiments and Results

The mission planning algorithm was executed in various environments to evaluate the proposed algorithm's performance. Based on the given mission scenario, the complex system had to deploy the buoyancy-controlled AUV and use two power-controlled AUVs to survey underwater regions within a range of 1,000 km. The survey regions were randomly generated within the range, and the number of survey regions was changed to examine the mission performance in relation to the difficulty level of the problem. Because each mission planner generated the same mission planning result for the same problem setting, ten problems were created for each number of survey regions, and each algorithm was used to solve these problems. The mission planning algorithms had a time constraint of 1 min, and the best result produced within this time was used.

5.1 Discussion on Planned Result

To examine the characteristics of the mission planning results using the PDDL before comparing the performances of the different algorithms, Table 2 summarizes the mission planning results of the algorithms for a relatively simple problem where there were two survey regions. As shown in the table, the total mission duration varied depending on the mission planning algorithm used. The convergence to the same optimal solution was found in the POPF and OPTIC algorithms, but the solution did not converge within 1 min in the COLIN algorithm. In the case of COLIN, the convergence to the optimal solution was confirmed when 2 min were given for the computation time.

In the mission planning results produced by POPF and OPTIC, it was found that the ASV moved to Waypoint 3 at 19.152, and at the same time, AUV1 surveyed the r0 region. Thus, PDDL 2.1 could plan multiple, heterogeneous missions simultaneously because each action was performed based on time. Furthermore, to minimize the final mission duration, the AUV may not be recovered immediately after completing the survey in a region; rather, it could be recovered later after completing another mission and returning. If it would be a

problem in a real-world operation to let the AUV remain underwater for a long time without recovering it, this could be solved by adding constraints. In the light of these results, the mission planner could be effective in solving complex problems, but it is necessary to verify the results in various scenarios before using them in real operations and check whether the user approves them for each situation.

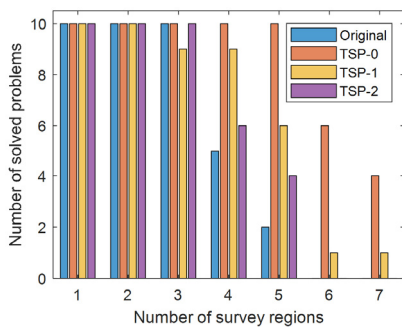
5.2 Mission Planning Success Rate

The difficulty of the problem tended to increase with the number of survey regions. Consequently, there were cases where an action sequence that could reach the goal state from the initial state within a limited time of 1 min could not be found. The mission success rate was examined based on the randomly generated number of survey regions in the mission scenarios, and Fig. 3 shows the results.

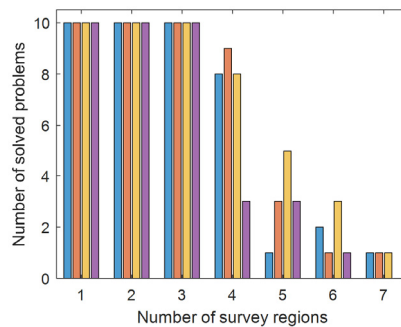
If constraints were added to the PDDL problem using the TSP in advance, the number of usable action cases decreased, but at the same time, the width of the action for reaching the target became narrower. Therefore, the success rate of the mission planning results did not increase unconditionally. Nevertheless, it was found that the mission success rate was generally higher or similar when constraints were

Table 2 Results of planning with PDDL 2.1 according to the planners. Each action sequence is represented as <start time: (action) [action duration]>.

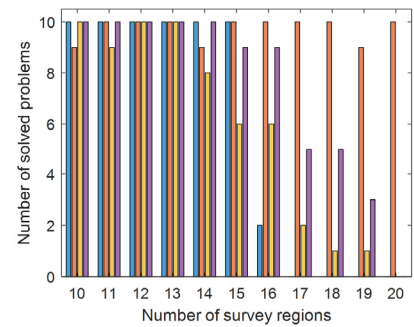
COLIN	POPF	OPTIC
0.000: (move wp0 wp2) [18.150]	0.000: (move wp0 wp2) [18.150]	0.000: (move wp0 wp2) [18.150]
18.151: (deploy a1 wp2) [1.000]	18.151: (deploy a1 wp2) [1.000]	18.151: (deploy a1 wp2) [1.000]
19.152: (survey a1 r0 wp2) [20.000]	19.152: (move wp2 wp3) [32.350]	19.152: (move wp2 wp3) [32.350]
19.153: (move wp2 wp3) [32.350]	19.152: (survey a1 r0 wp2) [20.000]	19.152: (survey a1 r0 wp2) [20.000]
51.504: (deploy a2 wp3) [1.000]	51.503: (deploy a2 wp3) [1.000]	51.503: (deploy a2 wp3) [1.000]
52.505: (survey a2 r1 wp3) [20.000]	52.504: (survey a2 r1 wp3) [20.000]	52.504: (survey a2 r1 wp3) [20.000]
72.506: (dock a2 wp3) [1.000]	72.505: (dock a2 wp3) [1.000]	72.505: (dock a2 wp3) [1.000]
73.507: (move wp3 wp2) [32.350]	73.506: (move wp3 wp1) [45.300]	73.506: (move wp3 wp1) [45.300]
86.860: (charge a2) [20.000]	118.807: (bc_deploy wp1) [1.000]	118.807: (bc_deploy wp1) [1.000]
105.858: (dock a1 wp2) [1.000]	119.808: (move wp1 wp2) [22.400]	119.808: (move wp1 wp2) [22.400]
106.859: (charge a1) [20.000]	142.209: (dock a1 wp2) [1.000]	142.209: (dock a1 wp2) [1.000]
106.861: (move wp2 wp1) [22.400]	143.210: (move wp2 wp0) [18.150]	143.210: (move wp2 wp0) [18.150]
129.262: (bc_deploy wp1) [1.000]		
130.263: (move wp1 wp0) [32.800]		



(a) COLIN



(b) POPF



(c) OPTIC

Fig. 3 Number of mission successes according to the mission planner. OPTIC algorithm can find all solutions up to 9 survey regions. TSP-0 means that the movable path is set as the optimal path from the TSP result. TSP-1 and TSP-2 are cases where moving to 1 and 2 neighboring nodes is possible, respectively.

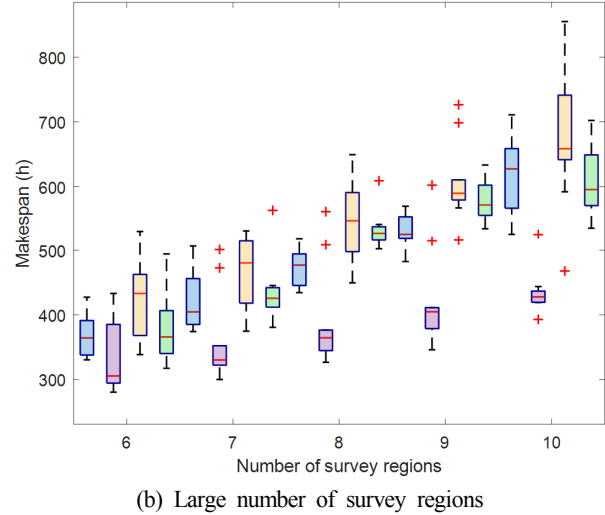
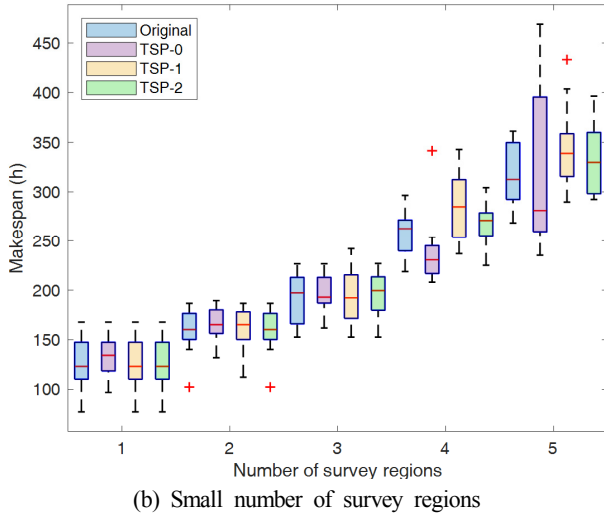


Fig. 4 Makespan according to the number of survey regions. As the mission planner minimizes makespan. Only the case of 10 survey regions in TSP-2 uses 9 samples, and 10 samples are used for others.

added. The mission success rate declined, starting from when the number of survey regions was four in the case of the COLIN and POPF algorithms. A high success rate was obtained when neighboring waypoints were not used in the case of COLIN and when a neighboring waypoint was used in the case of POPF. In the case of OPTIC, the success rate started to decrease after 16 survey regions because the stability was higher in a successful mission plan with this algorithm compared to the other two algorithms. When constraints were added in the PDDL, the mission sometimes failed in the past, but when the scale of the problem increased, the success rate was high. It was also confirmed that the number of successes was maintained at nine or higher for up to 24 survey regions in the case of TSP-0.

5.3 Mission Planning Performance

Using box-and-whisker plots, Fig. 4 shows the total mission durations of the mission planning results of the OPTIC mission planning algorithm. In the case of the COLIN and POPF algorithms, the mission success rate dropped drastically as the number of survey regions increased. Furthermore, because the pattern in the case of a small number of survey regions also did not show much difference from that of OPTIC, we only show the results of OPTIC.

When the number of survey regions was small, the scale of the problem was small, and the algorithm's optimization could be performed within a given limited time of 1 min. In this case, the performance of TSP-0, which had constraints on the waypoints where the AUV could move, was not good compared to other cases, but the performance difference was not very large. On the other hand, when the number of survey regions was large, the scale of the problem increased, making it difficult to get sufficiently close to the optimal solution within the limited time. In addition, because the TSP constraints provided heuristics close to the optimal solution, a good solution could be found more quickly. In TSP-1, the performance tended to decrease, and in TSP-2, although there was volatility, the performance was generally similar. Nevertheless, the performance did

not degrade entirely because both cases resulted in a higher mission success rate when the number of missions increased further to 16 or more. In sum, for the constraint used in finding an optimal solution, the effects of the constraint on the solution had a complex trade-off relationship, because the solution approached the optimal solution based on the strength of the constraint. Therefore, we can say that in this scenario, TSP-0 and TSP-2 provided more significant heuristics, improving the mission planning algorithm's performance.

6. Conclusions

Complex autonomous systems open up the possibility of performing high-difficulty missions over a long period, which could not be performed previously, through the collaboration of various systems. Complex autonomous systems constructed for use in a marine environment where communication is highly restricted and uncertainty is high should have highly reliable autonomy. To this end, it is necessary to analyze and act at a high level in terms of determining situations and making decisions. PDDL-based mission planning has the drawback that it is difficult to optimize a complex problem with a large task size because of the mission planning characteristics. However, such algorithms are commonly used in mission planners because they make it possible to proceed with a mission plan at the action level, and the generalizability and adaptability of mission expression are high. At the same time, when using complex autonomous systems, path planning accounts for a large portion of the total mission plan, and this problem can be solved more efficiently using an OR-based dynamic programming method. In this paper, we improved the performance of mission planning by applying OR-based path planning results to PDDL-based mission planning problems. The proposed method improved the success rate of the mission plan and the optimality of the mission planning result, thus showing that mission planning with higher reliability is possible.

Conflict of Interest

Jinwhan Kim serves as an editor of *Journal of Ocean Engineering and Technology*, but has no role in the decision to publish this article. No potential conflict of interest relevant to this article was reported.

Funding

This research was supported by National R&D Program through the National Research Foundation of Korea (NRF) funded by Ministry of Science and ICT (2020M3C1C1A02086303).

References

- Benton, J., Coles, A., & Coles, A. (2012, May). Temporal Planning with Preferences and Time-Dependent Continuous Costs. *Proceedings of the Twenty-Second International Conference on Automated Planning and Scheduling*.
- Carreno, Y., Pairet, È., Petillot, Y., & Petrick, R.P. (2020, June). A Decentralised Strategy for Heterogeneous AUV Missions via Goal Distribution and Temporal Planning. *Proceedings of the International Conference on Automated Planning and Scheduling*, 30(1), 431–439. Retrieved from <https://ojs.aaai.org/index.php/ICAPS/article/view/6738>
- Coles, A., Coles, A., Fox, M., & Long, D. (2021). Forward-Chaining Partial-Order Planning. *Proceedings of the International Conference on Automated Planning and Scheduling*, 20(1), 42–49. Retrieved from <https://ojs.aaai.org/index.php/ICAPS/article/view/13403>
- Coles, A., Coles, A., Fox, M., & Long, D. (2012). COLIN: Planning with Continuous Linear Numeric Change. *Journal of Artificial Intelligence Research*, 44, 1–96. <https://doi.org/10.1613/jair.3608>
- Elkins, L., Sellers, D., & Monach, W.R. (2010). The Autonomous Maritime Navigation (AMN) Project: Field tests, Autonomous and Cooperative Behaviors, Data Fusion, Sensors, and Vehicles. *Journal of Field Robotics*, 27(6), 790–818. <https://doi.org/10.1002/rob.20367>
- Geffner, H.A. (2003). PDDL 2.1: Representation vs. Computation. *Journal of Artificial Intelligence Research*, 20, 139–144. <https://doi.org/10.1613/jair.1995>
- Kim, B., Wang, Z., Kaelbling, L.P., & Lozano-Pérez, T. (2019). Learning to Guide Task and Motion Planning Using Score-Space Representation. *The International Journal of Robotics Research*, 38(7), 793–812. <https://doi.org/10.1177/0278364919848837>
- Kim, W.J., & Lee, K. (2018). A Study of the Development Test and Evaluation and Verification Procedure of a Multi-Mission USV, M-Searcher. *Journal of Ocean Engineering and Technology*, 32(5), 402–409. <https://doi.org/10.26748/KSOE.2018.6.32.5.402>
- Martinez, N.L., Martínez-Ortega, J.F., Castillejo, P., & Beltran Martinez, V. (2020). Survey of Mission Planning and Management Architectures for Underwater Cooperative Robotics Operations. *Applied Sciences*, 10(3), 1086. <https://doi.org/10.3390/app10031086>
- McGann, C., Py, F., Rajan, K., Thomas, H., Henthorn, R., & McEwen, R. (2007, September). T-rex: A Model-Based Architecture for Auv Control. In *3rd Workshop on Planning and Plan Execution for Real-World Systems*.
- Miloradović, B., Çürüklü, B., & Ekström, M. (2017, July). A Genetic Mission Planner for Solving Temporal Multi-Agent Problems with Concurrent Tasks. *Advances in Swarm Intelligence*, 481–493. https://doi.org/10.1007/978-3-319-61833-3_51
- Moon, J. (2021). Centralized, Distributed, Hybrid Task Planning Framework for Multi-Robot System in Diverse Communication Status. *Journal of Positioning, Navigation, and Timing*, 10(3), 215–220. <https://doi.org/10.11003/JPNT.2021.10.3.215>
- Muñoz, P., R-Moreno, M.D., & Barrero, D.F. (2016). Unified Framework for Path-Planning and Task-Planning for Autonomous Robots. *Robotics and Autonomous Systems*, 82, 1–14. <https://doi.org/10.1016/j.robot.2016.04.010>
- Palomeras, N., Carrera, A., Hurtós, N., Karras, G.C., Bechlioulis, C.P., Cashmore, M., ... Kormushev, P. (2016). Toward Persistent Autonomous Intervention in a Subsea Panel. *Autonomous Robots*, 40(7), 1279–1306. <https://doi.org/10.1007/s10514-015-9511-7>
- Py, F., Pinto, J., Silva, M.A., Johansen, T.A., Sousa, J., & Rajan, K. (2016, October). Europtus: A Mixed-Initiative Controller for Multi-Vehicle Oceanographic Field Experiments. In *International Symposium on Experimental Robotics*, 323–340. https://doi.org/10.1007/978-3-319-50115-4_29
- Silver, T., Chitnis, R., Curtis, A., Tenenbaum, J., Lozano-Perez, T., & Kaelbling, L.P. (2020). Planning with Learned Object Importance in Large Problem Instances Using Graph Neural Networks. *arXiv preprint arXiv:2009.05613*.
- Tsiogkas, N., & Lane, D.M. (2018). An Evolutionary Algorithm for Online, Resource-Constrained, Multivehicle Sensing Mission Planning. In *IEEE Robotics and Automation Letters*, 3(2), 1199–1206. <https://doi.org/10.1109/LRA.2018.2794578>.

Author ORCIDs

Author name	ORCID
Jang, Junwoo	0000-0003-0414-9622
Do, Haggi	0000-0002-8338-2977
Kim, Jinwhan	0000-0001-6886-2449